

# Учебник для экзамена LPI 101: Аппаратные средства и архитектура

Администрирование Linux для начинающих (LPIC-1), тема 101

Ян (Ian) Шилдс (Shields), Senior Programmer, EMC

**Описание:** Этим учебником, Ян Шилдс (Ian Shields) начинает готовить вас к сдаче Экзамена LPI 101 Профессионального Института Linux (Linux Professional Institute®):

Администрирование Linux для начинающих (Junior Level Administration) (LPIC-1). В этом первом (из пяти) учебнике Ян знакомит вас с настройкой вашего аппаратного обеспечения в Linux™. К концу этого учебника вы узнаете как Linux конфигурирует найденные аппаратные ресурсы на вашем современного ПК и где следует искать решение возникших проблем.

[Больше статей из этой серии](#)

**Дата:** 08.08.2005

**Уровень сложности:** простой

## Перед тем как начать

Узнайте чему может научить вас этот учебник и как извлечь из него максимум.

## Об этой серии учебников

Профессиональный Институт Linux (Linux Professional Institute -- LPI) осуществляет сертификацию системных администраторов в Linux по двум уровням: *для начинающих* (также называемый "первый уровень сертификации (certification level 1)") и *для специалистов* (Также называемый "Второй уровень сертификации ( certification level 2)"). Для достижения первого уровня сертификации вы должны сдать экзамены LPI 101 и LPI 102; для достижения второго уровня - экзамены LPI 210 и LPI 202..

developerWorks предоставляет учебники, помогающие в подготовке к каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем и для каждой темы существует соответствующий учебник для самостоятельного изучения на developerWorks. Экзамен LPI 101 содержит пять тем, которым соответствуют учебники от developerWorks:

Таблица 1. Экзамен LPI 101: Учебники и темы

Тема экзамена LPI 101	Учебник от developerWorks	Краткое содержание учебника
Тема 101	Учебник для экзамена LPI 101 (тема 101): Аппаратное обеспечение и архитектура	(Текущий учебник). Обучает конфигурированию ваших аппаратных ресурсов в Linux. К концу этого учебника, вы узнаете как Linux конфигурирует устройства, обнаруженные на современном компьютере и где искать решения возникших проблем.
Тема 102	<a href="#">Учебник для экзамена LPI 101:</a> <a href="#">Установка Linux и управление пакетами</a>	Представляет собой введение в установку Linux и управление пакетами. К концу этого учебника вы узнаете как Linux использует разделы жесткого диска, как Linux загружается, и как устанавливать и управлять пакетами программного обеспечения.

Тема 103 [Учебник для экзамена LPI 101: GNU и команды UNIX](#) Представляет собой введение в основы GNU и команды UNIX. К концу данного учебника вы узнаете как использовать команды в командной оболочке bash, включая использование команд текстовых процессоров и фильтров, как искать файлы и каталоги и как управлять процессами.

Тема 104 [Учебник для экзамена LPI 104: Устройства, файловые системы Linux, и стандарты иерархии файловых систем \[Filesystem Hierarchy Standard\]](#) Узнайте, как создавать файловые системы на разделах диска, а также как сделать их доступными для пользователей, управлять квотами пользователей и правами доступа к файлам, восстанавливать файловую систему при необходимости. Узнайте также о жестких и символьических ссылках, как найти файлы в вашей файловой системе и где их следует размещать. Детально смотри [список задач](#) ниже.

Тема 110 Система X Window Скоро ожидается.

Для того, чтобы сдать экзамены LPI 101 и LPI 102 (и достичь первого уровня сертификации), вы должны уметь:

- работать в командной строке Linux
- Выполнять простые операции сопровождения: управлять учетными записями пользователей, производить резервирование и восстановление, а также выключать и перезагружать компьютер
- Устанавливать и настраивать рабочую станцию (включая X), подсоединяться к локальной сети или подключать отдельно стоящий компьютер в сеть Internet посредством модема

Для продолжения подготовки к сертификации первого уровня смотри [Учебники для экзамена LPI 101 на developerWorks](#). Прочти больше о [полном наборе LPI-учебников на developerWorks](#).

## Об этом учебнике

Добро пожаловать в "Аппаратное обеспечение и архитектуру," первый из пяти учебников, разработанных для подготовки к экзамену LPI 101. В этом учебнике будет говориться о аппаратной части ПК и архитектуре.

Это учебник организован в соответствии с рабочей программой LPI по этой теме. Грубо говоря, на экзамене следует ожидать больше вопросов по темам с большим рейтингом.

*Таблица 2. Аппаратное обеспечение и архитектура: Рабочая программа к экзамену, вопросы которой раскрыты в этом учебнике*

Вопросы экзамена	LPI	Рейтинг	Содержание вопроса
1.101.1 <a href="#">Настройка основных параметров BIOS</a>	Рейтинг 1	Вы научитесь конфигурировать базовые настройки аппаратной части компьютера, изменения параметры BIOS. Вы узнаете об использовании таких параметров, как LBA для жестких дисков IDE с количеством цилиндров более 1024, включении или выключении интегрированной периферии (integrated peripherals), и настройке системы с (или без) внешней периферией, такой как клавиатура. Мы также обсудим корректные настройки IRQ, DMA и I/O	

		адресов для всех портов, управляемых из BIOS и параметры для управления ошибками.
1.101.3 <a href="#"><u>Настройка модема и звуковой карты</u></a>	Рейтинг 1	Вы узнаете как убедиться, что устройство соответствует требованиям совместимости и как установить модем и звуковую карту. Вы научитесь настраивать модем для соединения по телефонной линии и использовать его для соединений по PPP, SLIP, или CSLIP.
1.101.4 <a href="#"><u>Настройка SCSI устройств</u></a>	Рейтинг 1	Вы узнаете как настроить SCSI-устройства, используя SCSI BIOS и необходимые утилиты Linux. Вы рассмотрите различные типы SCSI. Вы узнаете как установить загрузочное SCSI устройство и как настроить желаемый порядок загрузки в смешанной системе со SCSI и IDE.
1.101.5 <a href="#"><u>Настройка различных карт-расширений ПК</u></a>	Рейтинг 3	Вы узнаете о различии между ISA и PCI картами в отношении параметров конфигурирования. Вы узнаете как проверить настройки IRQ, DMA, и портов ввода/вывода (I/O ports) для избежания конфликтов между устройствами.
1.101.6 <a href="#"><u>Настройка коммуникационных устройств</u></a>	Рейтинг 1	Вы узнаете как установить и настроить различные внутренние и внешние устройства связи, такие как модем, ISDN адAPTERЫ и DSL-коммутаторы (DSL switch). Вы узнаете о требованиях совместимости (особенно важных, если в качестве модема используется win-модем), необходимых аппаратных настройках для внутренних устройств (IRQ, DMA, порты ввода/вывода), и загрузке и настройке подходящего драйвера устройства. Мы также рассмотрим требования настройки интерфейсов.
1.101.7 <a href="#"><u>Настройка USB-устройств</u></a>	Рейтинг 1	Вы узнаете о том как активировать поддержку USB и как использовать и конфигурировать различные USB-устройства. Вы узнаете о корректном выборе для вашего USB-чипсета и соответствующем модуле. Мы также рассмотрим основы архитектуры многоуровневой модели USB и различные модули, используемые на разных уровнях.

## Требования

Для данного учебника нет специальных требований. Чтобы извлечь максимум из этого учебника, вы должны иметь основные знания о Linux и рабочую версию системы Linux, в которой вы сможете выполнять команды, приведенные в данном учебнике.

Различные версии программ могут по разному выводить результаты работы, поэтому то, что будет у вас на экране может отличаться от того, что приведено в листингах и на рисунках этого учебника.

# Учебник для экзамена LPI 101: Аппаратные средства и архитектура

## Параметры BIOS

В этом разделе рассматривается материал по теме 1.101.1 экзамена LPI 101 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 1.

Мы начнем с общего обзора современных персональных компьютеров, а затем обсудим вопросы настройки компьютера. Мы сосредоточимся на компьютерах, использующих процессоры семейства x86, такие как Intel® Pentium® или AMD Athlon, и шину PCI, поскольку они наиболее распространены в настоящее время.

Многие затронутые здесь темы имеют множество пересечений с рабочими программами LPI для специфической периферии. В следующих разделах этого учебника мы будем ссылаться на этот раздел, как на базовый материал.

## Обзор компьютеров и BIOS

Современный персональный компьютер (или ПК) состоит из центрального процессора (ЦП или CPU) для выполнения вычислений, а также некоторого объема памяти для хранения данных, которые используются процессором. Для того, чтобы такое устройство было полезным мы подключаем к нему периферийные устройства, такие как клавиатура, "мышь", монитор, жесткий диск, CD или DVD привод, принтер, сканер и сетевая карта, позволяющие нам вводить, хранить, печатать, отображать и передавать данные.

В описанном компьютере память, используемая процессором, называется памятью с произвольным доступом (Random Access Memory -- RAM) [Прим.пер.: В русскоязычной литературе также широко используется термин ОЗУ -- Оперативное Запоминающее Устройство]. В стандартном ПК эта память является *временной*, то есть для хранения данных ей необходимо электричество. Выключите компьютер и эта память очистится. Посмотрим с другой стороны: когда мы выключаем ПК, мы превращаем его в набор устройств, которые ничего не делают, до тех пор, пока не будут перепрограммированы. Перепрограммирование происходит в момент включения машины; этот процесс назван *начальной загрузкой* или *загрузкой* компьютера.

## Процесс начальной загрузки и BIOS

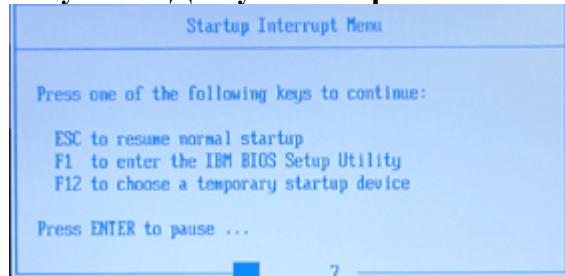
Процесс загрузки включает в себя загрузку операционной системы с внешнего устройства хранения информации, такого как floppy-диск (дискета), CD, DVD, жесткий диск, ключа защиты. Программа, выполняющая эту начальную загрузку постоянно хранится в компьютере и называется *Базовой системой ввода/вывода* (*Basic Input Output System -- BIOS*). BIOS хранится в постоянной памяти, иногда называемой *Память только для чтения* (*Read Only Memory -- ROM*) [Прим.пер.: В русскоязычной литературе также широко используется термин ПЗУ -- Постоянное Запоминающее Устройство]. В ранних ПК ROM-чип (микросхема ROM) был впаян или вставлен в специальное гнездо *материнской платы*). Обновление BIOS означало замену микросхемы ROM. Позднее стали использовать *электрически перепрограммируемую память только для чтения* (*Electrically Erasable Programmable Read Only Memories -- EEPROMs*). EEPROM позволяет обходиться без специального оборудования при обновлении BIOS. Сегодня наиболее часто встречающейся формой постоянной памяти является *Flash*-память, которая используется также и в цифровых камерах и ключах защиты. Flash-память также позволяет обновлять BIOS. [Прим.пер.: Изначально под ПЗУ понималась память именно *постоянная*, то есть микросхемы памяти, содержимое которых нельзя было изменить. Именно поэтому в старых материнских платах при обновлении BIOS необходимо

было менять саму микросхему, а не микрокод, как в настоящее время. Микросхемы же, допускавшие возможность перепрограммирования, назывались ППЗУ --  
Перепрограммируемое Постоянное Запоминающее Устройство]

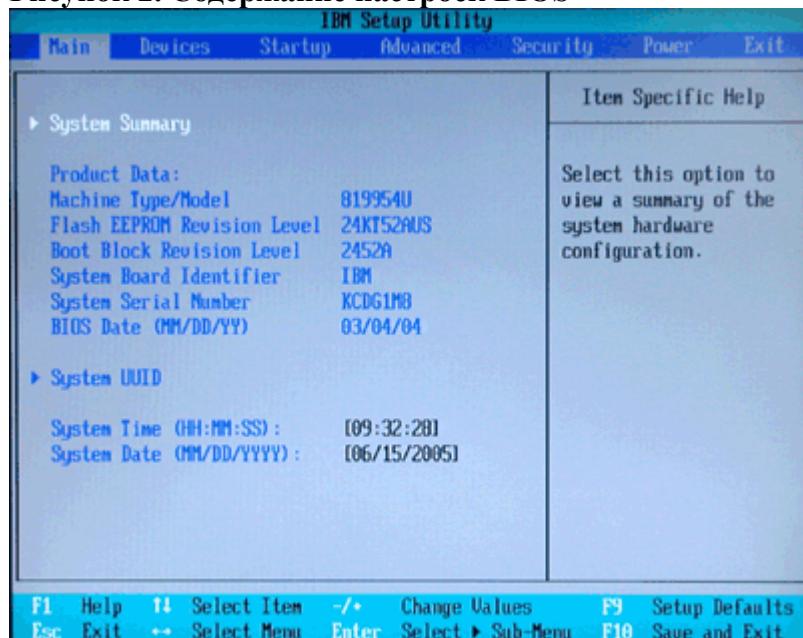
Кроме управления начальной загрузкой ПК, в настоящее время, программа BIOS обычно позволяет пользователю установить или проверить некоторые параметры конфигурации компьютера. Это включает в себя проверку установленных устройств, таких как RAM, жесткий диск, оптический привод, клавиатура, "мышь", и, возможно, встроенный монитор, звуковую карту и сетевые соединения. Пользователь может разрешить или запретить некоторые функции. Например, можно отключить встроенную в материнскую плату звуковую подсистему, чтобы использовать установленную звуковую карту. Пользователь также может выбрать устройство, с которого следует загружать систему, и установить пароль на доступ к компьютеру.

Чтобы получить доступ к экрану настройки BIOS вам потребуется подключенная к компьютеру клавиатура. При включении компьютера выполняется *Power On Self Test* [Самотестирование при включении] или *POST*. На некоторых компьютерах выводится сообщение с просьбой нажать определенную клавишу, чтобы запустить настройку, в противном случае будет продолжена нормальная загрузка, на других вы должны сами знать какую клавишу следует нажать до того, как запустится процесс стандартной загрузки, в то время как сообщение не появляется вообще или исчезло вследствие предыдущего изменения настроек. На некоторых компьютерах вам могут быть предложены и другие варианты кроме запуска программы настройки BIOS, например, как показано на Рисунке 1. В итоге вы должны увидеть окно вроде показанного на Рисунке 2.

**Рисунок 1. Доступ к настройкам BIOS**



**Рисунок 2. Содержание настроек BIOS**



Приведенные иллюстрации - это примеры того, что вы можете увидеть, вообще экраны настроек BIOS весьма разнообразны, поэтому не удивляйтесь, если на вашем компьютере они будут выглядеть иначе.

Рисунок 2 показывает нам, что Flash EEPROM (или система BIOS) имеет версию 24KT52AUS и датируется 4 Марта, 2004 тогда как текущая дата -- 9 Июня, 2005. Проверка сайта поддержки производителя (IBM) показывает, что имеется несколько версий BIOS, вышедших позднее, так что, возможно, хорошей мыслью будет обновить системную BIOS.

На Рисунке 2 можно увидеть несколько других пунктов меню. Мы рассмотрим их в следующих разделах этого учебника. Перед тем как сделать это, давайте еще поговорим о внутренней работе ПК.

## Шины, порты, IRQ, и DMA.

### Шины PCI и ISA

Периферийные устройства, включая те, что могут быть встроены в материнскую плату, взаимодействуют с процессором посредством шины. Наиболее широко используемый в настоящее время тип шины это шина *Peripheral Component Interconnect (связь периферийных компонентов)* или *PCI* которая практически заменила более раннюю шину *Industry Standard Architecture (Стандартная промышленная архитектура)* или *ISA*. Шина ISA иногда обозначалась как шина *AT* после IBM PC-AT в котором она была использована впервые в 1984. Во время перехода с ISA на PCI, многие компьютеры содержали обе шины со слотами (гнездами), позволявшими использовать или ISA, или PCI периферию.

Шина ISA поддерживала 8-битные и 16-битные карты, тогда как шина PCI поддерживает 32-битные устройства.

Существует еще пара стандартных шин, о которых вы также должны знать. Многие компьютеры имеют слот *Accelerated Graphics Port (Ускоренный графический порт)* или *AGP*, являющийся специальным слотом, основанным на спецификациях шины PCI 2.1, но оптимизированным по пропускной способности и скорости отклика, что необходимо для графических карт. Он медленно вытесняется новой шиной *PCI Express* или *PCI-E*, которая лишена многих ограничений базовой конструкции PCI.

Многое о файловых системах Linux мы узнаем в последующих учебниках этой серии, но сейчас мы рассмотрим файловую систему */proc*. Это не реальная файловая система на диске, а "псевдо-файловая система", предоставляющая информацию о работающем компьютере. В этой файловой системе файл */proc/pci* содержит информацию об устройствах, подключенных к шине PCI. Ниже приводятся некоторые соображения о том как избежать работы с этим специфичным файлом, поскольку команда [lspci](#) предоставляет ту же информацию.

Выполните команду [cat /proc/pci](#) и вы увидите нечто похожее на Листинг 1.

### Листинг 1. */proc/pci*

```
PCI devices found:
Bus 0, device 0, function 0:
  Host bridge: Intel Corp. 82845G/GL [Brookdale-G] Chipset Host Bridge
    (rev 1).
    Prefetchable 32 bit memory at 0xd0000000 [0xffffffff].
Bus 0, device 2, function 0:
  VGA compatible controller: Intel Corp. 82845G/GL [Brookdale-G] Chipset
    Integrated Graphics Device (rev 1).
    IRQ 11.
    Prefetchable 32 bit memory at 0x88000000 [0x8fffffff].
    Non-prefetchable 32 bit memory at 0x80000000 [0x8007ffff].
Bus 0, device 29, function 0:
  USB Controller: Intel Corp. 82801DB USB (Hub #1) (rev 1).
```

```
IRQ 11.  
I/O at 0x1800 [0x181f].  
Bus 0, device 29, function 1:  
    USB Controller: Intel Corp. 82801DB USB (Hub #2) (rev 1).  
        IRQ 10.  
        I/O at 0x1820 [0x183f].  
Bus 0, device 29, function 2:  
    USB Controller: Intel Corp. 82801DB USB (Hub #3) (rev 1).  
        IRQ 5.  
        I/O at 0x1840 [0x185f].  
Bus 0, device 29, function 7:  
    USB Controller: Intel Corp. 82801DB USB2 (rev 1).  
        IRQ 9.  
        Non-prefetchable 32 bit memory at 0xc0080000 [0xc00803ff].  
Bus 0, device 30, function 0:  
    PCI bridge: Intel Corp. 82801BA/CA/DB/EB PCI Bridge (rev 129).  
        Master Capable. No bursts. Min Gnt=4.  
Bus 0, device 31, function 0:  
    ISA bridge: Intel Corp. 82801DB LPC Interface Controller (rev 1).  
Bus 0, device 31, function 1:  
    IDE interface: Intel Corp. 82801DB Ultra ATA Storage Controller  
        (rev 1).  
        IRQ 5.  
        I/O at 0x1860 [0x186f].  
        Non-prefetchable 32 bit memory at 0x60000000 [0x600003ff].  
Bus 0, device 31, function 3:  
    SMBus: Intel Corp. 82801DB/DBM SMBus Controller (rev 1).  
        IRQ 9.  
        I/O at 0x1880 [0x189f].  
Bus 0, device 31, function 5:  
    Multimedia audio controller: Intel Corp. 82801DB AC'97 Audio  
        Controller (rev 1).  
        IRQ 9.  
        I/O at 0x1c00 [0x1cff].  
        I/O at 0x18c0 [0x18ff].  
        Non-prefetchable 32 bit memory at 0xc0080c00 [0xc0080dff].  
        Non-prefetchable 32 bit memory at 0xc0080800 [0xc00808ff].  
Bus 2, device 8, function 0:  
    Ethernet controller: Intel Corp. 82801BD PRO/100 VE (LOM) Ethernet  
        Controller (rev 129).  
        IRQ 9.  
        Master Capable. Latency=66. Min Gnt=8.Max Lat=56.  
        Non-prefetchable 32 bit memory at 0xc0100000 [0xc0100fff].  
        I/O at 0x2000 [0x203f].
```

Вы можете сравнить это с результатом получаемым по команде `lspci`. Обычно суперпользователь (root) не указывает путь для запуска этой команды, но обычным пользователям скорее всего потребуется его указать: [/sbin/lspci](#). Попробуйте выполнить это на своей машине.

## Порты ввода/вывода (IO Ports)

Когда процессору необходимо связаться с периферийными устройствами, он делает это через *порт ввода/вывода* (иногда его называют просто *порт*). Когда процессору необходимо передать данные или управляющую информацию для периферии он записывает ее в порт. Когда на устройстве есть данные или оно имеет статус готовности для процессора, то процессор читает данные или статус из порта. Большинство устройств имеют больше одного порта, ассоциированных с ними, обычно из числа измеряется первыми степенями двойки, такими как 8, 16 или 32. Передача данных обычно производится одним или двумя байтами.

Устройства не могут использовать порты одновременно, поэтому если у вас есть ISA-карты, вы должны убедиться, что каждое устройство имеет связанный с ним порт или порты. Изначально это делалось при помощи переключателей и перемычек на карте, некоторые поздние ISA-карты использовали систему под названием *Plug and Play* (*Подключи и Работай*) или *PnP*, которая будет обсуждаться позже в этом же разделе. PCI-карты все имеют PnP-настройки.

В файловой системе /proc, файл /proc/ioports говорит нам о портах ввода/вывода доступных в системе. Выполните команду `cat /proc/ioports`, чтобы увидеть результат (он будет похож на Листинг 2).

## Листинг 2. /proc/ioports

```
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0070-007f : rtc
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02f8-02ff : serial(auto)
0376-0376 : ide1
0378-037a : parport0
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial(auto)
0cf8-0cff : PCI conf1
1800-181f : Intel Corp. 82801DB USB (Hub #1)
    1800-181f : usb-uhci
1820-183f : Intel Corp. 82801DB USB (Hub #2)
    1820-183f : usb-uhci
1840-185f : Intel Corp. 82801DB USB (Hub #3)
    1840-185f : usb-uhci
1860-186f : Intel Corp. 82801DB Ultra ATA Storage Controller
    1860-1867 : ide0
    1868-186f : ide1
1880-189f : Intel Corp. 82801DB/DBM SMBus Controller
18c0-18ff : Intel Corp. 82801DB AC'97 Audio Controller
    18c0-18ff : Intel ICH4
1c00-1cff : Intel Corp. 82801DB AC'97 Audio Controller
    1c00-1cff : Intel ICH4
2000-203f : Intel Corp. 82801BD PRO/100 VE (LOM) Ethernet Controller
    2000-203f : e100
```

Порты нумеруются при помощи шестнадцатиричный чисел. Без сомнения из того, что вы увидите, кое-что вам покажется знакомым, например клавиатура (keyboard), таймер (timer), параллельный порт (parallel -- принтер), последовательный порт (serial -- модем) и видеокарта (vga+). Сравните это со некоторыми стандартными ассоциациями портов ввода/вывода, показанными на Листинге 3. Следует отметить, что, например, первый параллельный порт (parport0) владеет адресами в диапазоне от 0378 до 037A, что отображено в листинге /proc/ioports, но стандарт (LPT1) допускает использование для него диапазона от 378 до 37F.

### Листинг 3. Стандартные установки для портов ввода/вывода

```
1F0-1F8 - Hard Drive Controller, 16-bit ISA  
200-20F - Game Control  
210 - Game I/O  
220 - Soundcard  
278-27F - LPT2  
2F8-2FF - COM2  
320-32F - Hard Drive Controller, 8-bit ISA  
378-37F - LPT1  
3B0-3BF - Monochrome Graphics Adapter (MGA)  
3D0-3DF - Colour Graphics Adapter (CGA)  
3F0-3F7 - Floppy Controller  
3F8-3FF - COM1
```

### Прерывания

Итак, как процессор узнает когда заканчивается последний вывод или когда появляются данные для чтения? Обычно эта информация берется из регистра статуса, который может быть доступен при чтении из одного (или нескольких) портов, связанных с устройством. Но тут возникают две очевидные проблемы. Во-первых, процессор тратит время на проверку статуса. Во-вторых, если на устройстве есть данные, которые откуда-то поступают, вроде модема, то данные должны быть своевременно считаны, в противном случае они будут перезаписаны следующей порцией данных..

Эти две проблемы: бесполезных пустых циклов процессора и уверенности в своевременном считывании или записи данных решается посредством концепции *прерываний*. Прерывания также называются *Запросы на прерывание* или *IRQ*. Если с устройством происходит что-то о чём должен знать процессор, устройство вызывает прерывание и процессор временно останавливается, чтобы он не делал в настоящий момент.

Вспоминая материал предыдущего раздела, вряд ли будет удивительно, что информация о прерываниях также хранится в файловой системе /proc, в /proc/interrupts. Выполните команду [cat /proc/interrupts](#), чтобы увидеть результат, похожий на Листинг 4.

### Листинг 4. /proc/interrupts

CPU0		
0:	226300426	XT-PIC timer
1:	92913	XT-PIC keyboard
2:	0	XT-PIC cascade
5:	0	XT-PIC usb-uhci
8:	1	XT-PIC rtc
9:	2641134	XT-PIC ehci-hcd, eth0, Intel ICH4
10:	0	XT-PIC usb-uhci
11:	213632	XT-PIC usb-uhci
14:	1944208	XT-PIC ide0
15:	3562845	XT-PIC idel
NMI:	0	
ERR:	0	

Теперь прерывания нумеруются при помощи десятичных чисел в диапазоне от 0 до 15. И вновь сравните свой результат со стандартным распределением прерываний для ПК, показанным в Листинге 5.

## Листинг 5. Стандартные настройки IRQ

```
IRQ 0 - System Timer
IRQ 1 - Keyboard
IRQ 2(9) - Video Card
IRQ 3 - COM2, COM4
IRQ 4 - COM1, COM3
IRQ 5 - Available (LPT2 or Sound Card)
IRQ 6 - Floppy Disk Controller
IRQ 7 - LPT1
IRQ 8 - Real-Time Clock
IRQ 9 - Redirected IRQ 2
IRQ 10 - Available
IRQ 11 - Available
IRQ 12 - PS/2 Mouse
IRQ 13 - Math Co-Processor
IRQ 14 - Hard Disk Controller
IRQ 15 - Available
```

Изначально каждое устройство имеет свое собственное IRQ. Заметим, например, что в Листинге 5 IRQ5 часто используется или звуковой картой, или вторым параллельным портом (принтер). Если вам необходимы оба устройства, вы можете найти карты, которые могут быть настроены (обычно при помощи перемычек) на использование другого прерывания, например, IRQ15.

В настоящее время, PCI-устройства используют IRQ совместно, таким образом, когда что-то останавливает процессор, обработчик прерывания проверяет ему ли предназначено это прерывание и если нет, то передает следующему в цепочке. Листинги 4 и 5 не говорят нам об этом совместном использовании. Мы изучим команду [grep](#) в следующем учебнике, но сейчас мы сможем использовать ее для фильтрации вывода результата команды [dmesg](#) чтобы увидеть сообщения начальной загрузки о прерываниях (IRQ), как показано в Листинге 6, в котором совместно используемые прерывания мы выделили.

## Листинг 6. Прерывания, обнаруженные при начальной загрузке.

```
[ian@lyrebird ian]$ dmesg | grep -i irq
PCI: Discovered primary peer bus 01 [IRQ]
PCI: Using IRQ router PIIIX [8086/24c0] at 00:1f.0
PCI: Found IRQ 5 for device 00:1f.1
PCI: Sharing IRQ 5 with 00:1d.2
Serial driver version 5.05c (2001-07-08) with MANY_PORTS MULTIPORT
    SHARE_IRQ SERIAL_PCI ISAPNP enabled
ttyS0 at 0x03f8 (irq = 4) is a 16550A
ttyS1 at 0x02f8 (irq = 3) is a 16550A
PCI: Found IRQ 5 for device 00:1f.1
PCI: Sharing IRQ 5 with 00:1d.2
ICH4: not 100% native mode: will probe irqs later
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
ide1 at 0x170-0x177,0x376 on irq 15
PCI: Found IRQ 11 for device 00:1d.0
PCI: Sharing IRQ 11 with 00:02.0
usb-uhci.c: USB UHCI at I/O 0x1800, IRQ 11
PCI: Found IRQ 10 for device 00:1d.1
usb-uhci.c: USB UHCI at I/O 0x1820, IRQ 10
PCI: Found IRQ 5 for device 00:1d.2
PCI: Sharing IRQ 5 with 00:1f.1
```

```
usb-uhci.c: USB UHCI at I/O 0x1840, IRQ 5
PCI: Found IRQ 9 for device 00:1d.7
ehci-hcd 00:1d.7: irq 9, pci mem f885d000
parport0: irq 7 detected
PCI: Found IRQ 9 for device 02:08.0
PCI: Found IRQ 9 for device 02:08.0
parport0: irq 7 detected
PCI: Found IRQ 11 for device 00:02.0
PCI: Sharing IRQ 11 with 00:1d.0
PCI: Found IRQ 9 for device 00:1f.5
PCI: Sharing IRQ 9 with 00:1f.3
i810: Intel ICH4 found at IO 0x18c0 and 0x1c00, MEM 0xc0080c00 and
      0xc0080800, IRQ 9
```

## DMA

Ранее мы упоминали, что связь с периферийными устройствами через порты ввода/вывода производится одним или двумя байтами одновременно. Для быстрых устройств обслуживание прерываний может поглотить большую часть возможностей процессора. Более быстрым методом является использование *Direct Memory Access* (*Прямого доступа к памяти*) или *DMA*, при котором несколько инструкций ввода/вывода сообщают устройству куда в оперативной памяти (ОЗУ) можно писать или откуда читать данные и затем контроллер DMA осуществляет низкоуровневое управление реальными потоками данных между ОЗУ и периферийными устройствами.

Поднимите руку те, кто догадался где можно найти информацию об используемых каналах DMA. Если вы сказали, что в /proc/dma, то вы правы. Выполните команду [cat /proc/dma](#). Чтобы увидеть нечто подобное Листингу 7.

### Листинг 7. /proc/dma

```
4: cascade
```

Это все? Важно запомнить, что большинство устройств будут использовать только один из ограниченного числа DMA-каналов при реальных приеме/передаче, поэтому /proc/dma часто будет выглядеть практическим пустым, как в нашем примере. Мы также можем просканировать сообщения загрузки для выявления DMA-совместимых устройств также, как мы делали это в предыдущем случае с IRQ. Листинг 8 показывает типичный результат.

### Листинг 8. /proc/dma

```
[ian@lyrebird ian]$ dmesg | grep -i dma
ide0: BM-DMA at 0x1860-0x1867, BIOS settings: hda:DMA, hdb:pio
ide1: BM-DMA at 0x1868-0x186f, BIOS settings: hdc:DMA, hdd:DMA
hda: 312581808 sectors (160042 MB) w/8192KiB Cache,
      CHS=19457/255/63, UDMA(100)
hdc: 398297088 sectors (203928 MB) w/7936KiB Cache,
      CHS=24792/255/63, UDMA(33)
ehci-hcd 00:1d.7: enabled 64bit PCI DMA
```

## Plug and play

Ранние ПК обладали фиксированным числом потров и IRQ для отдельных устройств, таких

как клавиатура или параллельный порт принтера. Это осложняло добавление новых устройств или даже использование двух однотипных, вроде двух модемов или двух принтеров, первый последовательный порт обычно назывался COM1, а второй -- COM2. Linux обычно ссылается на них как *ttyS0* и *ttyS1*. Некоторые карты можно было настраивать при помощи перемычек, которые позволяли модему, например, работать по порту COM1 или COM2. По мере увеличения устройств, исходного количества выделяемых адресов под порты и прерывания стало не хватать, и была разработана технология *Plug and Play* или *PnP*. Идея заключалась в разрешении устройству сообщать системе сколько и каких ресурсов ему необходимо, а BIOS затем сообщала устройству какие из имеющихся ресурсов ему следует использовать. Это полуавтоматическое конфигурирование было представлено в IBM PS/2 который использовал шинную архитектуру, названную *microchannel*. Позднее, эта идея и имя plug and play было использовано и в ISA-картах, в основном в модемах и звуковых картах, которые были наиболее популярными картами-расширениями в то время. Шина PCI продвинула идею дальше и все PCI-устройства от рождения являются plug and play.

Если вам случится работать за компьютерами с ISA PnP устройствами, то помните, что вам следует избегать конфликтов портов и прерываний между устройствами. Порты не могут быть использованы двумя устройствами одновременно; каждое устройство **должно** иметь свой собственный порт. Это же имеет место и для DMA каналов. За некоторыми исключениями, ISA-устройства не могут совместно использовать и IRQ. Если у вас есть не-PnP устройства, то вы должны вручную сконфигурировать каждое устройство, чтобы оно не пересекалось с другими. PnP должно выполнять конфигурирование автоматически. Однако, с некоторыми ISA-устройствами это не всегда хорошо работает. Вы можете разрешить конфликты, используя *isapnptools*, обсуждаемую ниже, или вы можете переназначить некоторые порты и IRQ не PnP устройствам, чтобы система заработала.

Для систем на основе ядер, предшествующих 2.4, пакет *isapnptools* позволял пользователям настраивать PnP-устройства. Команда *isapnpr* считывает файл конфигурации (обычно */etc/isapnpr.conf*) для настройки PnP-устройств. Обычно это выполняется во время загрузки Linux. Команда *pnpdump* сканирует PnP-устройства и формирует список ресурсов, которые или необходимы, или предпочтительны для использования вашими PnP-картами. Формат полученного списка подходит для использования командой *isapnpr*, как только вы раскомментируете команды, которые хотите использовать. Вы должны убедиться, что конфликты ресурсов устранены. Обратитесь к тан-страницам команд *isapnpr* и *pnpdump* для дополнительной информации по их использованию

Начиная с ядра 2.4, поддержка PnP внедрена в ядро и пакет *isapnptools* стал не нужен. К примеру, он был удален из Red Hat 7.3, выпущенной в мае 2002. Поддержка PnP схожа с поддержкой PCI, описанной ранее. Вы можете использовать команду *lspnp* (часть пакета *kernel-pcmcia-cs*) для отображения информации о PnP-устройствах. Эта информация содержится в файле */proc/bus/pnp*. Этот файл отсутствует в системах, содержащих только PCI.

## Жесткие диски IDE

В современных ПК наиболее распространены жесткие диски *Integrated Drive Electronics* или *IDE* [Прим.пер.: На самом деле сейчас (середина 2006 г.) наиболее распространены диски SATA, обсуждаемые ниже, а IDE медленно исчезают]. Они также известны как *AT Attachment* или *ATA* диски, что появилось после IBM PC-AT. Другим используемым типом дисков является также достаточно популярный интерфейс *Small Computer System Interface* или *SCSI*, особенно на серверах. IDE диски имеют преимущество низкой цены, а SCSI интерфейс позволяет подключить большее количество дисков, с большим потенциалом перекрытия операций для различных дисков на однойшине и, как следствие, большей производительностью.

Недавно на рынке появился новый тип дисков, под названием *Serial ATA* или *SATA*. Спецификации SATA призваны устранить некоторые ограничения спецификаций ATA,

обеспечив хорошую совместимость с ATA.

## BIOS и размер IDE дисков

IDE диски разбиты на *сектора (sector)*, единица данных в 512 байт. Жесткий диск может состоять из нескольких вращающихся дисков-пластин, так что сектора располагаются в концентрических окружностях, каждая из которых называется *цилиндром (cylinder)*. Данные с каждого отдельного дискачитываются или записываются при помощи *головки (head)*. Чтобы найти некоторый сектор, диск перемещает набор головок, связанных с цилиндром, выбирает соответствующую головку и ждет когда требуемый сектор окажется под головкой. Отсюда возникает термин *CHS* (Cylinder [Цилиндр], Head [Головка] и Sector [Сектор]). Можно также услышать другое название *геометрия диска*.

К сожалению, исторически, ранние BIOS содержали ограничение величины каждого из параметров C, H и S, а DOS, популярная операционная система для ПК, содержала дополнительные ограничения. На протяжении 1990-ых, размеры дисков быстро превзошли искусственные ограничения CHS, накладываемые BIOS и DOS. Привлекались различные стратегии для перевода реальных значений CHS в "виртуальные", из-за чего возникали проблемы связанные либо с самой BIOS, либо с низкоуровневыми приложениями типа Ontrack Disk Manager.

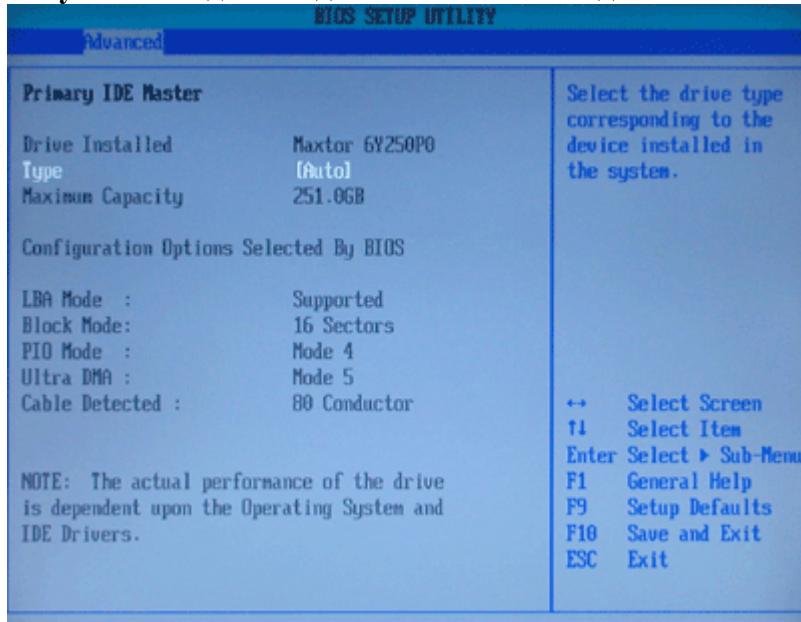
Даже без искусственных ограничений BIOS или DOS, идея CHS допускала максимум 65536 цилиндов, 16 головок, и 255 секторов/дорожек. Это ограничивает вместимость 267386880 секторами, или приблизительно 137 ГБ. Заметьте, что размер дисков, в отличие от других размеров в ПК измеряется в степенях 10, так что 1ГБ=1,000,000,000 байт.

Идея была в том, чтобы система игнорировала геометрию и оставляла ее обработку самому диску. Вместо обращения к значению CHS, система просто запрашивает *Logical Block Address* (*Адрес Логического блока*) или *LBA*, а электроника диска вычисляет к какому реальному сектору идет обращение на чтение/запись. Этот процесс был стандартизирован в 1996 с принятием стандарта ATA-2 (ANSI стандарт X3.279-1996, *AT Attachment Interface with Extensions* [*Интерфейс AT с расширениями*]).

Как говорилось ранее, BIOS необходима для загрузки системы, поэтому ей необходимо знать об организации диска, чтобы загрузить стартовую программу. Старые BIOS, не понимающие LBA-диски могут иметь ограничение по загрузке с первых 1024 цилиндов диска, или, по крайней мере, с первых 1024 цилиндов, в понимании геометрии диска самой BIOS! Такие BIOS вероятно в настоящее время весьма редки, но если вам необходимо работать с одной из них, она может иметь установки для поддержки LBA, а также вам, вероятно, понадобится разместить каталог /boot в разделе в первых 1024 цилиндрах. Даже когда ваша система будет прекрасно загружаться с самого конца самого большого диска, многие инструменты разбики диска в Linux будут предупреждать вас о том, что раздел находится за пределами первых 1024 цилиндов.

На Рисунке 3 показана информация, доступная в BIOS моей материнской платы Intel для 250ГБ IDE диска на одной из моих Linux-систем.

Рисунок 3. Вид BIOS для большого LBA-диска



В листинге 9 показана часть результата применения команды `hdparm -I /dev/hda` для диска подобного приведенному на Рисунке 3 в системе Linux (в данном случае Fedora Core 3). Заметьте, что значения CHS ограничены 4,128,705 секторами, значение LBA установлено в 268,435,455 секторов или 137ГБ. А реальный объем измеряется в единицах LBA48. Это 490,234,752 сектора или 251ГБ.

#### Листинг 9. Результат команды `hdparm -I /dev/hda`

```
/dev/hda:
ATA device, with non-removable media
      Model Number: Maxtor 6Y250P0
      Serial Number: Y638VBWE
      Firmware Revision: YAR41BW0
Standards:
      Supported: 7 6 5 4
      Likely used: 7
Configuration:
      Logical      max      current
      cylinders    16383    65535
      heads        16       1
      sectors/track 63       63
      --
      CHS current addressable sectors:      4128705
      LBA    user addressable sectors:   268435455
      LBA48  user addressable sectors: 490234752
      device size with M = 1024*1024:      239372 MBytes
      device size with M = 1000*1000:     251000 MBytes (251 GB)
Capabilities:
      LBA, IORDY(can be disabled)
      Queue depth: 1
      ..
```

По умолчанию ПК загружается с первого IDE диска компьютера. Некоторые компьютеры имеют пункты BIOS, позволяющие изменить это, однако большинство загружаются именно так. Сначала компьютер загружает маленький кусочек кода из *master boot record* (главной

загрузочной записи) которая, в свою очередь, предоставляет информацию с какого раздела грузиться. Мы рассмотрим подробнее начальные загрузчики для Linux в следующих учебниках.

## Имена дисков Linux

Большую часть о том, как Linux использует диски мы рассмотрим в последующих учебниках данной серии. Однако, сейчас хорошо бы представить вам другую важную файловую систему Linux -- `/dev`. Это, подобно `/proc`, псевдо-файловая система описывающая устройства, которые могут быть в Linux системах. Внутри файловой системы `/dev` вы обнаружите такие записи, как `/dev/hda`, `/dev/hda5`, `/dev/sda`, `/dev/sdb1` и так далее. Вы также найдете множество других записей для других типов устройств, но сейчас давайте взглянем на те из них, которые начинаются или с `/dev/hd`, или с `/dev/sd`.

Устройства, начинающиеся с `/dev/hd`, такие как `/dev/hda` или `/dev/hda5` ссылаются на **IDE-диски**. Первый диск первого контроллера IDE это `/dev/hda`, а второй, если имеется, это `/dev/hdb`. Точно также первый диск второго контроллера IDE это `/dev/hdc`, а второй -- `/dev/hdd`. Как видно из Листинга 10, в `/dev` прописано намного больше, нежели имеется в вашей системе.

### Листинг 10. Записи `/dev/hd?` и `/dev/sd?`

```
[ian@lyrebird ian]$ ls /dev/hd?
/dev/hda /dev/hdd /dev/hdg /dev/hdj /dev/hdm /dev/hdp /dev/hds
/dev/hdb /dev/hde /dev/hdh /dev/hdk /dev/hdn /dev/hdq /dev/hdt
/dev/hdc /dev/hdf /dev/hdi /dev/hdl /dev/hdo /dev/hdr
[ian@lyrebird ian]$ ls /dev/sd?
/dev/sda /dev/sde /dev/sdi /dev/sdm /dev/sdq /dev/sdu /dev/sdy
/dev/sdb /dev/sdf /dev/sdj /dev/sdn /dev/sdr /dev/sdv /dev/sdz
/dev/sdc /dev/sdg /dev/sdk /dev/sdo /dev/sds /dev/sdw
/dev/sdd /dev/sdh /dev/sdl /dev/sdp /dev/sdt /dev/sdx
```

Также как и для IRQ ранее, мы можем использовать команду `dmesg`, чтобы узнать какие дисковые устройства обнаружены во время загрузки. Вывод для одной из моих машин приведен в Листинге 11.

### Листинг 11. Жесткие диски, обнаруженные при загрузке

```
[ian@lyrebird ian]$ dmesg | grep "[hs]d[a-z]"
Kernel command line: ro root=LABEL=RHEL3 hdd=ide-scsi
ide_setup: hdd=ide-scsi
    ide0: BM-DMA at 0x1860-0x1867, BIOS settings: hda:DMA, hdb:pio
    ide1: BM-DMA at 0x1868-0x186f, BIOS settings: hdc:DMA, hdd:DMA
hda: WDC WD1600JB-00EVA0, ATA DISK drive
hdc: Maxtor 6Y200P0, ATA DISK drive
hdd: SONY DVD RW DRU-700A, ATAPI CD/DVD-ROM drive
hda: attached ide-disk driver.
hda: host protected area => 1
hda: 312581808 sectors (160042 MB) w/8192KiB Cache,
    CHS=19457/255/63, UDMA(100)
hdc: attached ide-disk driver.
hdc: host protected area => 1
hdc: 398297088 sectors (203928 MB) w/7936KiB Cache,
    CHS=24792/255/63, UDMA(33)
hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 hda9 hda10 hdall >
hdc: hdc1 < hdc5 hdc6 hdc7 hdc8 >
```

```
hdd: attached ide-scsi driver.
```

Из Выделенных строк Листинга 11, мы можем видеть, что в системе имеется два IDE-диска (hda и hdc), а также привод DVD-RW (hdd). Заметьте, hdb нет, что говорит о том, что на первом IDE-контроллере второго диска нет. Диск может иметь четыре *основных* раздела (*primary*) и неограниченное количество *логических* (*logical*). Рассмотрев диск hdc Листинга 11, мы можем увидеть, что он имеет один *primary*-раздел (hdc1) и четыре логических (hdc5, hdc6, hdc7, и hdc8). В теме 104 в последующем учебнике этой серии мы увидим, что на самом деле hdc1 это контейнер (или *расширенный (extended)* раздел) для логических разделов.

Исторически такие устройства, как sda и sdb являлись **SCSI-дисками**, которые мы рассмотрим далее, при изучении [настройки SCSI-устройств](#) для ядра 2.4, IDE CD и DVD устройства обычно управляются через эмуляцию SCSI. Такие устройства часто появляются в /dev похожие на /dev/cdrom, что является символьной ссылкой на эмулируемое SCSI-устройство. Для описанной системы, Листинг 12 показывает, что /dev/cdrom это ссылка на /dev/scd0, а не на /dev/hdd, как можно было бы ожидать. Заметьте, что параметр ядра hdd=ide-scsi в Листинге 11 является указанием, что ide-scsi привод был присоединен к hdd.

## Листинг 12. IDE SCSI-эмulation

```
[ian@lyrebird ian]$ ls -l /dev/cdrom
lrwxrwxrwx 1 root    root   9 Jan 11 17:15 /dev/cdrom -> /dev/scd0
```

Сейчас вы убедитесь, что и **USB**, и **SATA** устройства хранения обозначаются как **sd**, а не **hd**.

## Стандартная периферия

Выше мы упоминали такую периферию, как последовательный и параллельный порты, которые обычно интегрированы в материнскую плату, и рассмотрели некоторые стандартные порты ввода/вывода, а также IRQ ассоциируемые с этими устройствами. Последовательные порты, в действительности, использовались для соединения различных устройств и исторически трудно настраиваются. С появлением устройств стандарта *IEEE 1394*, известного также, как *Firewire* и *Universal Serial Bus (Универсальной шины данных)* или **USB**, автоматическая настройка и "горячее" подключение устройств повсеместно заменило рутину корректной настройки последовательных и параллельных портов. Фактически, *legacy-free* (*свободные от наследства*) системы не поддерживают стандартные последовательный и параллельный порты. В них не поддерживается ни флоппи-драйв, ни PS/2 клавиатура, ни PS/2 "мышь".

Теперь мы обсудим некоторые основные настройки BIOS, которые вам может понадобиться изменить.

### Serial ports (COMn) [Последовательные порты]

Стандартные последовательные порты нумеруются от COM1 до COM4. Если в вашей системе есть единственное гнездо последовательного порта (изначально 25-контактный DB25 разъем, но в настоящее время обычно 9-контактный разъем DB9), то вероятно используются стандартные для COM1 адреса и прерывания (IRQ), а именно порт ввода/вывода (IO port) 3F8 и IRQ 4. Стандартные адреса и прерывания для последовательных портов приведены в Таблице 3.

*Таблица 3. Параметры последовательного порта*

Имя	Адрес	IRQ
COM1	3F8-3FF	4
COM2	2F8-2FF	3
COM3	3E8-3EF	4
COM4	2E8-2EF	3

Вы можете заметить, что COM1 и COM3 совместно используют IRQ 4, а COM2 и COM4, в свою очередь, IRQ 3. Поскольку драйвер и устройство в действительности не могут совместно использовать прерывания, и устройство не может вообще не использовать прерывание, то это означает, что реальная система использует только COM1 и COM2.

Изредка, вам может понадобиться или отключить встроенный последовательный порт, или настроить его на использование другого адреса и IRQ. Наиболее вероятной причиной сделать это является конфликт между PnP модемом и ISA-слотом или желание использовать PnP-модем на COM1. Мы рекомендуем изменять это только в том случае, если Linux не может определить вашу конфигурацию.

### **Parallel ports (LPTn) [Параллельные порты]**

Стандартные параллельные порты нумеруются от LPT1 до LPT4, хотя обычно присутствует только два. Если в вашем компьютере есть единственное гнездо параллельного порта, то он, вероятно, по умолчанию использует адрес и IRQ для LPT1, а именно порт ввода/вывода 378 и IRQ 7. Стандартные адреса портов ввода/вывода и IRQ для параллельных портов приведены в Таблице 4.

*Таблица 4. Параметры параллельного порта*

Имя	Адрес	IRQ
LPT1	378-37F	7
LPT2	278-27F	5
LPT*	3BC-3BE	

Заметьте, что порты ввода/вывода 3BC-3BE изначально использовались графическим адаптером Hercules, который также имеет параллельный порт. Многие системы BIOS присваивают этот диапазон LPT1 и затем два других диапазона становятся LPT2 и LPT3 соответственно, вместо LPT1 и LPT2.

Многие системы не используют прерываний для принтеров, поэтому IRQ реально может использоваться, а может и нет. Не редкость также совместное использование IRQ 7 для печати и звуковой карты (совместимой с Sound Blaster).

Параллельные порты изначально обычно использовались для печати с данными, поступающими на принтер и несколькими линиями, зарезервированными для отчета о статусе. Позднее, параллельные порты использовались для подсоединения различных устройств (включая ранние CD-ROM и ленточные приводы), в связи с чем направленность только на вывод данных сменилась двунаправленным потоком данных.

Текущий стандарт параллельных портов это *IEEE Std. 1284-1994 Standard Signaling Method for a Bi-Directional Parallel Peripheral Interface for Personal Computers* (Стандарт Метода Передачи сигналов для Двунаправленного Параллельного Интерфейса Периферии Персонального Компьютера), который определяет пять сигнальных режимов. Ваш BIOS во время настройки может предоставить вам на выбор один из них *bi-directional* (дву направленный), *EPP*, *ECP* и *EPP and ECP*. ECP расшифровывается как *Enhanced Capabilities Port* (Порт с Расширенными Возможностями) и разработан для использования с принтерами. EPP расшифровывается как *Enhanced Parallel Port* (Улучшенный Параллельный Порт) и разработан для таких устройств, как CD-ROMы и ленточный приводы, которым необходим значительный поток данных в обоих направлениях. Выбор BIOS по умолчанию

скорее всего это ECP. Как и для последовательных портов, изменять это следует только если у вас есть устройство, не функционирующее должным образом.

### **Порт Floppy дисковода**

Если в вашей системе присутствует стандартный контроллер floppy диска, то он использует порты 3F0-3F7. Если вы установите стандартный floppy-дисковод в компьютер, который продавался без него, то вам необходимо будет включить соответствующие опции в BIOS. Просмотрите документацию производителя для выяснения деталей.

### **Клавиатура и "мышь"**

Контроллер клавиатура/"мыши" использует порты 0060 и 0064 для стандартной клавиатуры и "мыши", подсоединенных к гнезду PS/2. Многие системы выдают ошибку Power-On-Self-Test (POST) [Самотестирование При Включении], если клавиатура не подключена. Большинство машин разработанных для использования в качестве серверов и многие настольные в настоящее время имеют опции BIOS для нормального запуска без клавиатуры.

Работать на компьютере без клавиатуры (или "мыши") проблематично. Но сервера часто работают именно так. Управление осуществляется посредством сети используя инструменты web-администрирования или интерфейс командной строки, такой как telnet или (предпочтительнее) ssh.

Установка безклавиатурной системы обычно подразумевает использование терминала (или эмулятора терминала), подсоединеного к последовательному порту. Обычно вам необходима клавиатура и монитор, чтобы убедиться, что BIOS настроена правильно и последовательный порт включен. Также вам может потребоваться сформированный загрузочный диск или CD для осуществления установки Linux.

Другой подход используется в системах JS20 blade server это эмуляция последовательного соединения в сети.

| [предыдущая](#) | [следующая](#)

[В начало](#)

## **Модемы и звуковые карты**

Этот раздел содержит материал по теме 1.101.3 экзамена LPI 101 Администрирование для начинающих (LPIC-1). Рейтинг темы - 1.

### **Модемы**

Модем (*modem*) (образовано от *modulator/demodulator*) это устройство для преобразования цифровых сигналов, используемых компьютером в последовательные потоки аналоговых данных, передаваемых по телефонной линии. В первые дни ПК, модемы были внешними устройствами, которые подсоединялись к последовательному порту. Позднее, модемы стали выпускаться в виде плат, которые могли быть вставлены внутрь системного блока, что снижало стоимость размещения и подключения к сети, ликвидируя необходимость в кабеле между последовательным портом и модемом. Следующим снижением стоимости стало перенесение некоторых функций, обычно выполнявшихся модемом, на программное обеспечение ПК. Модемы такого типа можно назвать, в зависимости от терминологии, soft-модемам, HCF модемами, HSP модемами, HSF модемами или модемами без контроллера (*controllerless modem*). Такие модемы были разработаны для снижения стоимости систем, которые в основном работают под Microsoft Windows. Термин *win-модем* часто используется в отношении этих устройств, хотя Win-модем® является зарегистрированной торговой маркой U.S. Robotics, которая производит различные модемы под этим именем.

Большинство внешних и полнофункциональных внутренних модемов будут работать без

проблем и в Linux. Некоторые из модемов, требующих помощи программного обеспечения Операционной системы ПК, также будут работать в Linux и список модемов этой категории постоянно пополняется. Soft-модемы, работающие в Linux часто называют *lin-модемами* и есть сайт, посвященный им ([linmodems.org](http://linmodems.org)). Если у вас такой модем, то в начале следует посетить указанный сайт lin-модемов и загрузить последнюю версию инструмента scanmodem. Она сообщит все что известно о доступном драйвере (если он есть) для вашего модема.

Если у вас ISA-модем, то вам нужно убедиться, что порты, IRQ и DMA каналы не конфликтуют с другими устройствами. Для дополнительной информации смотри предыдущий раздел [Настройки BIOS](#).

Модемы, рассматриваемые в данном разделе являются *асинхронными* модемами. Есть также другой класс модемов, называемых *синхронными* модемами, использующихся для HDLC, SDLC, BSC или ISDN. Упрощенно, можно сказать, что асинхронная передача применяется для передачи отдельных байтов информации, а синхронная связь используется для передачи блоков информации.

Большинство соединений Linux используют *Internet Protocol* или *IP*. Поэтому Linux-системам необходимо нечто вроде IP для асинхронных линий, которые изначально разрабатывались не для блоковых протоколов вроде IP. Первый вариант реализации этого назывался *Serial Line Interface Protocol* (*Протокол интерфейса последовательной линии*) или *SLIP*. Его вариант, использующий сжатые заголовки, назвали *CSLIP*. В наши дни, Большинство Internet-провайдеров (ISP -- Internet Service Provider) поддерживают dialup-соединения с использованием *Point-to-Point Protocol* или *PPP*.

*Linux Networking-HOWTO* и *Руководство сетевого администратора* доступные в проекте Linux Documentation Project (смотри [Ресурсы](#)) предоставляют информацию о настройке SLIP, CSLIP и PPP.

При связи посредством модема, существует множество параметров настройки, которые вам возможно придется изменить на вашем компьютере с Linux. Наиболее важно установить скорость соединения системы с модемом. Обычно она выбирается больше чем номинальная скорость соединения в линии, и часто устанавливается максимально возможное значение для последовательного порта и модема. Одним из способов настроить или посмотреть параметры модема, используемые драйвером последовательного порта, является программа **setserial**. Команда setserial проиллюстрирована Листингом 13. Заметьте, что параметр -G приводит к форматированному выводу, удобному для применения в качестве установочных параметров с setserial. В данном случае, UART (Universal Asynchronous Receiver Transmitter) имеет буфер 16550, что является стандартом для UART в современных ПК. Установлена скорость 115,200 bps, которая также обычно используется с этим UART для большинства современных внешних модемов 56kbps. Следует отметить, что скорость по умолчанию на некоторых более новых системах может быть установлена вплоть до 460,800bps. Если ваш модем не откликается, то это первое, что вы должны проверить.

### Листинг 13. Команда setserial

```
[root@attic4 ~]# setserial /dev/ttyS0  
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4  
[root@attic4 ~]# setserial -G /dev/ttyS0  
/dev/ttyS0 uart 16550A port 0x03f8 irq 4 baud_base 115200 spd_normal skip_test
```

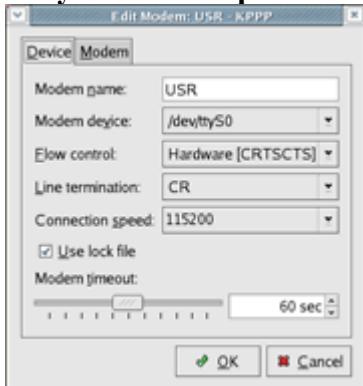
Единственное, что следует сказать особо о setserial так это то, что она не обращается к самому устройству. Все, что она выводит -- это параметры, используемые драйвером

последовательного порта, если только вы не используете параметры autoconfig и auto\_irq parameters. В этом случае setserial попросит ядро опросить устройство. Для более полной информации об этих и других параметрах командысмотрите man-страницы setserial.

Мы рассмотрим сеть более подробно в учебнике для экзамена LPI 102 (Смотри [Ресурсы](#)). А между тем если вы хотите настроить PPP-соединение, то имеется несколько превосходных инструментов, которые помогут вам сделать это. Программа kppp имеет приятный графический интерфейс и проста в использовании. Команда wvdial является хорошим инструментом командной строки для настройки dialup соединений. В добавок к этому, дистрибутивы могут иметь другие инструменты или специально для PPP или dialup соединений, или как часть более общих инструментов настройки сети, такие как system-config-network в Fedora Core 4.

Другим аспектом модемных соединений, который также обычно контролируется коммуникационной программой, но может быть изменён или иметь значение по умолчанию записанное в самом модеме, является *flow control* (*управление потоком данных*). Он определяет способ, при помощи которого один модем передаёт другому команду подождать, пока не будет очищен буфер получения данных. Такая команда может выдаваться программным обеспечением путем отправки символов XON и XOFF, но более хороший способ, используемый также для PPP-соединений, называется *hardware flow control* (*аппаратное управление потоком данных*) при котором для индикации готовности к приему данных используются значения сигнальной линии модема. Используемые сигналы называются *Clear to Send* (*Нечего передавать*) или CTS и *Ready to Send* (*Готов к передаче*) или RTS, так что вы часто будете видеть их или нечто подобное при описании управления потоком данных с использованием RTS/CTS. На Рисунке 4 показано как настроить скорость и аппаратное управление потоком данных с использованием программы kppp.

**Рисунок 4. Настройка параметров модема в kppp**



## Звуковые карты

Большинство персональных компьютеров, продаваемых в настоящее время содержат звуковую карту.

### Звуковой порт (Sound Blaster)

Серия звуковых карт The Creative Labs Sound Blaster series стала de facto промышленным стандартом для звуковых карт. Хотя существует множество превосходных звуковых карт других брендов, большинство из них предоставляют режим совместимости для одной или нескольких серий Sound Blaster. Оригинальная карта Sound Blaster была 8-битной и работала в оригинальном IBM PC. Позднее 16-битные модели для PC-AT и совместимые с ними использовали 16-битную шину PC-AT или ISA. Сейчас, большинство таких карт используют шину PCI. Многие материнские платы содержат даже встроенный чип, совместимый с Sound Blaster. Звуковые устройства могут быть также подсоединенены через USB, хотя мы не будем

рассматривать их.

Порты, используемые ISA-картой Sound Blaster это 0220-022F, хотя часто можно было выбрать базовый адрес 240, 260 или 280. Подобно этому, IRQ обычно выбирались из стандартного набора 2, 5, 7, или 10. По умолчанию использовалось IRQ 5. Также обычно можно было настраивать DMA каналы.

Как и для всех ISA-устройств, вы должны убедиться, что порты, IRQ и DMA каналы не конфликтуют с другими устройствами. Для дополнительной информации смотри предыдущий раздел [Настройки BIOS](#).

### MIDI порт (MPU-401)

Многие звуковые карты также имеют интерфейс для подключения устройства *MIDI* (от Musical Instrument Digital Interface -- Цифровой Интерфейс Музыкальных Инструментов). Вообще этот интерфейс эмулирует Roland MPU-401. Стандартные порты, используемые ISA-интерфейсом MPU-401 это 0200-020F.

Как и для всех ISA-устройств, вы должны убедиться, что порты, IRQ и DMA каналы не конфликтуют с другими устройствами. Для дополнительной информации смотри предыдущий раздел [Настройки BIOS](#).

### Настройка поддержки звука в Linux

Современные ядра 2.4 и 2.6 имеют поддержку звука для огромного числа разнообразных звуковых устройств, встроенную в ядро, обычно в качестве модуля. Как и для других устройств, мы можем использовать команду `pnpdump` для ISA-устройств, или команду `lspci` для PCI устройств, чтобы вывести информацию об устройстве. Листинг 14 содержит вывод команды `lspci` для звуковой системы от Intel (Intel sound system), встроенной в материнскую плату.

### Листинг 14. Использование lspci для отображения звуковых ресурсов

```
[root@lyrebird root]# lspci | grep aud
00:1f.5 Multimedia audio controller: Intel Corporation 82801DB/DBL/DBM
(ICH4/ICH4-L/ICH4-M) AC'97 Audio Controller (rev 01)
```

Модули ядра -- это предпочтительный способ для обеспечения поддержки различных устройств. Необходимо только загрузить модули, соответствующие устройствам реально присутствующим в системе, причем они могут выгружаться и подгружаться без перезагрузки системы. Для ядра версии 2.4 и более ранних, информация о конфигурации ядра хранится в `/etc/modules.conf`. Для ядер 2.6, система модулей ядра была изменена и теперь информация хранится в `/etc/modprobe.conf`. В любом случае, команда `lsmod` отформатирует содержимое `/proc/modules` и отобразит состояние загруженных модулей.

В Листинге 15 приведено содержимое `/etc/modprobe.conf` для ядра 2.6, а Листинг 16 содержит вывод команды `lsmod`, связанный со звуковыми устройствами этой системы.

### Листинг 15. Пример /etc/modprobe.conf (Ядро 2.6)

```
[root@attic4 ~]# cat /etc/modprobe.conf
alias eth0 e100
alias snd-card-0 snd-intel8x0
install snd-intel8x0 /sbin/modprobe --ignore-install snd-intel8x0 && \
/usr/sbin/alsactl restore >/dev/null 2>&1 || :
remove snd-intel8x0 { /usr/sbin/alsactl store >/dev/null 2>&1 || : ; }; \
```

```
/sbin/modprobe -r --ignore-remove snd-intel8x0
alias usb-controller ehci-hcd
alias usb-controller1 uhci-hcd
```

#### Листинг 16. Вывод команды lsmod, связанный со звуком (Ядро 2.6)

```
[root@attic4 ~]# lsmod |egrep '(snd)|(Module)'
Module           Size  Used by
Module           Size  Used by
snd_intel8x0      34689  1
snd_ac97_codec    75961  1 snd_intel8x0
snd_seq_dummy     3653   0
snd_seq_oss       37057  0
snd_seq_midi_event 9153   1 snd_seq_oss
snd_seq           62289  5 snd_seq_dummy,snd_seq_oss,snd_seq_midi_event
snd_seq_device    8781   3 snd_seq_dummy,snd_seq_oss,snd_seq
snd_pcm_oss       51185  0
snd_mixer_oss     17857  1 snd_pcm_oss
snd_pcm           100169  3 snd_intel8x0,snd_ac97_codec,snd_pcm_oss
snd_timer         33605   2 snd_seq,snd_pcm
snd               57157  11 snd_intel8x0,snd_ac97_codec,snd_seq_oss,
                  snd_seq,snd_seq_device,snd_pcm_oss,snd_mixer_oss,snd_pcm,snd_timer
soundcore          10913  1 snd
snd_page_alloc     9669   2 snd_intel8x0,snd_pcm
```

В Листинге 17 приведено содержимое /etc/modules.conf для ядра 2.4, а Листинг 18 содержит вывод команды lsmod, связанный со звуковыми устройствами этой системы. Заметьте, что файлы modules.conf и modprobe.conf схожи.

#### Листинг 17. Пример /etc/modules.conf (Ядро 2.4)

```
[root@lyrebird root]# cat /etc/modules.conf
alias eth0 e100
alias usb-controller usb-uhci
alias usb-controller1 ehci-hcd
alias sound-slot-0 i810_audio
post-install sound-slot-0 /bin/aumix-minimal -f /etc/ .
aumixrc -L >/dev/null 2>&1 || :
pre-remove sound-slot-0 /bin/aumix-minimal -f /etc/ .
aumixrc -S >/dev/null 2>&1 || :
```

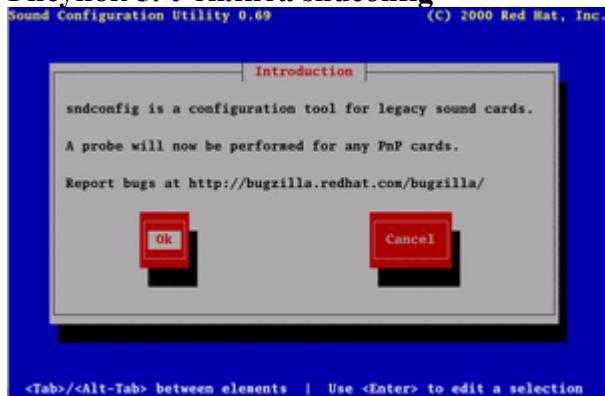
#### Листинг 18. Вывод команды lsmod, связанный со звуком (Ядро 2.4)

```
Module           Size  Used by      Not tainted
smbfs            43568  1 (autoclean)
i810_audio       28824  0 (autoclean)
ac97_codec       16840  0 (autoclean) [i810_audio]
soundcore        6436   2 (autoclean) [i810_audio]
st                30788  0 (autoclean) (unused)
```

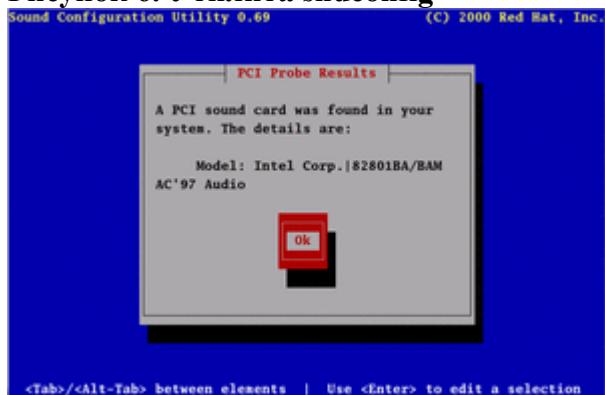
Поддержка звука на многих системах 2.4 и более ранних обеспечивается благодаря драйверам *Open Sound System (OSS) Free*. Сегодня многие системы используют драйвера

*Advanced Linux sound architecture* или *ALSA*. Утилита **sndconfig** была создана Red Hat для конфигурирования ISA PnP звуковых карт. Она также работает и с PCI звуковыми картами. Эта утилита может присутствовать и в системах, не использующих драйвера ALSA, хотя поддержка современных модулей сделала ее практически не нужной. Эта утилита опрашивает звуковую карту, воспроизведя речь Линуса Торвальдса (Linus Torvalds) в качестве теста, а затем обновляет файл /etc/modules.conf. Типичная операция показана на Рисунках 5 и 6.

**Рисунок 5. Утилита sndconfig**



**Рисунок 6. Утилита sndconfig**



| [предыдущая](#) | [следующая](#)

## Настройка SCSI-устройств

В этом разделе излагается материал по теме 1.101.4 для экзамена LPI 101 Администрирование для начинающих (LPIC-1). Рейтинг темы 1.

### Обзор SCSI

*Small Computer System Interface* (*Системный Интерфейс Малых Компьютеров*), больше известный как *SCSI*, это интерфейс, разработанный для соединения потоковых устройств, таких как ленточные и блочные устройства хранения типа дисков, CD-ROM или DVD приводов. Он также используется для других устройств, таких как сканеры и принтеры. *SCSI* произносится как "скэзи". *SCSI* был разработан для размещения нескольких устройств на однойшине. Одно устройство, называемое *контроллер* отвечает за управление шиной. *SCSI*-устройства могут быть как внутренними, так и внешними.

Имеется три главных версии стандартов *SCSI* от Американского Института Национальных Стандартов (American National Standards Institute -- ANSI).

### SCSI

это оригинальный стандарт (X3.131-1986), сейчас обычно называется *SCSI-1*. Он появился благодаря усилиям Shugart Associates в создании стандартного интерфейса

подключения дисковых устройств. Этот стандарт поддерживал до 8 устройств на одном кабеле. SCSI-1 использует пассивную оконечную схему [passive termination] (более подробно об этом далее). Это стандарт в настоящее время исчез, хотя SCSI-1 устройства могут все еще работать на современных SCSI кабелях, предполагающих соответствующую оконечную схему. Интерфейс данных был параллельным и 8-битным с максимальной скоростью передачи 5 МБ/с (Мегабайт/сек.). Стандарт SCSI был разработан для дисков, но был очень гибок и использовался для других устройств, преимущественно сканеров и медленных устройств, таких как Zip<sup>(TM)</sup>. FConnection использовал 50-ти жильный кабель, обычно Centronics, а позднее с 50-контактный D-shell, похожий на DB-25 RS-232 последовательного соединения,

### **SCSI-2**

был принят как стандарт ANSI X3.131-1994 в 1994. Эта версия удвоила скорость шины до 10МБ/с, а также ввела так называемую *широкую* или 16-битную передачу данных. 16-битная шина работая на скорости 10МБ/с могла передавать 20МБ/с данных. Для 8-битных или узких SCSI-устройств использовался 50-жильный кабель, а для новых широких устройств -- 68-жильный. Была также повышена плотность кабеля, что позволило использовать более миниатюрные и дешевые соединители. SCSI-2 также стандартизировал набор команд SCSI и ввел раздельную передачу сигнала (*differential signaling*) для улучшения качества на высоких скоростях. Позднее это было названо Передача сигналов с *High Voltage Differential* (*Высоковольтным разделением*) или *HVD*. HVD требует активной оконечной схемы. 8-битные и 16-битные устройства можно подсоединять к одному и тому же кабелю, если позаботиться о соответствующей оконечной схеме. SCSI-2 поддерживает до 16 устройств на одном кабеле из которых по крайне мере 8 могут быть узкими.

### **SCSI-3**

это набор стандартов, а не один стандарт. Это позволяет улучшать стандарты для быстроизменяющихся технологических областей, избегая необходимости пересмотра стандартов стабильной технологии. Итоговая архитектура определяется стандартом ANSI X3.270-1996, который также известен как *SCSI-3 Architecture Model* или *SAM*. Ранние стандарты SCSI теперь преобразованы в стандарты *SCSI Parallel Interface* (*Параллельный интерфейс SCSI*) или *SPI*. Скорость была вновь увеличена и современные 16-битные устройства способны передавать данные со скоростью до 320МБ/с при скорости шины 160МБ/с.

SCSI-3 ввел Оптоволоконные каналы SCSI (Fiber Channel SCSI) с поддержкой до 126 устройств на одну шину, поддерживающей соединение свыше 1ГБ/с или 2ГБ/с по оптоволоконным каналам на расстояния в несколько километров. Это помогает смягчить имеющиеся ограничения, связанные с использованием стандартных SCSI-кабелей. Другим важным нововведением было *Single Connector Attachment* или *SCA*, используемое только для широких (16-битных) устройств. SCA это 80-контактное соединительное звено, которое включало контакты от 68-контактного звена, а также питание и некоторые дополнительные контакты. SCA разрабатывалась для безопасного горячего подключения устройств в работающей системе и часто используется в устройствах объединенных в систему хранения *Redundant Array of Independent disks* (*Избыточный массив независимых дисков*) или *RAID*, а также в сетевых устройствах хранения и серверных стойках.

Выше мы упомянули *оконечную схему*, не объяснив что это такое. Электрические спецификации шины SCSI требуют, чтобы каждый конец шины был корректно завершен. Вы должны использовать соответствующий тип терминатора для вашей шины: пассивный, HVD или LVD. Если вы смешиваете широкие и узкие устройства, то будьте внимательны, поскольку терминатор для узких устройств может располагаться в месте отличном от

терминатора для широких устройств. Если контроллер работает только с внутренней шиной или только с внешней, то он обычно обеспечивается терминатором, автоматическим или настраиваемым через BIOS. Просмотрите руководство к вашему конкретному контроллеру. Если контроллер управляет обоими и внешним и внутренним сегментами, то обычно он не должен оборудоваться терминатором.

Некоторые устройства способны играть роль терминатора, что выставляется перемычками или переключателями. И вновь обратитесь к руководству вашего устройства. В противном случае завершение (*termination*) обычно реализуется оконечным блоком, подсоединяемым к кабелю. Какой бы тип терминатора вы не использовали, будьте очень осторожны если вы совместно используете и широкие и узкие устройства на однойшине, поскольку завершение для узких и широких устройств может оказаться в разных местах кабеля.

### **SCSI ID (Идентификация)**

Теперь вы можете удивиться как же система управляет таким количеством устройств на одном кабеле. Каждое устройство, включая контроллер имеет свой *ID*, выражаемый числом. Для узких (8-битных) SCSI, ID-номера находятся в диапазоне от 0 до 7. Широкие SCSI добавляют номера от 8 до 15. Узкие устройства могут использовать только номера от 0 до 7, в то время как широкие могут использовать номера от 0 до 15. Контроллеру обычно присваивается номер 7. ID устройств могут назначаться при помощи перемычек, переключателей или при помощи циферблата на устройстве, а также программно. Устройствам, использующим Single Connector Attachment (SCA), ID обычно присваиваются автоматически, поскольку эти устройства могут подключаться во время работы.

Устройства на SCSIшине имеют приоритет. Приоритет для узких устройств изменяется от 0 (наиизший) до 7 (наивысший), так что контроллер с адресом 7 имеет наивысший приоритет. Дополнительные ID для широких устройств имеют приоритет от 8 (наиизший) до 15 (наивысший), но 15 имеет приоритет меньше 0. Поэтому реальная шкала приоритета выглядит так 8, 9, 10, 11, 12, 13, 14, 15, 0, 1, 2, 3, 4, 5, 6, 7. Более медленные устройства и устройства, которые не могут допускать задержек (такие как записывающие CD или DVD приводы) должны иметь высокоприоритетные ID, чтобы гарантированно иметь значительные ресурсы.

Устройства, подобные RAID контроллерам могут выделять единственный ID для шины, но могут объединять несколько дисков. В добавок к ID, адресация SCSI допускает *a Logical Unit Number (Логическую Нумерацию Устройств)* или *LUN*. Ленточные и отдельные дисковые приводы или не сообщают LUN, или выдают LUN равный 0.

SCSI адаптер может поддерживать более одного кабеля или *канала*, а также в системе может быть несколько SCSI-адаптеров. Таким образом, полный ID устройства состоит из номера адаптера, номера канала, ID устройства и LUN.

Такие устройства как пишущие CD приводы использующие ide-scsi эмуляцию и USB устройства хранения также будут появляться словно имеют свой собственный адаптер.

### **Имена и файлы Linux для SCSI устройств**

Вернемся к разделу о BIOS когда мы обсуждали имена, присваиваемые Linux [IDE приводам](#), такие как /dev/hda и /dev/hdc. Для IDE контроллера, который может поддерживать один или два жестких диска это просто. Второй IDE привод на втором адаптере это всегда /dev/hdd, даже если еще одним диском является ведущий (primary) на первом адаптере (/dev/hda). Для SCSI ситуация становится более сложной, поскольку мы можем подсоединить к одному кабелю жесткие диски, ленточные устройства, CD и DVD приводы, а также другие устройства.

Linux присваивает имена устройствам по мере того, как они обнаруживаются во время загрузки. Поэтому первый жесткий диск на первом канале первого адаптера станет /dev/sda,

второй -- /dev/sdb, и так далее. Первое ленточное устройство будет /dev/st0, второе -- /dev/st1, и так далее. Первое CD-устройство станет /dev/sr0 или /dev/scd0, а второе -- /dev/sr1 или /dev/scd1. Устройствам, использующим эмуляцию SCSI, вроде USB устройств хранения и (вплоть до ядра 2.6) IDE CD или DVD приводам будут также выделяться имена в пространстве имен.

Хоть мы и не хотим полностью разбираться во всех сложностях со SCSI именованием, очень важно помнить, что эта нумерация производится заново при каждой перезагрузке. Если вы добавляете или удаляете жесткий диск SCSI, то все приводы выше него при последующей перезагрузке получат другие имена. То же происходит и с устройствами других типов. В другом учебнике этой серии мы более подробно изучим разбиение дисков, метки и файловые системы, но сейчас мы хотим предостеречь вас от одной вещи. Поскольку диски могут иметь до 15 разделов, каждый из которых имеет имя, связанное с именем устройства (например, /dev/sda1, /dev/sda2 и так далее до /dev/sda15), это может стать причиной путаницы, когда ваша система попытается смонтировать файловые системы. Очень тщательно планируйте добавление нового или удаление имеющегося SCSI устройства и по возможности вместо имен устройств используйте метки SCSI дисков.

Мы познакомились с файловой системой /proc в разделе [Настройки BIOS](#). Файловая система /proc также содержит информацию о SCSI устройствах. В Листинге 19 приведено содержимое /proc/scsi/scsi для системы с двумя SCSI устройствами: жестким диском с ID 0 и контроллером с ID 8.

### Листинг 19. /proc/scsi/scsi

```
[root@waratah root]# cat /proc/scsi/scsi
Attached devices:
Host: scsil0 Channel: 00 Id: 00 Lun: 00
  Vendor: IBM-PSG Model: DPSS-336950M F  Rev: S94S
  Type: Direct-Access                      ANSI SCSI revision: 03
Host: scsil0 Channel: 00 Id: 08 Lun: 00
  Vendor: IBM      Model: YGLv3 S2        Rev: 0
  Type: Processor                     ANSI SCSI revision: 02
```

Если вы хотите узнать какое реальное устройство соответствует скажем /dev/sda, то вы можете использовать команду `scsi_info`. Листинг 20 подтверждает, что наш первый (и единственный) SCSI жесткий диск это /dev/sda.

### Листинг 20. Команда scsi\_info

```
[root@waratah root]# scsi_info /dev/sda
SCSI_ID="0,0,0"
MODEL="IBM-PSG DPSS-336950M F"
FW_REV="S94S"
```

Однако, заметьте, что некоторые системы, такие как Fedora Core 2, не содержат команды `scsi_info` (являющейся частью пакета `kernel-pcmcia-cs`).

Более поздние системы используют драйвер *SCSI Generic* или *sg* (Универсальный драйвер). При использовании *sg* драйвера вы сможете найти дополнительную информацию в ветке /proc/scsi/sg вашей файловой системы. Вы также будете иметь устройства вроде /dev/sg0, /dev/sg1, /dev/sg2 и так далее. Универсальные устройства обычно соответствуют другим

типам устройств, типа жесткого диска как /dev/sda или ленты вроде /dev/st0.

Пакет sg3\_utils содержит несколько утилит для манипулирования и определения параметров подсистем SCSI. В действительности, команда **sg\_map** выводит таблицу соответствий (map) sg-имен и других имен устройств если они существуют. Заметьте, что сканеры не имеют другого имени, только универсальное. Листинг 21 содержит результат выполнения sg\_map в системе с оптическим диском IDE, который использует SCSI эмуляцию и двумя USB-дисками.

### Листинг 21. Команда sg\_map

```
[root@lyrebird root]# sg_map  
/dev/sg0  /dev/scd0  
/dev/sg1  /dev/sda  
/dev/sg2  /dev/sdb
```

Для sg, соответствующая scsi\_info утилита называется **sginfo**. Вы можете использовать либо универсальное имя устройства, либо более знакомое имя от sginfo. Листинг 22 содержит вывод команды sginfo для трех устройств Листинга 21. Заметьте, что sginfo не предоставила информацию о /dev/sg1, хотя как видно из листинга, команда scsi\_info показывает его как USB-диск. В данном случае устройство было извлечено из системы. Информация о нем осталась (и ее можно найти в /proc/scsi/scsi). Команда sginfo для получения информации опрашивает устройства, в то время как scsi\_info использует связанную информацию. Поэтому sginfo должна выполняться из под root, а scsi\_info не требует этого, хотя не-root пользователям может потребоваться указать полный путь /sbin/scsi\_info.

### Листинг 22. Команда sginfo

```
[root@lyrebird root]# sginfo /dev/scd0  
INQUIRY response (cmd: 0x12)  
-----  
Device Type          5  
Vendor:             SONY  
Product:            DVD RW DRU-700A  
Revision level:     VY08  
  
[root@lyrebird root]# sginfo /dev/sg1  
INQUIRY response (cmd: 0x12)  
-----  
Device Type          0  
Vendor:  
Product:  
Revision level:  
  
[root@lyrebird root]# sginfo /dev/sg2  
INQUIRY response (cmd: 0x12)  
-----  
Device Type          0  
Vendor:              WD  
Product:             2500JB External  
Revision level:     0411  
  
[root@lyrebird root]# scsi_info /dev/sg1  
SCSI_ID="0,0,0"  
MODEL=" USB DISK 12X"
```

FW\_REV="2.00"

## **SCSI BIOS и последовательность загрузки.**

В то время как SCSI является стандартом для большинства серверов, многие настольные компьютеры и ноутбуки обычно не поддерживают SCSI. Такие системы обычно загружаются с флоппи дисков, CD или DVD приводов или первого жесткого диска IDE в компьютере. Порядок загрузки обычно настраивается в окне настройки BIOS, так как мы видели в разделе [Настройки BIOS](#), и иногда динамически при помощи нажатия клавиш или их комбинаций во время старта системы.

Загрузочная спецификация BIOS (смотри [Ресурсы](#)) определяет метод добавления карт, таких как SCSI-карты, выводит сообщение при включении и вызывает BIOS карты для ее конфигурирования. SCSI карты обычно используют ее для настройки подсистем SCSI, управляемых картой. Например, карта Adaptec AHA-2930U2 выводит сообщение

Press <Ctrl><A> for SCSISelect (TM) Utility!

(Нажмите Ctrl+A для запуска утилиты SCSISelect), позволяющее пользователю, нажав одновременно клавиши ctrl и A, войти в BIOS адаптера. Другие карты имеют сходную процедуру входа в BIOS карты для ее настройки.

Оказавшись в BIOS карты, вы увидите экраны страницы, при помощи которых часто можно настроить адрес SCSI контроллера (обычно 7), загрузочное SCSI устройство (обычно ID 0), скорость шины и должен ли контроллер обеспечивать оконечную схему или нет. Некоторые старые карты могут потребовать, чтобы загрузочное устройство имело ID 0, но большинство современных карт позволяют вам выбрать любое устройство. Вы можете, и вероятно захотите, настроить и другие параметры, такие как возможность форматирования жесткого диска. За потребностями обратитесь к документации производителя вашей карты. После настройки поведения шины SCSI вы обычно должны еще указать BIOS ПК на необходимость загрузки со SCSI диска, а не IDE. Проконсультируйтесь с руководством к вашей системе для определения можете ли вы загрузиться с не-IDE диска и как это можно настроить.

| [предыдущая](#) | [следующая](#)

## **Платы расширения ПК**

В этом разделе излагается материал по теме 1.101.5 для экзамена LPI 101  
Администрирование для начинающих (LPIC-1). Рейтинг темы 1.

Материал, который вы должны знать для данного раздела, был изложен при обсуждении [Настроек BIOS](#). Вам следует вновь просмотреть информацию о DMA, IRQ, портах и различных типах шин и адаптеров в разделе [Шины, порты, IRQ, и DMA](#) чтобы понять содержимое файлов /proc/dma, /proc/interrupts, и /proc/ioports, а также как их использовать для определения конфликтов. Просмотрите материал о /proc/pci и команде lspci. Также повторите раздел [Plug and play](#), чтобы знать об ISA и Plug and Play картах. Там же вы найдете информацию о isapnp и pnpdump.

## **Коммуникационные устройства**

В этом разделе рассматривается материал по теме 1.101.6 экзамена LPI 101  
Администрирование для начинающих (LPIC-1). Рейтинг темы 1.

В этом разделе рассматриваются разнообразные коммуникационные устройства, включая модемы ISDN адаптеры и DSL коммутаторы. Материал данного раздела разбит на две

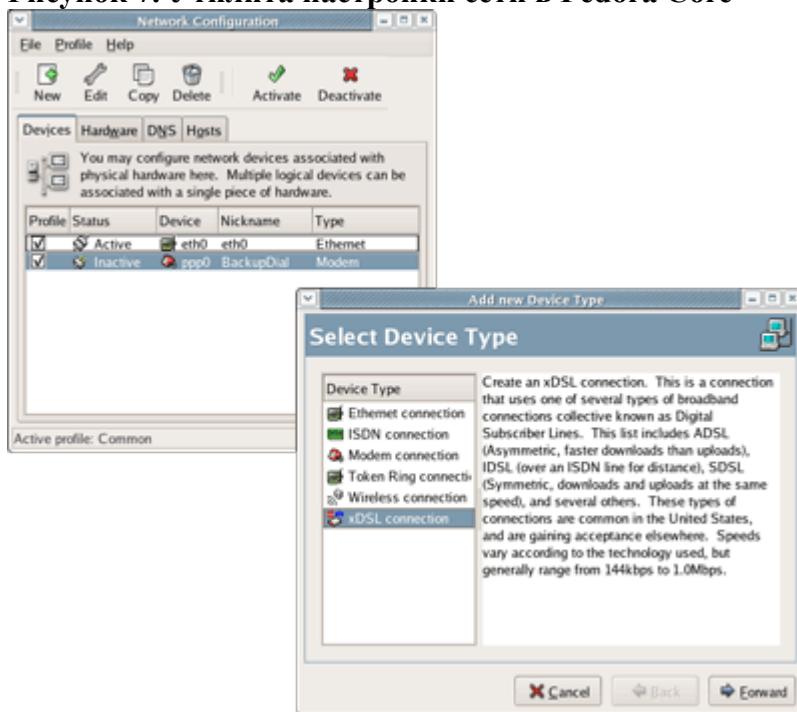
основные категории:

1. Выбор и установка коммуникационного устройства, и
2. Связь с устройством

Выбор коммуникационного устройства подобен выбору любого другого устройства для вашей системы, то есть оно должно соответствовать вашему типу шины (PCI or ISA) и необходима поддежка устройства в Linux. Вам следует просмотреть обсуждение DMA, IRQ, портов и различных типов шин и адаптеров в разделе [Шины, порты, IRQ, и DMA](#) чтобы понимать содержимое файлов /proc/dma, /proc/interrupts, и /proc/ioports, а также как использовать их для определения конфликтов. Просмотреть материал о /proc/pci и команде lspci. Также повторите раздел [Plug and play](#), чтобы знать об ISA и Plug and Play картах. Там же вы найдете информацию о isapnp и pnpdump.

Ядро Linux с каждой версией поддерживает все больше и больше устройств, так что, во-первых, посмотрите есть ли поддержка нужного устройства в уже используемом вами дистрибутиве. Если поддержка уже есть, то в вашем дистрибутиве уже может быть утилита, помогающая в настройке устройства. На Рисунке 7 показан инструмент настройки сети в Fedora Core 4. Вы можете видеть, что соединение ethernet уже настроено (и активно), а также настроено модемное соединение для резервного копирования по телефонной линии с использованием PPP. Система уже поддерживает подключение ISDN, Token Ring (кольцевые ЛВС), беспроводное и xDSL соединения.

**Рисунок 7. Утилита настройки сети в Fedora Core**



Если вы собираетесь установить драйвер для коммуникационного устройства, прежде всего проверьте является ли требуемый драйвер частью вашего дистрибутива. Если является, но еще не установлен, то установите его. В противном случае вам следует попытаться найти пакет для вашей системы с уже скомпилированным драйвером. И только если ничего не нашли вы можете собрать драйвер из исходных текстов самостоятельно. Мы рассмотрим построение пакетов в учебнике для экзамена LPI 101, Тема 102. (Смотри [Ресурсы](#)).

Для ISDN соединения вам также понадобиться синхронный драйвер PPP, поскольку обычный, используемый для работы с асинхронными модемами, разработан для режима посимвольной передачи, а не для блочного. Как мы уже указывали в разделе о модемах, мы

рассмотрим настройку соединений более подробно в учебнике для экзамена LPI 102

DSL соединения могут быть одного из нескольких типов. Некоторые предоставляют ethernet порт, связанный с сетью провайдера. Аутентификация в этом случае обычно производится с использованием ethernet MAC-адреса вашего компьютера. Если вы подсоединяете маршрутизатор (или другой компьютер) к DSL модему, то вам может понадобиться изменить Mac-адрес компьютера, чтобы добиться работоспособности соединения. Проще, если провайдер использует *Point-to-Point Protocol over Ethernet* или *PPPoE*. В этом случае вам выдается имя пользователя и пароль для использования при соединении. В таком случае, если вы используете маршрутизатор, вы обычно настраиваете соединение на нём, а ваш компьютер просто использует стандартное ethernet-подключение. Реже вам потребуется использовать *PPPoA* или *PPP* через *ATM* соединение.

Беспроводные соединения могут потребовать от вас имя сети, к которой вы подключаетесь. Это называется *Service Set Identifier* (*Сервис установки идентификатора*) или *SSID*. Если в сети используется шифрование, такое как *Wired Equivalent Privacy* или *WEP* или *WiFi Protected Access* или *WPA*, то вам понадобится соответствующим образом настроить соединение.

| [предыдущая](#) | [следующая](#)

## USB устройства

В этом разделе освещается материал темы 1.101.7 для экзамена LPI 101 Администрирование для начинающих (LPIC-1). Рейтинг темы 1.

### Обзор USB

В этом разделе мы рассмотрим поддержку *Universal Serial Bus* (*Универсальной последовательной шины*) или *USB* устройств в Linux. USB была разработана консорциумом компаний с целью предоставить единственную, простую шину для подключения периферии. В разделе [Настройки BIOS](#), мы рассмотрели аспекты управления портами, IRQ и DMA ресурсами для машин с шиной ISA. Дизайн USB позволяет устройствам подключаться на лету и использовать стандартные гнезда для подключения устройств. USB-устройства включают клавиатуры, "мыши", принтеры, сканеры, жесткие диски, flash-драйвы, камеры, модемы, сетевые адAPTERы и колонки. Список продолжает расти. Имеющаяся в Linux поддержка довольно всеобъемлюща, хотя некоторые устройства требуют специальных драйверов а другие, преимущественно принтеры, могут не поддерживаться или могут поддерживаться частично.

Компьютерные системы могут обладать одним или более *контроллерами* или *хабами*, к которым могут подключаться USB устройство или другой (внешний) хаб. Хаб может поддерживать до 7 устройств, некоторые или каждое из которых могут иметь дополнительные хабы. Хаб внутри системного блока называется *root hub* (*корневой хаб*). Каждая такая звездоподобная топология может поддерживать до 127 хабов или устройств.

**Замечание:** Часто, мы говорим *USB port*, подразумевая возможность поддержки USB в компьютере и гнездо для подключения (сравните с последовательным или параллельным портами), а не внутренний адрес порта, используемый устройством.

USB системы являются многослойными.

1. Слой *Bus Interface* (*Интерфейс шины*) обеспечивает физическую, сигнальную, и пакетную связь между хостом и устройствами, обеспечивая передачу данных между хостом и устройствами.
2. Слой *Device* (*устройство*) используется системным ПО для базовых USB-операций с устройством посредством шины. Это позволяет хосту определить характеристики устройства, включая класс устройства, имя производителя, имя устройства,

требования к напряжению и многие другие функции наподобие скорости устройства и уровня поддержки USB.

3. Слой *Function* (*Функция*) предоставляет дополнительные возможности, специфичные для устройства. Соответствие слоев хоста и ПО устройства позволяет выполнять функции, присущие устройству.

Ранние спецификации USB (1.0 и 1.1) поддерживали скорость до 12Мб/с (Мегабит в секунду). Устройства, соответствующие этим спецификациям соответственно являются низкоскоростными, это принтеры, "мыши", клавиатуры, сканеры и модемы. Новая спецификация USB 2.0 поддерживает скорость до 480Мб/с, что сравнимо с жестким диском и внешним CD/DVD приводом. Некоторые USB 2.0 устройства имеют обратную совместимость, что позволяет использовать их в старых системах, хотя и не все скоростные устройства обладают ею. Если ваш компьютер не имеет встроенной поддержки USB 2.0, то вы можете воспользоваться PCI картой (или PC картой для ноутбука), предоставляющей один или несколько портов USB 2.0.

USB-кабель довольно тонок и содержит 4 провода: два для передачи сигнала плюс силовой и нейтральный. Разъем, подключаемый к хабу, имеет плоский прямоугольный штеккер (называемый А штеккер), а разъем, подключаемый к устройству или к хабу более низкого уровня имеет штеккер больше похожий на квадрат (называемый В штеккер). Несколько отличается существующий mini-B штеккер, для подсоединения к компьютеру небольших устройств вроде фотоаппарата. USB устройства и хабы могут получать питание по USBшине или могут иметь собственные источники питания.

### Модуль поддержки USB в Linux

USB в настоящее время полностью поддерживается в Linux. Большая часть изменений проявилась в ветке ядра 2.6. Многое было перенесено из ядер 2.4, какая-то поддержка имеется даже в ядрах 2.2. Linux поддерживает как USB 2.0, так и ранние спецификации. Ввиду подключения на лету (горячего подключения), заложенной в самой природе USB, поддержка обычно производится посредством модулей ядра, которые могут загружаться или выгружаться по необходимости. В этом учебнике мы предполагаем, что все модули, которые вам необходимы для вашего дистрибутива или доступны, или уже установлены. Если вам необходимо скомпилировать свое собственное ядро, то обратитесь к теме 201 экзамена LPI 201

После того, как вы убедитесь, что ваш компьютер имеет USB порты, вы можете посмотреть что обнаружила система Linux, используя команду `lspci`, как показано в Листинге 23. Мы отфильтровали вывод, чтобы отобразить только устройства, соответствующие USB.

### Листинг 23. Вывод команды `lspci` для USB устройств

```
[root@lyrebird root]# lspci | grep -i usb
00:1d.0 USB Controller: Intel Corporation 82801DB/DBL/DBM
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #1 (rev 01)
00:1d.1 USB Controller: Intel Corporation 82801DB/DBL/DBM
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #2 (rev 01)
00:1d.2 USB Controller: Intel Corporation 82801DB/DBL/DBM
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #3 (rev 01)
00:1d.7 USB Controller: Intel Corporation 82801DB/DBM
    (ICH4/ICH4-M) USB2 EHCI Controller (rev 01)
```

Вы можете видеть, что в этой системе имеется четыре USB контроллера. Поля UHCI и EHCI показывают модули-драйвера, необходимые для поддержки контроллера. Корректный драйвер USB 1.1 зависит от чипсета, используемого в вашем контроллере. USB 2.0 требует

EHCI драйвер плюс USB 1.1 драйвер. Смотри Таблицу 5.

**Таблица 5. Драйвера USB в Linux**

*Таблица 5. Драйвера USB в Linux*

Драйвер	Чипсет
EHCI	USB 2.0 Поддерживает - требует один UHCI, OHCI или JE
UHCI	Intel и VIA Чипсеты
JE	Это альтернатива UHCI для ядер 2.4. Если UHCI не работает, и у вас Intel или VIA чипсет, то попробуйте JE
OHCI	Compaq, большинство PowerMacs, iMacs, и PowerBooks, OPTi, SiS, ALi

Мы рассматривали команду `lsmod` и файлы конфигурации модулей /etc/modules.conf (ядро 2.4) и /etc/modprobe.conf (ядро 2.6) ранее при обсуждении поддержки звука. В Листинге 24 показано несколько модулей, связанных с USB устройствами, которые загружаются в той же системе, что использовалась и в Листинге 23. В этой системе есть USB-мышь

#### **Листинг 24. Использование lsmod для отображения загруженных USB модулей**

```
[root@lyrebird root]# lsmod | egrep 'usb|hci|hid|mouse|Module'
Module           Size  Used by
usbserial        23420  0  (autoclean) (unused)
mousedev         5524   1
hid              22244  0  (unused)
input             5888   0  [keybdev mousedev hid]
ehci-hcd         20008  0  (unused)
usb-uhci          25740  0  (unused)
usbcore           77376  1  [usbserial hid ehci-hcd usb-uhci]
```

Заметим однако, что модуль usbcore используется всеми остальными USB модулями наряду с модулем hid (human interface device -- интерфейс пользовательских устройств).

#### **Отображение информации о USB**

Итак, теперь мы кое-что знаем о модулях поддержки USB, как мы можем обнаружить, что USB Устройство было подсоединенено к компьютеру? Эта информация может быть обнаружена в системе файлов /proc/bus/usb. Файл /proc/bus/usb/devices содержит сводную информацию о подключенных в настоящее время USB устройствах. Частичный список для нашей системы приведен в Листинге 25.

#### **Листинг 25. Часть содержимого /proc/bus/usb/devices**

```
[root@lyrebird root]# cat /proc/bus/usb/devices
T: Bus=04 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=480 MxCh= 6
B: Alloc= 0/800 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 2.00 Cls=09(hub ) Sub=00 Prot=01 MxPS= 8 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 2.04
S: Manufacturer=Linux 2.4.21-32.0.1.EL ehci-hcd
S: Product=Intel Corp. 82801DB USB2
S: SerialNumber=00:1d.7
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
```

```

E: Ad=81(I) Atr=03(Int.) MxPS= 2 Ivl=256ms
T: Bus=03 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 1.00 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 0.00
S: Product=USB UHCI Root Hub
S: SerialNumber=1840
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 8 Ivl=255ms

```

Выделенное *Spd=480* нами выше соответствует шине USB 2.0, а *Spd=12* -- устройствам USB 1.1 (или возможно USB 1.0). Далее в списке видно, что наша мышь имеет *Spd=1.5*. Полтора мегабита в секунду вполне достаточно для мыши.

Что касается других пунктов, которые можно увидеть в системе файлов /proc, то, возможно, вам будет приятно узнать, что имеется команда **lsusb** помогающая вам отображать эту информацию. В частности, вы можете получить древовидное изображение ваших USB устройств, используя опцию **-t**. Это отобразит иерархию подключения. Вы можете использовать опцию **-d** для вывода информации о конкретном устройстве, после того, как получите результат от системы с использованием опции **-t**. Опция **-v** генерирует подробный вывод с интерпретацией многих полей, что мы могли вывидеть в Листинге 25. Для Листинга 26, мы подсоединили внешний хаб, цифровую камеру Nikon, USB-брелок и внешний жесткий диск USB 2.0, а затем выполнили команду.

## Листинг 26. Использование команды lsusb

```

[root@lyrebird root]# lsusb -t
Bus# 4
`-Dev# 1 Vendor 0x0000 Product 0x0000
  |-Dev# 2 Vendor 0x0409 Product 0x0059
    |-Dev# 8 Vendor 0x04b0 Product 0x0108
    |-Dev# 4 Vendor 0xd7d Product 0x1400
    `-Dev# 7 Vendor 0x1058 Product 0x0401
  `-Dev# 3 Vendor 0x07d0 Product 0x1202
Bus# 3
`-Dev# 1 Vendor 0x0000 Product 0x0000
Bus# 2
`-Dev# 1 Vendor 0x0000 Product 0x0000
Bus# 1
`-Dev# 1 Vendor 0x0000 Product 0x0000
  `-Dev# 2 Vendor 0x1241 Product 0x1111
[root@lyrebird root]# lsusb -d 0x0409:0x0059
Bus 004 Device 002: ID 0409:0059 NEC Corp. HighSpeed Hub
[root@lyrebird root]# lsusb -d 0x04b0:0x0108
Bus 004 Device 008: ID 04b0:0108 Nikon Corp. Coolpix 2500
[root@lyrebird root]# lsusb -d 0xd7d:0x1400
Bus 004 Device 004: ID 0d7d:1400 Phison Electronics Corp.
[root@lyrebird root]# lsusb -d 0x1058:0x0401
Bus 004 Device 007: ID 1058:0401 Western Digital Technologies, Inc.
[root@lyrebird root]# lsusb -d 0x07d0:0x1202
Bus 004 Device 003: ID 07d0:1202 Dazzle
[root@lyrebird root]# lsusb -d 0x1241:0x1111
Bus 001 Device 002: ID 1241:1111 Belkin Mouse
[root@lyrebird root]#

```

В Листинге 27 приведена часть подробного вывода после выполнения команды `lsusb`. Это часть для USB-брелока. Заметьте, что устройство сообщает свои максимальные требования к питанию (200mA). Отметим также, что устройство может быть обнаружено как SCSI устройство. Используя команду `dmesg` или `fdisk -l` вы можете выяснить, какое SCSI устройство соответствует этому устройству. Большинство камер оборудованы USB портами, а также устройствами чтения карт (картридерами). Флэш-устройства и жесткие диски причисляемые к классу устройств хранения управляются в Linux как SCSI устройства. Многие камеры идут с программами под Windows, помогающими загружать картинки с камеры. В Linux вы можете просто смонтировать SCSI устройство, представляющее камеру и скопировать фотографии на свой жесткий диск, где вы сможете редактировать их в таких программах, как GNU Image Manipulation Program [GNU Программа Манипулирования Изображениями] (the GIMP). Затем вы можете стереть файлы с карты памяти или записать файлы на нее из Linux, используя камеру как вычурную замену флоппи-диску.

### Листинг 27. Подробный вывод (его часть) команды

```
[root@lyrebird root]# lsusb -vd 0x0d7d:0x1400

Bus 004 Device 004: ID 0d7d:1400 Phison Electronics Corp.
Device Descriptor:
  bLength          18
  bDescriptorType   1
  bcdUSB         2.00
  bDeviceClass      0 (Defined at Interface level)
  bDeviceSubClass    0
  bDeviceProtocol     0
  bMaxPacketSize0     64
  idVendor        0x0d7d Phison Electronics Corp.
  idProduct        0x1400
  bcdDevice        0.02
  iManufacturer       1
  iProduct          2 USB DISK 12X
  iSerial           3 0743112A0083
  bNumConfigurations  1
Configuration Descriptor:
  bLength          9
  bDescriptorType   2
  wTotalLength      32
  bNumInterfaces     1
  bConfigurationValue  1
  iConfiguration      0
  bmAttributes       0x80
  MaxPower          200mA
Interface Descriptor:
  bLength          9
  bDescriptorType   4
  bInterfaceNumber    0
  bAlternateSetting    0
  bNumEndpoints      2
  bInterfaceClass     8 Mass Storage
  bInterfaceSubClass   6 SCSI
  bInterfaceProtocol  80 Bulk (Zip)
  iInterface          0
  ...
...
```

Еще один кусочек информации, который мы можем извлечь о шине и ID наших USB

устройств из Листинга 26, это способ определения какие модули требуются для конкретного устройства. Приведем пару примеров в Листинге 28.

### Листинг 28. Подробный вывод (его часть) команды lsusb

```
[root@lyrebird root]# usbmodules --device /proc/bus/usb/004/003  
usb-storage  
[root@lyrebird root]# usbmodules --device /proc/bus/usb/004/007  
usb-storage  
hid
```

### Подключение на лету (Горячее подключение)

Есть две команды, при помощи которых ваша система может использоваться для управления подключением USB устройств на лету: *usbmgr* и *hotplug*. В зависимости от того, какую из них вы будете использовать файлы настроек можно обнаружить в каталогах */etc/usbmgr* или */etc/hotplug* соответственно. Новые системы скорее всего уже имеют поддержку подключения на лету.

Подключение на лету USB устройств (а также PC карт) подразумевает, что пользователь подключает устройство в то время, как система уже работает. При этом система должна :

- Определить тип устройства, найти драйвер и запустить его
- Связать драйвер с устройством
- Уведомить другие подсистемы об устройстве. Что позволяет дискам смонтироваться или, например, добавиться очереди для печати.

| [предыдущая](#) | [следующая](#)

## Ресурсы

### Научиться

- [Оригинал учебника, на английском языке](#)
- Обзор в целом [Серия учебников для экзаменов LPI](#) на developerWorks для изучения основ Linux и подготовки к сертификации по системному администрированию.
- В [Программе LPIC](#), можно найти список заданий, примерные вопросы и детальные программы для трех уровней сертификации по системному администрированию Профессионального Института Linux (Linux Professional Institute).
- Исчерпывающую историю жестких дисков смотри в [Справочнике \(Reference Guide\) - Жесткие диски \(Hard Disk Drives\)](#). Раздел [Интерфейсы и настройки жестких дисков \(Hard Disk Interfaces and Configuration\)](#) включает информацию о SCSI, а также сравнение интерфейсов IDE/ATA и SCSI.
- [Проект The Linux documentation project](#) содержит много полезной документации о Linux, включая:
  - [Большие диски HOWTO \(Large Disk HOWTO\)](#) о геометрии дисков, предел в 1024 цилиндра, и другие дисковые ограничения.
  - [Linux 2.4 подсистема SCSI HOWTO](#), описывает SCSI в Linux, включая именование устройств.
  - [Linux SCSI Generic \(sg\) HOWTO](#) о новом общем SCSI-драйвере и утилитах в Linux
  - [Руководство Администратора Сети](#) об организации сети в Linux
  - [Linux Networking-HOWTO](#) о SLIP, CSLIP, и PPP

- [Linux PPP HOWTO](#) об настройке PPP в Linux
- Другие ресурсы для Linux-разработчиков можно найти в [Linux-разделе developerWorks](#).

## **Получить продукты и технологии**

- Создайте свой следующий проект для Linux с использованием [trial-версий ПО от IBM](#), доступных для загрузки напрямую с developerWorks.

| [предыдущая](#) |

# Учебник для экзамена LPI 101: Установка Linux и управление пакетами

*Администрирование для начинающих (LPIC-1) тема 102*

Ян (Ian) Шилдс (Shields), Старший программист, EMC

**Описание:** В этом учебнике, Ян Шилдс продолжает готовить вас к сдаче экзамена Профессионального Института Linux (Linux Professional Institute®) LPI 101

Администрирование для начинающих (LPIC-1). В этом втором из пяти учебников, Ян расскажет вам об установке Linux™ и управлении пакетами. К концу этого учебника вы узнаете, как Linux использует разделы, как Linux загружается, как устанавливать пакеты программ и управлять ими.

[Больше статей из этой серии](#)

**Дата:** 09.09.2005

**Уровень сложности:** простой

## Перед тем как начать

Узнайте чему вы сможете научиться в этом учебнике и как извлечь из него максимальную пользу.

## Об этой серии учебников

Linux Professional Institute Профессиональный институт Linux (LPI) производит сертификацию системных администраторов Linux для начинающих и специалистов. Чтобы пройти каждый из уровней сертификации необходимо сдать два LPI-экзамена. Перед тем, как сдавать экзамены, изучите [учебники для подготовки к LPI-экзаменам](#) на developerWorks, чтобы подготовиться к каждой теме экзаменов.

Следующие пять учебников помогут вам подготовиться к первым двум LPI-экзаменам по системному администрированию начального уровня: экзамен LPI 101. Готовится следующая серия учебников для подготовки к следующему экзамену начального уровня: экзамен LPI 102. Оба экзамена LPI 101 и LPI 102 необходимы для сертификации начального уровня. Сертификация начального уровня также известна как Сертификация уровня 1. Для прохождения первого уровня вы должны уметь:

- Работать в командной строке Linux
- Выполнять простые задачи по поддержке: помогать пользователям, добавлять пользователей в большие системы, резервировать данные и восстанавливать их из резервной копии, а также выключать и перезагружать систему.
- Устанавливать и настраивать рабочую станцию (включая X), а также подсоединять ее в локальную сеть, или подключать отдельно стоящий компьютер к сети Интернет через модем.

*Таблица 1. Экзамен LPI 101: Учебники и темы*

Учебник	Тема	Краткое содержание
<u><a href="#">Учебник для экзамена LPI 101 (тема 101): Аппаратное обеспечение и архитектура</a></u>	101	Обучает конфигурированию ваших аппаратных ресурсов в Linux. К концу этого учебника, вы узнаете как Linux конфигурирует устройства, обнаруженные на современном компьютере и где искать решения возникших проблем.

Учебник для Тема (Настоящий учебник) Представляет собой введение в установку экзамена LPI 101:102 Linux и управление пакетами. К концу этого учебника вы Установка Linux узнаете как Linux использует разделы жесткого диска, как Linux и управление загружается и как устанавливать и управлять пакетами пакетами программного обеспечения.

Учебник для Тема Скоро выйдет!  
экзамена LPI 101:103

GNU и команды  
UNIX

Учебник для Тема Скоро выйдет!  
экзамена LPI 104:104

Linux, Файловые  
системы и FHS

Учебник для Тема Скоро выйдет!  
экзамена LPI 110:110

Система X  
Window

Каждая тема и подтема имеют свой рейтинг, отражающий их важность.

## Об этом учебнике

Добро пожаловать в "Установку Linux и управление пакетами", второй из пяти учебников, разработанных для подготовки к экзамену LPI 101. В этом учебнике вы узнаете о следующих аспектах установки Linux и управления пакетами:

Таблица 2. Экзамен LPI 101, тема 102: Подтемы и рейтинг

Подтема	Рейтинг	Краткое содержание
1.102.1 Разработка структур жесткого диска	5	Вы узнаете о том как создать структуру разделов для системы Linux, включая размещение файловых систем или swap-пространства на отдельных разделах или дисках, подгонку структуры к нуждам используемой системы. Вы узнаете как разместить /boot на разделе так, чтобы это соответствовало требованиям BIOS к загрузке.
1.102.2 Установка менеджера загрузки	1	Вы узнаете как выбрать, установить и настроить менеджер загрузки, как обеспечить загрузку из разных мест и настроить опции загрузки для восстановления данных (например, используя загрузку с дискеты).
1.102.3 Сборка и установка программ из исходных текстов	5	Вы научитесь компилировать и устанавливать программы из исходных текстов. Вы узнаете как распаковать файлы исходных текстов и настраивать Makefile, например изменить пути или добавить дополнительные каталоги include.
1.102.4 Управление общими библиотеками	3	Вы научитесь определять общие библиотеки, от которых зависят программы и устанавливать их при необходимости. Вы также узнаете где хранятся системные библиотеки.
1.102.5 Использование управления пакетами от Debian	8	Вы научитесь управлять пакетами, используя менеджер пакетов от Debian. Вы будете использовать командную строку и интерактивные инструменты для установки, обновления или удаления пакетов, а также для поиска пакетов, содержащих специфичные файлы и программное обеспечение. Вы научитесь извлекать информацию о пакете, такую как версию, содержимое, зависимости, целостность пакета и статус установки. Вы узнаете как сделать это для

1.102.6 Использование Управления пакетами от Red Hat (RPM)	8	установленных и не установленных пакетов. Вы узнаете об использовании управления пакетами в дистрибутивах Linux, использующих RPM-пакеты. Вы научитесь устанавливать, переустанавливать, обновлять и удалять пакеты, также как узнавать об их статусе и версии. Вы узнаете как определить информацию о пакете, такую как версия, статус, зависимости, целостность и подписи. Вы узнаете как определить какие файлы находятся в пакете, а также как найти пакет со специфичными файлами.
--	---	--

## Предварительные замечания

Для того, чтобы извлечь из этого учебника максимум вы должны знать основы Linux и иметь рабочую версию системы Linux, в которой вы сможете выполнять команды, приведенные в данном учебнике. Вы также должны понимать как использовать BIOS для жестких дисков, что описано в первом учебнике данной серии "["Учебник для экзамена LPI 101 \(тема 101\): Аппаратное обеспечение и архитектура."](#)"

## Схема жесткого диска

В этом разделе излагается материал по теме 1.102.1 для экзамена LPI 101  
Администрирование для начинающих (LPIC-1). Рейтинг темы 5.

В этом разделе показано как разместить файловые системы Linux на ваших жестких дисках, что расширит ваши знания о жестких дисках, полученные в первом учебнике данной серии "["Учебник для экзамена LPI 101 \(тема 101\): Аппаратное обеспечение и архитектура."](#)"

Следующий учебник, "Учебник для экзамена LPI 101 (тема 104): Устройства, файловые системы Linux, и FHS" еще больше углубит ваши знания о файловых системах и инструментах создания разделов различного типа.

## Обзор файловых систем

Файловая система Linux состоит из *файлов*, которые размещаются на диске или другом *блочном устройстве хранения в каталогах*. Также как во многих других системах, каталоги в Linux могут содержать другие каталоги, которые называют *подкаталогами*. В отличие от такой системы как Microsoft® Windows® с концепцией отдельных файловых систем для каждой буквы диска (A:, C: и так далее.), файловая система Linux является одним деревом с каталогом / в качестве *корневого каталога (root)*.

Вы можете удивиться почему же схема жесткого диска важна, если файловая система это одно большое дерево?! Дело в том, что каждое блочное устройство, такое как раздел жесткого диска, CD-ROM или дискета на самом деле уже имеют свою файловую систему. Вы создаете отдельную ветку файловой системы путем *монтирования (mounting)* файловых систем других устройств в точку дерева, называемой *точкой монтирования (mount point)*.

Обычно вы начинаете процесс монтирования монтированием файловой системы одного из разделов жесткого диска как /. Вы можете смонтировать другие разделы жесткого диска как /boot, /tmp или /home. Например, вы можете монтировать файловую систему дискеты (floppy-диска) как /mnt/floppy, и файловую систему CD-ROM как /media/cdrom1. Вы можете также монтировать файлы других компьютеров, используя сетевые файловые системы, такие как NFS. Существуют и другие варианты монтирования файловых систем, но уже перечисленных достаточно, чтобы понять идею этого процесса. Для описания процесса монтирования *файловой системы* некоторого устройства обычно просто говорят: "монтируется устройство", что следует понимать как "монтирование файловой системы устройства".

Итак, предположим вы только что смонтировали корневую файловую систему (/) и хотите смонтировать IDE CD-ROM, /dev/hdd, в точку монтирования /media/cdrom. Точка монтирования должна существовать до того, как вы смонтируете в нее CD-ROM. Когда вы монтируете CD-ROM, файлы и подкаталоги CD-ROM становятся файлами и подкаталогами внутри /media/cdrom. Любые файлы и подкаталоги, что уже имелись в /media/cdrom становятся не видны, хотя они все еще существуют на блочном устройстве, содержащем точку монтирования /media/cdrom. Как только CD-ROM будет размонтирован, все оригинальные файлы и каталоги вновь станут видны. Вам следует избегать проблем, связанных с размещением других файлов в каталоге, который собираетесь использовать в качестве точки монтирования.

В Таблице 1 показаны каталоги, которые должны быть в / согласно Filesystem Hierarchy Standard [Стандарту иерархии файловых систем] (более подробно об FHS, смотри [Ресурсы](#)).

Таблица 1. Каталоги FHS в /

Каталог	Описание
bin	Бинарные файлы важных программ
boot	Статические файлы загрузчика
dev	Файлы устройств
etc	Настройки этой системы
lib	Важные общие библиотеки и модули ядра
media	Точка монтирования для временных мультимедийных устройств
mnt	Точка монтирования для временного монтирования файловых систем
opt	Дополнительные пакеты для программного обеспечения
sbin	Важные бинарные файлы системы
srv	Данные служб, запущенных на компьютере
tmp	Временные файлы
usr	Вторичная иерархия
var	Часто изменяемые данные

## Разделы

Первый учебник этой серии, "[Учебник для экзамена LPI 101 \(тема 101\): Аппаратное обеспечение и архитектура](#)" слегка затронул разделы жестких дисков, и теперь мы собираемся разобраться в них более детально.

Первый жесткий диск IDE в системе Linux это /dev/hda, а первый SCSI диск это /dev/sda. Жесткий диск разбит на *сектора* по 512 байт. Все сектора на пластине жесткого диска, которые могут быть прочитаны без перемещения головки, называются *трэками* [*Прим.пер.: в русскоязычной литературе также встречается термин "дорожка"*]. Диски обычно имеют более одной пластины. Набор дорожек разных пластин, которые могут быть прочитаны без перемещения головок называется *цилиндром*. *Геометрия* жесткого диска выражается в цилиндрах, дорожках (или *головках*) на цилиндр и секторах на дорожке.

Ограничения на возможные значения каждой из этих величин, использовавшиеся операционной системой, привели к тому, что указанные в BIOS параметры геометрии диска пришлось преобразовывать, чтобы появилась возможность работы с большими дисками. В конце концов и этих методов стало не достаточно. Большинство последних разрабатываемых технологий жестких дисков могут использоваться только с *логической адресацией блоков (LBA -- logical block addressing)*, так что физические единицы геометрии CHS все менее важны и отображаемая геометрия может быть не совсем верна или вообще не иметь связи со структурой современных дисков. Диски больших размеров, которые используются сегодня, работают с расширением LBA, известным как LBA48 и отличающимся тем что на

нумерацию секторов резервируется до 48 бит.

Пространство жесткого диска разбито (или разделено) на *разделы (partition)*. Разделы не могут перекрываться; пространство не входящее ни в один раздел называется *свободным пространством (free space)*. Разделы имеют имена /dev/hda1, /dev/hda2, /dev/hda3, /dev/sda1 и тому подобные. IDE диски имеют ограничение в 63 раздела, а SCSI диски до 15. Разделы обычно содержат целое число цилиндров (что связано с возможностью ошибочной ссылки на цилиндр).

Если две различные программы для разбиения по разному понимают номинальную геометрию диска, то одна из программ может сообщать об ошибке или иметь проблемы с разделами, созданными другой программой для разбиения. Вы также можете встретить проблемы такого рода если диск был перемещен из одной системы в другую, особенно если возможности BIOS различны. В Linux вы можете узнать номинальную геометрию, просмотрев соответствующий файл в файловой системе /proc, например, /proc/ide/hda/geometry. Эта геометрия используется такими инструментами разбиения диска как fdisk и parted. Листинг 1 показывает использование команды **cat** для отображения /proc/ide/hda/geometry, а следом идет информационное сообщение, полученное при использовании инструмента разбиения **parted**.

### Листинг 1. Геометрия жесткого диска

```
[root@lyrebird root]# cat /proc/ide/hda/geometry
physical      19457/255/63
logical      19457/255/63
[root@lyrebird root]# parted /dev/hda
GNU Parted 1.6.3
Copyright (C) 1998, 1999, 2000, 2001, 2002 Free Software Foundation, Inc.
This program is free software, covered by the GNU General Public License.
```

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

```
Using /dev/hda
Information: The operating system thinks the geometry on /dev/hda is
19457/255/63. Therefore, cylinder 1024 ends at 8032.499M.
(parted)
```

Заметьте, что в Листинге 1, **parted** вычислила номинальную позицию конца 1024 цилиндра. Цилиндр 1024 очень важен для тех систем, где BIOS может загружаться только с раздела, который размещается в первых 1024 цилиндрах диска. Наиболее вероятно, что это произойдет с BIOS, не имеющей поддержки LBA. На современных машинах это уже не проблема, хотя вам следует быть осторожными, поскольку это ограничение может существовать.

Существует три вида разделов: *primary (основной)*, *logical (логический)*, и *extended (расширенный)*. *The partition table* (Таблица разделов) расположена в *главной загрузочной записи (master boot record -- MBR)* диска. MBR это первый сектор диска, так что таблица разделов не очень большая его часть. Это ограничивает количество основных разделов числом 4. Когда требуется более четырех разделов, а это бывает часто, один из основных разделов должен быть определен как *расширенный*. Диск может содержать только один расширенный раздел.

*Расширенный раздел* это не более чем контейнер для логических разделов. Эта схема

разбиения изначально использовалась в MS DOS и PC DOS, и позволяет использовать диски ПК в DOS, Windows или Linux системах.

В Linux может быть от 1 до 4 основных и расширенных разделов, то есть dev hda может иметь четыре основных раздела: /dev/hda1, /dev/hda2, /dev/hda3 и /dev/hda4. Или оно может иметь один основной раздел /dev/hda1 и один расширенный /dev/hda2. Если определены логические разделы, то их нумерация начинается с 5, то есть первый логический раздел на /dev/hda будет нумероваться /dev/hda5, даже если на диске основного раздела нет, а есть только расширенный (/dev/hda1).

На Листинге 2 приведен вывод команды **parted** с ключом **p**, отображающий информацию о разделах для того же диска, что и в Листинге 1. Заметьте, что эта система содержит несколько различных файловых систем Windows и Linux.

## Листинг 2. Отображение таблицы разделов при помощи parted

```
(parted) p
Disk geometry for /dev/hda: 0.000-152627.835 megabytes
Disk label type: msdos
      Minor    Start       End     Type   Filesystem   Flags
1          0.031  16300.327 primary    ntfs        boot
2     16300.327 25846.765 primary    fat32        lba
3     25846.765 26842.983 primary    ext3
4     26842.983 152625.344 extended
5     26843.014 28898.173 logical  linux-swap
6     28898.205 48900.981 logical  ext3
7     48901.012 59655.432 logical  ext3
8     59655.463 75657.678 logical  ext3
9     75657.709 95001.569 logical  ext3        boot
10    95001.601 122997.656 logical  reiserfs
11   122997.687 152625.344 logical  ext3
```

## Распределение места на диске

Как указывалось ранее, файловая система Linux имеет единственное большое дерево с корнем /. Из этого очевидно почему данные на диске или CD-ROM должны быть смонтированы, но возможно из этого не совсем ясно зачем создавать различные разделы на жестком диске. Некоторыми вескими причинами разделения файловых систем могут являться:

- Файлы загрузки. Некоторые файлы должны быть доступны для BIOS или загрузчика во время загрузки.
- Несколько жестких дисков. Обычно каждый жесткий диск может быть разбит на один или несколько разделов, каждый со своей файловой системой, которая должна быть смонтирована куда-нибудь в дереве файловой системы.
- Файлы совместного доступа. Некоторые системы могут подразумевать совместное использование статических файлов, таких как исполняемые файлы программ. Динамические файлы, такие как домашний каталог пользователя или файлы почтовой очереди также могут использоваться совместно, чтобы пользователи могли осуществлять вход на любой машине из нескольких, находящихся в сети и работать со своим домашним каталогом и почтовой системой.
- Возможность переполнения. Если заполненность файловой системы приближается к 100 процентам, то обычно хорошей мыслью будет хранить файлы, необходимых

системе для функционирования на отдельном разделе.

- Квоты. Квоты, ограничивающие доступное пространство в файловой системе для пользователя или группы.
- Монтирование только для чтения. До изобретения журналируемых файловых систем восстановление файловых систем после системного сбоя часто занимало много времени. Поэтому редко изменяющиеся файловые системы (такие как каталоги исполняемых программ) могут быть смонтированы в режиме только для чтения, чтобы не тратить много времени на их проверку после системного сбоя.

В добавок к используемым файловым системам вы также должны выделить место на диске для раздела подкачки (swap раздела). В системе Linux под него отводится обычно один или, возможно, несколько разделов.

## Определение параметров

Предположим, вы настраиваете систему, в которой есть по крайней мере один жесткий диск, и вы хотите грузиться с этого жесткого диска. (В этом учебнике не рассматривается настройка бездисковой рабочей станции, которая загружается по сети, а также не принимается во внимание использование Linux LiveCD или DVD). Несмотря на то, что размер раздела можно изменить позднее, это обычно требует некоторых усилий, поэтому важно сделать правильный выбор в начале. Итак, начнем.

Во первых вы должны быть уверены, что система будет загружаться. Некоторые старые компьютеры имеют ограничение при котором BIOS может осуществить загрузку только с раздела, который полностью расположен в первых 1024 цилиндрах диска. Если у вас подобная машина, то вы **должны** создать раздел, который затем будет монтироваться как /boot и хранить ключевые файлы, необходимые для загрузки системы. После загрузки, система Linux возьмет работу с диском на себя и ограничение в 1024 цилиндра больше не будет влиять на работу системы. Если вам необходимо создать раздел /boot, то обычно достаточно около 100МБ.

Следующее с чем вы должны определиться это размер swap-раздела. При сегодняшних параметрах оперативной памяти, swap представляет собой вторичную более медленную память. Общепринято было создавать swap раздел соразмерным имеющейся оперативной памяти. В настоящее время вы можете выделить под него 500МБ для рабочей станции и около 1ГБ для сервера. Если особые обстоятельства требуют этого, то вы можете увеличить его размер, однако если вы сделаете это, то ваша система может потерять производительности, так что лучше следует увеличить реальную оперативную память. Можно использовать и файл подкачки, однако выделение отдельного раздела предпочтительнее.

Теперь мы подошли к критической точке. Требования для персональной рабочей станции менее предсказуемы, чем для сервера. Я рекомендую всем (и в особенности новичкам) размещать большинство стандартных каталогов (/usr, /opt, /var, /etc) на одном большом разделе. Это особенно полезно для пользователей, впервые устанавливающих Linux и не имеющих ясного представления, что будет получено в конечном итоге. Рабочая станция с графическим рабочим столом и разумным набором средств разработки может потребовать от 2 до 3 гигабайт плюс место, необходимое пользователю. Но некоторые более массивные инструменты разработки могут потребовать несколько гигабайт каждый. Я обычно выделяю где-то от 10 до 20 ГБ на одну операционную систему, а остальное оставляю свободным для загрузки другого дистрибутива.

Рабочая нагрузка сервера более стабильна, но и нехватка места на файловой системе может привести к большим неприятностям. Так что для них я советую создавать несколько разделов, распределённых по нескольким дискам с использованием аппаратного или

программный RAID или же группы логических томов.

Вам также может потребоваться определить загрузку каждой файловой системы и будет ли файловая система доступна для нескольких операционных систем или будет использоваться только одной. Вы можете использовать ваш опыт, инструменты планирования размера, а также предположения о перспективах роста, чтобы определить наилучшее распределение дискового пространства для вашей системы.

Вне зависимости от того, что вы настраиваете -- рабочую станцию или сервер -- у вас будут некоторые файлы, уникальные для каждой операционной системы, расположенной на локальном диске. Обычно это /etc для системных параметров, /boot для файлов, необходимых во время загрузки, /sbin для файлов, необходимых для загрузки или восстановления системы, /root для домашнего каталога суперпользователя, /var/lock для lock файлов, /var/run для информации работающей системы и /var/log для файлов журналов (log-файлов) этой системы. Другие файловые системы, такие как /home для домашних каталогов пользователей, /usr, /opt, /var/mail, /var/spool/news могут располагаться на отдельных разделах или смонтированы по сети в соответствии с вашими нуждами и предпочтениями.

| [предыдущая](#) | [следующая](#)

## Менеджеры загрузки

Этот раздел охватывает материал для темы 1.102.2 экзамена LPI 101 Администрирование для начинающих (LPIC-1). Рейтинг темы 1.

В этом разделе обсуждает процесс загрузки ПК и два основных менеджера загрузки, используемых в Linux: LILO и GRUB. Мы рассмотрим выбор менеджера загрузки, а также восстановление, когда что-то идет не так.

### Обзор процесса загрузки

Перед тем, как углубиться в LILO и GRUB, давайте рассмотрим как ПК начинает работу или загружается. Код, называемый *BIOS* (от *Basic Input Output Service* -- Базовая служба ввода/вывода) хранится в энергонезависимой памяти, такой как ROM, EEPROM или flash-памяти. Когда ПК включается или перезагружается, запускается этот код. Обычно он выполняет тестирование при включении питания (power-on self test -- POST) для проверки машины. В конце он загружает первый сектор главной загрузочной записи (MBR) загрузочного диска.

Как обсуждалось в предыдущем параграфе [Разделы](#), MBR также содержит таблицу разделов, так что объём выполняемого кода в MBR меньше чем 512 байт, так что он не может содержать много инструкций. Заметим, что каждый диск, даже дискета (floppy), содержит исполняемый код в своем MBR, даже если код просто выводит сообщение вроде "Non-bootable disk in drive A:" ("В дисководе A: не загрузочный диск"). Этот код загружается BIOS из первого сектора, называемого *первичный загрузчик* (*first stage boot loader*) или *загрузчик первой стадии* (*stage 1 boot loader*).

Стандартный загрузчик в MBR жесткого диска, который использовался операционными системами MS DOS, PC DOS и Windows, проверяет таблицу разделов для определения основного раздела на загрузочном диске, помеченного как *активный* (*active*), загружает первый сектор этого раздела и передает управление в начало загруженного кода. Этот новый кусок кода также известен как *загрузочная запись раздела* (*partition boot record*). Загрузочная запись раздела тоже является загрузчиком первой стадии, но он уже достаточно интеллектуален, чтобы загрузить несколько блоков раздела. В этих блоках располагается код *загрузчика второй стадии* (*stage 2 boot loader*). В MS-DOS и PC-DOS загрузчик второй стадии непосредственно переходит к загрузке оставшейся части операционной системы. Таким образом, операционная система при загрузке проходит несколько вспомогательных

шагов, пока не перейдет в рабочее состояние.

Описанная схема работает прекрасно для компьютеров с одной операционной системой. Но что произойдет, если вам понадобится несколько операционных систем. Скажем Windows 98, Windows XP и три различных дистрибутива Linux? Вы могли бы при помощи некой программой (такой как DOS FDISK) сменить активность разделов и перезагрузиться. Но это утомительно. Кроме того, диск может иметь только четыре основных раздела, а стандартный MBR загрузчик умеет выполнять загрузку только с основного раздела. Но приведенный пример подразумевает пять операционных систем, каждой из которых нужен раздел!

Решение состоит в том, чтобы использовать некоторый специальный код, который позволяет пользователю выбрать систему для загрузки. Приведем примеры:

1. Loadlin, исполняемая DOS-программа запускаемая в работающей системе DOS для загрузки с Linux раздела. Она была популярна, когда настройка множественной загрузки была сложным и рискованным делом.
2. OS/2 Boot Manager -- программа, устанавливавшаяся в небольшой специальный раздел. Раздел помечался как активный и стандартный MBR процесса загрузки запускал Менеджер загрузки (Boot Manager), который выводил меню, позволявшее пользователю выбрать систему для загрузки.
3. Умный менеджер загрузки, программа, которая может размещаться в разделе операционной системы и вызываемая или загрузочной записью активного раздела или основной загрузочной записью Примеры:
  - BootMagic™, часть Norton PartitionMagic™
  - LILO, the LInux LOader (Загрузчик Linux)
  - GRUB, the GRand Unified Boot loader (Главный Унифицированный Загрузчик)

Очевидно, что если вы можете передать управление программе содержащей более 512 байт для выполнения своих задач, то не трудно обеспечить загрузку с логических разделов или загрузку с разделов, находящихся не на загрузочном диске. Все эти решения предоставляют эти возможности или путем загрузки загрузочной записи с любого раздела или потому, что знают какой файл или файлы следует загрузить, чтобы начать процесс загрузки.

Далее мы сосредоточимся на LILO и GRUB, поскольку это загрузчики, включенные во многие дистрибутивы Linux. В процессе установки вашего дистрибутива вам, вероятно, было предложено установить один из них на выбор. Оба могут работать с большинством современных дисков. Запомните, что технологии жестких дисков развиваются стремительно, поэтому вы всегда должны проверить и убедиться, что выбранный вами загрузчик, а также дистрибутив Linux и ваша BIOS будут работать с новеньким сияющим диском. Не соблюдение этого условия может привести к потере данных.

Загрузчик второй стадии, используемый LILO и GRUB позволяет вам выбрать какую из имеющихся операционных систем или их версий следует загрузить. Однако LILO и GRUB значительно различаются в том, что изменение системы требует выполнения некой команды для создания заново настроек загрузки LILO при обновлении ядра или выполнении других подобных операций, тогда как для GRUB это можно сделать отредактировав текстовый файл настроек. LILO существует давно. GRUB новее. Оригинальный GRUB теперь стал *GRUB Legacy*, а GRUB 2 разрабатывается под патронажем Free Software Foundation (смотри [Ресурсы](#)).

## LILO

LILO, или LInux LOader (Загрузчик Linux), один из наиболее распространенных загрузчиков Linux. LILO может быть установлен в MBR вашего загрузочного жесткого диска или в загрузочную запись раздела. Он также может быть установлен на сменных носителях, таких как дискеты (floppy-диски), CD или USB диски. Если вы еще не знакомы с LILO, то хорошей мыслью будет попрактиковаться на дискете или USB диске, что мы и будем делать в наших

примерах.

При установке Linux вы обычно указываете в качестве менеджера загрузки LILO или GRUB. Если вы выбираете GRUB, то скорее всего LILO не будет установлен автоматически. В этом случае вам потребуется установить пакет LILO вручную. Если вам необходима помощь для этого, то смотрите раздел этого учебника об управлении пакетами ниже. В дальнейшем мы предполагаем, что LILO уже установлен.

Основная функция команды **lilo**, расположенной в /sbin/lilo, запись загрузчика первой стадии и создание файла распределения памяти (/boot/map), используя данные конфигурации, которые обычно располагаются в /etc/lilo.conf. У неё есть несколько дополнительных назначений, которые мы опишем позднее. Для начала давайте посмотрим на типичный файл конфигурации LILO, который может быть использован при двойной загрузке Windows и Linux.

### Листинг 3. Пример /etc/lilo.conf

```
prompt
timeout=50
compact
default=linux
boot=/dev/fd0
map=/boot/map
install=/boot/boot.b
message=/boot/message
lba32
password=mypassword
restricted

image=/boot/vmlinuz-2.4.21-32.0.1.EL
    label=linux
    initrd=/boot/initrd-2.4.21-32.0.1.EL.img
    read-only
    append="hdd=ide-scsi root=LABEL=RHEL3"

other=/dev/hda1
    loader=/boot/chain.b
    label=WIN-XP
```

Первый набор опций, приведенных выше -- это глобальные опции, управляющие работой LILO. Второй и третий представляют опции для каждого образа двух операционных систем, которые мы хотим загружать через LILO, в данном примере Red Hat Enterprise Linux 3 и Windows XP.

Глобальные опции в нашем примере это:

#### **prompt**

вызывает вывод сообщения загрузки.

#### **timeout**

указывает в десятых долях секунды задержку перед автоматической загрузкой системы по умолчанию. В нашем примере timeout=50, что эквивалентно задержке в 5 секунд.

#### **compact**

пытается объединить запросы на чтение для смежных секторов. Это ускоряет загрузку и уменьшает размер файла распределения памяти.

#### **default**

указывает какая операционная система будет грузиться по умолчанию. Если не указано,

то грузится первая по списку. В нашем примере если пользователь не выберете что-либо в течении 5 секунд, то будет загружена система Linux.

### **boot**

указывает куда будет установлен LILO. В нашем примере это дискета (floppy диск) /dev/fd0. Для установки в MBR первого жесткого диска укажите boot=/dev/hda. Наша система RHEL 3 в действительности располагается на /dev/hda11, поэтому мы должны указать boot=/dev/hda11 если хотим установить LILO в этот раздел. Если этот параметр опущен, то LILO попытается использовать загрузочный сектор устройства, смонтированного в данное время как root (/).

### **map**

указывает расположение файла распределения памяти, используемый LILO, для вывода сообщений пользователю и загрузки операционных систем указанных в секциях image файла lilo.conf. По умолчанию это /boot/map.

### **install**

указывает новый файл для установки в качестве загрузочного сектора. По умолчанию устанавливается /boot/boot.b, являющийся частью пакета LILO.

### **message**

определяет сообщение, появляющееся перед приглашением загрузки. Оно должно быть менее 65535 байт. Если ваша система отображает графическую оболочку меню LILO, то вы обнаружите, что /boot/message содержит файл картинки. На некоторых системах Red Hat это будет файл 300x200 пикселов в формате PCX. Для систем SUSE это может быть 16 цветный bitmap-файл размером 640x480 пикселов. В этом случае вы также можете обнаружить несколько дополнительных параметров. Посмотрите документацию, идущую с вашей системой. Например, моя система SUSE SLES9 хранит его в /usr/share/doc/packages/lilo/README.bitmaps.

### **lba32**

указывает, что LILO следует использовать для дисков режим LBA32 вместо CHS или линейной адресации секторов.

### **password**

определяет пароль, который следует ввести перед загрузкой образа. Заметим, что это обычный текст, поэтому следует установить такие атрибуты файла /etc/lilo.conf, чтобы запретить просмотр этого файла всем пользователям кроме root. Он не должен совпадать с паролем суперпользователя (root). И password, и следующая за ним опция, restricted, на самом деле являются примерами опций для каждого образа, которые для удобства могут быть указаны в глобальной секции. Если указано именно так, то одно и то же значение используется для всех образов до тех пор, пока оно не будет переопределено в разделе настроек конкретного образа.

### **restricted**

смягчает запрос пароля так, что пароль запрашивается только если пользователь пытается использовать при загрузке дополнительные параметры. Вы можете использовать это, чтобы позволить пользователю загружаться нормально без ввода пароля, но заставить ввести пароль при загрузке в режиме единичного пользователя.

Следующая секция описывает опции специфичные для RHEL3.

### **image**

указывает, что это секция системы Linux, которая загружается из файла. Параметром является имя файла образа ядра Linux.

### **label**

это необязательная метка, которую вы можете ввести вместо полного имени файла образа.

### **initrd**

это имя инициализационного RAM-диска, который содержит модули, необходимые ядру

до того, как будет смонтирована файловая система.

#### **read-only**

указывает, что корневая файловая система должна быть изначально смонтирована в режиме только для чтения. Последующие стадии загрузки обычно перемонтируют ее в режим чтение/запись после того, как она будет проверена.

#### **append**

указывает опции, передаваемые ядру. В нашем примере указано, что для /dev/hdd должна использоваться эмуляция SCSI (2.4 и более ранние ядра использовали таким образом оптические устройства типа CD-ROM). Также указано, что раздел с меткой RHEL3 должен монтироваться как корневой (/).

В последней секции указаны опции для нашей не-Linux системы.

#### **other**

указывает имя устройства, содержащего устройство (или файл) в котором находится загрузочный сектор загружаемой системы.

#### **loader**

указывает используемый загрузчик. LILO поддерживает chain.b, который просто загружает эту загрузочную запись загрузочного раздела или, как вариант, /boot/os2\_d.b который может использоваться для загрузки OS/2 со второго жесткого диска.

#### **label**

не обязательная метка, которую вы можете использовать вместо полного имени образа при выборе образа.

Теперь если мы вставим пустую дискету мы сможем выполнить команду **lilo** (/sbin/lilo) для создания загрузочной дискеты как показано в Листинге 4. Заметьте, что команда lilo имеет пять уровней детальности вывода. Добавьте дополнительный -v для каждого уровня.

#### **Листинг 4. Создание загрузочной дискеты с lilo**

```
[root@lyrebird root]# lilo -v -v
LILO version 21.4-4, Copyright (C) 1992-1998 Werner Almesberger
'lba32' extensions Copyright (C) 1999,2000 John Coffman

Reading boot sector from /dev/fd0
Merging with /boot/boot.b
Secondary loader: 11 sectors.
Mapping message file /boot/message
Compaction removed 43 BIOS calls.
Message: 74 sectors.
Boot image: /boot/vmlinuz-2.4.21-32.0.1.EL
Setup length is 10 sectors.
Compaction removed 2381 BIOS calls.
Mapped 2645 sectors.
Mapping RAM disk /boot/initrd-2.4.21-32.0.1.EL.img
Compaction removed 318 BIOS calls.
RAM disk: 354 sectors.
Added linux *
Boot other: /dev/hda1, on /dev/hda, loader /boot/chain.b
Compaction removed 0 BIOS calls.
Mapped 6 (4+1+1) sectors.
Added WIN-XP
/boot/boot.0200 exists - no backup copy made.
Map file size: 8192 bytes.
```

Writing boot sector.

Теперь мы получили загрузочную дискету LILO. Если LILO обнаружит ошибку, то вы увидите сообщение о ней и загрузочный сектор не будет записан. Например, если в нашем файле /etc/lilo.conf мы пропустим опцию lba32, то мы увидим нечто похожее на приведенное в Листинге 5. Это будет совет использовать опции linear или lba32. В этом случае мы используем командную строку для указания опции -l, что равнозначно указанию опции linear в lilo.conf. Если мы проделаем это еще раз с опцией -L, то lilo должен завершить все успешно и результат будет схож с приведенным ранее.

### Листинг 5. Пример /etc/lilo.conf с ошибкой

```
[root@lyrebird root]# lilo
Warning: device 0x030b exceeds 1024 cylinder limit
Fatal: geo_comp_addr: Cylinder number is too big (16284 > 1023)
[root@lyrebird root]# lilo -l
Warning: device 0x030b exceeds 1024 cylinder limit
Fatal: sector 261613688 too large for linear mode (try 'lba32' instead)
```

При тестировании вашей загрузочной дискеты, измените запись boot=/dev/fd0 в вашем файле lilo.conf, чтобы установить LILO в MBR или загрузочную запись раздела. Например, указание boot=/dev/hda установит LILO в основную загрузочную запись вашего первого жесткого диска IDE.

Вы получили некоторое представление о LILO и его конфигурационном файле, включая то как изменять некоторые опции конфигурации из командной строки lilo. Более подробную информацию вы найдете в man-страницах lilo, используя команду **man lilo**. Вы можете найти еще более развернутую информацию в руководстве пользователя в формате postscript, устанавливаемом вместе с пакетом lilo. Оно должно быть установлено в вашем каталоге с документацией, но точное место расположения может варьироваться от системы к системе. Одним из способов ее обнаружения является фильтрация списка пакетов при помощи grep. Листинг 6 показывает это для основанной на rpm системы RHEL3, которую мы используем в качестве примера.

### Листинг 6. Обнаружение руководства пользователя при помощи rpm.

```
[ian@lyrebird ian]$ rpm -ql lilo | grep ".ps$"
/usr/share/doc/lilo-21.4.4/doc/Technical_Guide.ps
/usr/share/doc/lilo-21.4.4/doc/User_Guide.ps
```

## Дополнительные параметры командной строки LILO

LILO имеет несколько дополнительных параметров командной строки.

### **lilo -q**

отображает информацию из файла распределения памяти

### **lilo -R**

настроит lilo на автоматическую загрузку определенной системы при следующей перезагрузке. Это очень удобно для автоматической перезагрузки удаленных систем.

### **lilo -I**

отображает информацию о пути к файлу ядра

## **lilo -u**

удаляет lilo и восстанавливает прежнюю загрузочную запись.

При загрузке системы Linux через LILO, вам может понадобиться указать дополнительные параметры. Например, если графическая оболочка не загружается, то вы можете захотеть загрузиться в режиме mode 3 или в однопользовательском режиме для восстановления. Любой текст, набираемый вами после имени метки будет передан ядру. Например, в нашем примере мы можем выбрать систему RHEL просто набрав "linux". Для загрузки в mode 3 или однопользовательском режиме, следует набрать одну из указанных строк соответственно.

```
linux 3  
linux single
```

Напомним также, что с LILO вы **должны** выполнить команду lilo каждый раз при обновлении файла настроек (/etc/lilo.conf). Вам также следует выполнять команду lilo если вы добавляете, перемещаете или удаляете разделы или производите любые другие изменения, которые могут повредить генерированный загрузчик.

## **GRUB**

GRUB или GRand Unifood Boot loader (Главный Унифицированный Загрузчик), это второй из двух наиболее распространенных загрузчиков Linux. Как и LILO, GRUB может быть установлен в MBR вашего загрузочного жесткого диска или в загрузочную запись раздела. Также он может быть установлен на сменных носителях, таких как дискета, CD или USB драйв. Если вы еще не знакомы с GRUB, то хорошей идеей будет попрактиковаться на диске или USB диске, что мы и будем делать в приводимом примере.

GRUB, или GNU GRUB, в настоящее время разрабатывается под патронажем Free Software Foundation. Новая версия, GRUB 2 находится в стадии разработки, а оригинальная версия GRUB 0.9x теперь известна как Grub Legacy.

В процессе установки Linux вы обычно выбираете в качестве менеджера загрузки или LILO, или GRUB. Если вы выбираете LILO, скорее всего GRUB не будет установлен автоматически. В этом случае вам потребуется установить пакет GRUB вручную. Смотри секцию управления пакетами ниже в этом учебнике, если для этого вам необходима помощь. В дальнейшем мы будем предполагать, что он уже установлен.

GRUB имеет файл настроек, расположенный обычно в /boot/grub/grub.conf. Если ваша система поддерживает символьные ссылки, как большинство файловых систем Linux, то вероятно у вас есть /boot/grub/menu.lst как символьная ссылка на /boot/grub/grub.conf.

Команда **grub** (/sbin/grub, а на некоторых системах, /usr/sbin/grub) -- это небольшая, но чрезвычайно мощная оболочка, которая поддерживает различные команды для установки GRUB, загрузки систем, размещения и отображения конфигурационных файлов и подобных задач. В этой оболочке используется тот же код, который загружается на второй стадии загрузчика GRUB, поэтому полезно изучить GRUB без выполнения перезагрузки компьютера. Стадия 2 в GRUB запускается или в командном режиме, или в меню, позволяя вам выбрать операционную систему из меню или указать специальную команду загрузки системы. Имеется также несколько других команд, таких как **grub-install**, которые используют оболочку grub и помогают автоматизировать задачи вроде установки GRUB.

Листинг 7 содержит часть конфигурационного файла GRUB. Просматривая его, помните одну важную вещь -- GRUB нумерует диски, разделы и все остальное что должно быть пронумеровано, начиная с 0, а не с 1.

## Листинг 7. Пример конфигурационного файла GRUB /boot/grub/menu.lst.

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
#          all kernel and initrd paths are relative to /, eg.
#          root (hd1,5)
#          kernel /boot/vmlinuz-version ro root=/dev/hdc6
#          initrd /boot/initrd-version.img
#boot=/dev/hdc6
default=2
timeout=10
splashimage=(hd0,6)/boot/grub/splash.xpm.gz
password --md5 $1$/8Kl21$3VPIphs6REHeHccwzjQY0.

title Red Hat Linux (2.4.20-31.9)
    root (hd0,6)
    kernel /boot/vmlinuz-2.4.20-31.9 ro root=LABEL=RH9 hdd=ide-scsi
    initrd /boot/initrd-2.4.20-31.9.img

title Red Hat Linux (2.4.20-6)
    root (hd0,6)
    kernel /boot/vmlinuz-2.4.20-6 ro root=LABEL=RH9 hdd=ide-scsi
    initrd /boot/initrd-2.4.20-6.img

title Red Hat Enterprise Linux WS A (2.4.21-32.0.1.EL)
    root (hd0,10)
    kernel /boot/vmlinuz-2.4.21-32.0.1.EL ro root=LABEL=RHEL3 hdd=ide-scsi
    initrd /boot/initrd-2.4.21-32.0.1.EL.img

title Ubuntu, kernel 2.6.10-5-386
root      (hd1,10)
kernel   /boot/vmlinuz-2.6.10-5-386 root=/dev/hdb11 ro quiet splash
initrd   /boot/initrd.img-2.6.10-5-386
savedefault
boot

title Ubuntu, kernel 2.6.10-5-386 (recovery mode)
lock
root      (hd1,10)
kernel   /boot/vmlinuz-2.6.10-5-386 root=/dev/hdb11 ro single
initrd   /boot/initrd.img-2.6.10-5-386
boot

title Win/XP
    rootnoverify (hd0,0)
    chainloader +1

title Floppy
    root (fd0)
    chainloader +1
```

Также как и в конфигурационном файле LILO, первый набор опций определяет поведение GRUB. Для GRUB он называется *menu commands* (*команды меню*) и должен быть указаны до остальных команд. Остальные секции содержат опции для каждого образа операционных систем, которые мы хотим загружать через GRUB. Заметьте, что "title" -- это команда меню. Каждому вхождению title сопутствует одна или несколько базовых команд или пунктов меню. Пример LILO был типичным простым вариантом двойной загрузки

систем Windows и Linux. Этот же пример был создан на том же компьютере, что и предыдущий, но в него мы добавили несколько дополнительных операционных систем, чтобы показать вам кое-что из возможностей загрузчика. Вы можете распознать многие сходные элементы, появляющиеся в конфигурационных файлах и LILO, и GRUB. Вы можете задаться мыслью, что же следует изменить, чтобы добавить эти несколько дополнительных операционных систем к предыдущему примеру LILO.

Команды меню, применяемые ко всем остальным секциям в нашем примере это:

**#**

Любые строки, начинающиеся с # это комментарии и GRUB их игнорирует. Этот конфигурационный файл изначально был создан anaconda, установщиком Red Hat. Если вы установили GRUB при установке Linux, то возможно вы обнаружите комментарии, добавленные в конфигурационный файл GRUB. Комментарии обычно выступают в качестве помощи программам обновления системы, чтобы настройки GRUB оставались рабочими при обновлениях ядра. Если вы самостоятельно редактируете файл конфигурации, то обратите внимание на любые пометки, оставленные в этих целях.

**default**

указывает какая система будет грузиться по умолчанию, если пользователь не сделает выбор в отведенное время (timeout). В нашем примере, default=2, что означает загрузку третьей записи. напомним, что GRUB использует нумерацию с 0, а не с 1. Если ничего не указано, то по умолчанию будет грузиться первая запись, то есть запись с номером 0.

**timeout**

указывает в секундах время задержки перед началом загрузки системы по умолчанию. Заметьте, что LILO для указания задержки использует десятые доли секунды, тогда как GRUB использует целые секунды.

**splashimage**

Указывает фоновое или *splash*-изображение, которое будет отображаться в загрузочном меню. GRUB обращается к первому жесткому диску как (hd0) и первому разделу этого диска (hd0,0), так что указание splashimage=(hd0,6)/boot/grub/splash.xpm.gz означает использование файла /boot/grub/splash.xpm.gz, расположенного в разделе 7 первого жесткого диска. Запомните, что нумерация с нуля. Отметим также, что образ является XPM файлом, сжатым при помощи gzip. Поддержка *splash*-изображения это патч (patch), который может содержаться, а может и не содержаться в вашем дистрибутиве.

**password**

определяет пароль, который требуется ввести для доступа к меню, а также ввода команд GRUB и изменения настроек. Как и в LILO пароль может быть обыкновенным текстом. GRUB позволяет также хранить пароли в виде MD5 файлов, как показано в нашем примере. Это в некоторой степени более безопасно и большинство администраторов устанавливают пароли именно так. Если пароль не используется, то пользователь получает полный доступ к командной строке GRUB.

В нашем примере используется пять дистрибутивов Linux (три Red Hat и два Ubuntu) плюс Windows XP и возможность загрузки с дискеты. Команды, используемые в этих секциях:

**title**

это заголовок-описание, отображающийся в строке меню при загрузке GRUB. Вы используете клавиши стрелок для перемещения вверх и вниз по заголовкам, а затем нажимаете клавишу **Enter** для выбора некоторого элемента.

**root**

указывает раздел, с которого следует загружаться. Помните, что как и для splashimage, нумерация начинается с 0, так что первая система Red Hat, указанная как root (hd0,6) это на самом деле расположена на разделе 7 первого жесткого диска (в данном случае /dev/hda7), В то время как для первой системы Ubuntu в качестве root указан (hd1,10)

расположенный на втором жестком диске (/dev/hdb11). GRUB попытается смонтировать это раздел для его проверки и обеспечения загрузки операционной системы с разными параметрами.

#### **kernel**

указывает образ ядра, который следует загрузить, а также параметры ядра. Это похоже на комбинацию команд LILO image и append. В приведенном примере у нас имеется два разных ядра Red Hat 9, плюс ядро рабочей станции Red Hat Enterprise Linux 3 Workstation, и одна и та же система Ubuntu с двумя различными наборами параметров ядра.

#### **initrd**

Имя *RAM-диска*, содержащего модули, необходимые ядру перед монтированием файловой системы.

#### **savedefault**

Приведено здесь для иллюстрации. Если указана команда меню default=saved, и для операционной системы используется команда savedefault, то загрузка этой операционной системы приведет к тому, что она станет выбором по умолчанию до тех пор, пока не будет загружена операционная система, также имеющая команду savedefault. В нашем примере указание default=2 приводит к игнорированию всех сохраненных умолчаний (saved default).

#### **boot**

не обязательный параметр, который указывает GRUB загружать выбранную операционную систему. Это действие по умолчанию, после того, как выполнены все команды для выделенного.

#### **lock**

в приведенном примере используется для второй системы Ubuntu. Эта система будет загружена в режиме единственного пользователя, что позволяет пользователю производить изменения в системе, которые обычно требуют прав суперпользователя (root). Если указана эта опция, то вы должны также указать password в начальных опциях, иначе пользователь сможет изменить вашу опцию lock и загрузить систему или добавить "single" к другой из имеющихся записей. При желании можно также указать различные пароли для каждой записи.

#### **rootnoverify**

похожа на root, за исключением того, что GRUB не пытается смонтировать файловую систему или проверить ее параметры. Обычно она используется для таких файловых систем, как NTFS, которые не поддерживаются GRUB. Вы также можете использовать ее если хотите загрузить главную загрузочную запись (MBR) жесткого диска, например для доступа к другому конфигурационному файлу или для перезагрузки предыдущего загрузчика.

#### **chainloader**

указывает, что в качестве файла первой стадии (stage 1 file) будет загружен другой файл. Значение "+1" эквивалентно 0+1, что означает загрузку одного сектора, начиная с сектора 0, то есть загрузку первого сектора устройства, указанного в root или rootnoverify.

Теперь вы имеете некоторое представление о том, что вы можете найти в стандартном файле /boot/grub/grub.conf (или /boot/grub/menu.lst). Существует множество других команд GRUB для предоставления обширного контроля за процессом загрузки, а также для помощи в установке grub и выполнении других задач. Вы можете узнать больше о них в руководстве GRUB, которое должно быть доступно в вашей системе по команде [info grub](#).

Теперь, когда у нас есть файл конфигурации GRUB нам необходимо создать загрузочную дискету, чтобы протестировать его. Наипростейший способ сделать это -- использовать команду **grub-install** как показано в Листинге 8. Если вы устанавливаете GRUB на

дискету или в раздел, то вы должны сначала размонтировать это устройство. Но это не требуется, если вы устанавливаете GRUB в MBR жесткого диска, поскольку вы монтируете только разделы (`/dev/hda1`, `/dev/hda2` и т. д.), а не весь жесткий диск (`/dev/hda`).

### Листинг 8. Установка GRUB на дискету.

```
[root@lyrebird root]# umount /dev/fd0
umount: /dev/fd0: not mounted
[root@lyrebird root]# grub-install /dev/fd0
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.
(fd0)   /dev/fd0
(hd0)   /dev/hda
(hd1)   /dev/hdc
(hd2)   /dev/sda
```

Замечание: Вы также можете использовать имя устройства GRUB (`fd0`) вместо `/dev/fd0`, но если вы это делаете, то должны заключить его в кавычки, чтобы избежать его интерпретации командной оболочкой. Например:

```
grub-install '(fd0)'
```

Если вы начали работу с пустой дискетой, и уже смонтировали ее, то вы увидите, что она осталось пустой. Произошло только то, что GRUB записал измененный загрузчик первой стадии в первый сектор дискеты. Это не отображается в файловой системе. Этот загрузчик первой стадии загружает загрузчик второй стадии и конфигурационный файл с вашего жесткого диска. Попытайтесь загрузиться с дискеты и вы увидите весьма небольшую активность работы с нею перед тем, как отобразится меню.

Карта устройств покажет вам как GRUB подгоняет свое внутреннее представление ваших дисков (`fd0`, `hd0`, `hd1`) к представлению Linux (`/dev/fd0`, `/dev/hda`, `/dev/hdb`). В системе с одним или двумя IDE жесткими дисками и, может быть, с дисководом это, возможно, будет корректным. Если карта устройств уже существует GRUB воспользуется ею без проверки. Если вы просто добавили новый диск и хотите выполнить генерацию новой карты устройств, то к команде `grub-install` следует добавить опцию `--resize`. Например,

Сразу же после тестирования вашей дискеты, вы готовы к установке GRUB в MBR вашего жесткого диска. Для первого жесткого диска IDE вам следует использовать:

```
grub-install /dev/hda
или
grub-install '(hd0)'
```

Чтобы установить его в загрузочную запись раздела 11, используйте:

```
grub-install /dev/hda11
или
grub-install '(hd0,10)'
```

Запомните, что GRUB нумерует с 0.

## Обновления системы

Большинство дистрибутивов предоставляют инструменты обновления систем. Эти инструменты обычно осведомлены об установленном загрузчике и часто обновляют конфигурационный файл автоматически. Если вы собрали свое собственное ядро или предпочитаете использовать конфигурационные файлы с нестандартным именем или расположением, то вам может потребоваться обновить конфигурационный файл самостоятельно.

- Если вы используете LILO, то вы **должны** выполнить команду `lilo`, и не важно обновили ли вы конфигурационный файл или произвели такие изменения, как добавление жесткого диска или удаление раздела.
- Если вы используете GRUB, вы можете отредактировать файл `/boot/grub/grub.conf` чтобы произвести изменения, а загрузчик второй стадии GRUB прочтает этот файл при следующей перезагрузке. Обычно вам не нужно переустанавливать GRUB только потому, что вы добавили новое ядро. Однако, если вы передвигаете раздел или добавляете диски, то вам может понадобиться переустановить GRUB. Запомните, что загрузчик первой стадии очень мал, потому как он просто содержит список адресов блоков загрузчика второй стадии. При передвижении раздела адресация меняется, поэтому первая стадия больше не сможет обнаружить вторую стадию. Далее мы опишем некоторые стратегии восстановления, а также обсудим загрузчик стадии 1.5 GRUB.

## Восстановление

Теперь мы рассмотрим что может пойти не так при тщательно подогнанных установках загрузки, особенно когда вы используйте множественную загрузку. Первое что следует запомнить -- не паникуйте. Обычно восстановление это всего лишь несколько шагов. Мы предоставим вам несколько стратегий, которые помогут вам во многих кризисных ситуациях.

Эти стратегии и инструменты покажут вам, что любой, имеющий физический доступ к машине обладает значительными возможностями. Подобно этому любой, имеющий доступ к командной строке grub, имеет доступ к файлам вашего компьютера без учёта соглашений о правах доступа или любых других средств безопасности, предоставляемых работающей операционной системой. Имейте это ввиду при выборе загрузчика. Выбор между LILO и GRUB во многом основан на личных предпочтениях. Основываясь на том, что вы уже знали, а также на том о чём было рассказано, вы должны уже быть подготовлены к выбору загрузчика, который наиболее соответствует вашим нуждам и стилю работы.

### Установка других систем может повредить ваш MBR.

Когда-нибудь при установке операционной системы вы случайно можете перезаписать ваш MBR. Некоторые системы, такие как DOS и Windows всегда устанавливают свой собственный MBR. В таком случае обычно это очень легко восстановить. Если вы выработали привычку создавать загрузочную дискету каждый раз при запуске lilo или перезагрузке (GRUB), то все очень легко. Просто загрузите Linux с вашей загрузочной дискеты и перезапустите lilo или grub-install.

Если так случилось, что у вас нет загрузочной дискеты, но есть хоть какой-то дистрибутив Linux, то обычно вы можете загрузиться с установочного диска в режиме восстановления (recovery mode). Если вы сделаете это, то корневая файловая система вашего диска будет или смонтирована в некоторую точку восстановления (recovery point), или диск не будет смонтирован вообще. Вы можете использовать команду `chroot` чтобы сделать эту добавочную точку вашим корневым каталогом (`/`). Затем запустите lilo или grub-install чтобы создать новую загрузочную дискету или переписать MBR. Я обычно предпочитаю сначала создать загрузочную дискету и загрузиться с неё, чтобы перед перезаписью MBR убедиться что с моими настройками загрузчика всё в порядке, но вы можете быть более смелыми чем я.

Листинг 9 показывает пример использования рабочего окружения, которое мы создали в наших предыдущих примерах конфигурирования. В этом примере, я загрузился с загрузочного диска Red Hat Enterprise Linux, который смонтировал /dev/hda11 в /mnt/sysimage. Большинство дистрибутивов в режиме восстановления выдают вам большой экран с приглашением командной строки, а не графическое окно, с которым вы привыкли работать. Можете считать, что это окно терминала, которое вы открыли от имени суперпользователя. Другими словами будьте очень осторожны записывая что-либо на жесткий диск. В Листинге 9 вводимое пользователем выделено **жирным шрифтом**.

#### Листинг 9. Использование диска для восстановления и chroot.

```
sh-3.00# chroot /mnt/sysimage
sh-2.05b# lilo
Added linux *
Added WIN-XP
sh-2.05b# grub-install '(fd0)'
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.
(fd0)   /dev/fd0
(hd0)   /dev/hda
(hd1)   /dev/hdc
(hd2)   /dev/sda
sh-2.05b#
```

Как только загрузочная дискета будет создана, нажмите **ctrl-d**, чтобы выйти из среды chroot, а затем перезагрузите компьютер, не забыв удалить установочный носитель. Если у вас нет на руках установочного CD или DVD, то существует множество CD для восстановления и Live-CD Linux, доступных в сети, а также несколько дискет и USB-дисков.

Хотя это выходит за границы данного учебника, но вам может быть будет полезно узнать, что можно иметь MBR, загружающий систему Windows 2000 или Windows XP, и установить Lilo или GRUB в загрузочную запись раздела. Программа загрузки ntldr также может загружать другие цепочки загрузочных секторов, хотя ее настройка -- дело не простое. Вам понадобится скопировать загрузочный сектор на Windows-раздел и изменить скрытый файл boot.ini чтобы это сработало.

#### Вы переместили раздел.

Если вы переместили раздел и забыли про настройки загрузки, то могут возникнуть некоторые проблемы. Обычно LILO или GRUB отказываются загружаться. LILO возможно выводит 'L', что говорит о том, что первая стадия загрузки прошла, и на этом загрузка остановилась. GRUB выводит сообщение об ошибке. А произошло следующее: загрузчик первой стадии, у которого есть список секторов которые нужно загрузить, чтобы перейти к стадии 2, пытается загрузить сектора, расположенные по этим адресам, но они уже не содержат загрузочные сигнатуры второй стадии. Если вы создали загрузочную дискету используя описанные выше методы, то помните: все что и lilo и grub-install размещают на дискете -- это только загрузочный сектор, так что ваша загрузочная дискета скорее всего не поможет. Как и в предыдущем примере, вы можете попытаться загрузить различные окружения для восстановления и переделать загрузочную дискету с LILO или GRUB. Затем перезагрузиться, проверить вашу систему и вновь установить загрузчик в MBR.

Вы могли заметить, что наши примеры конфигурирования использовали метки разделов.

Например,

```
append="hdd=ide-scsi root=LABEL=RHEL3"  
или  
kernel /boot/vmlinuz-2.4.20-31.9 ro root=LABEL=RH9 hdd=ide-scsi
```

Я часто использую метки подобные этим, чтобы избежать проблем при перемещении разделов. В этом случае вам все еще нужно обновить конфигурационный файл GRUB или LILO, но не нужно изменять /etc/fstab. Это может быть особенно полезно, если я создаю образ раздела на одном компьютере и восстанавливаю его в другом месте на другом компьютере.

### **Использование раздела /boot.**

Другим подходом к восстановлению или, возможно, избежания этого является использование отдельного раздела для /boot. Этот раздел не требует очень много места, возможно 100МБ или около того. Разместите этот раздел там, где он не будет требовать перемещения и где его номер не будет меняться при добавлении или перемещении других разделов. В смешанной системе Windows и Linux хорошим выбором для раздела /boot будет /dev/hda2.

Другой причиной для создания раздела /boot может быть случай, когда ваша корневая файловая система не поддерживается загрузчиком. Например, считается общепринятым форматировать раздел /boot в ext2 или ext3, тогда как для корневого раздела (/) использовать LVM.

Если у вас установлено несколько дистрибутивов, **не используйте** единственный раздел /boot для них всех. Не забудьте настроить LILO или GRUB на загрузку с раздела, который в последствии будет смонтирован как /boot. Помните также, что программа обновления дистрибутива обычно обновляет конфигурацию GRUB или LILO именно для этой системы. В среде со множеством операционных систем вы можете пожелать связать с одной из них раздел /boot и сделать ее главной, а во всех остальных -- при необходимости править конфигурационные файлы вручную. В качестве другого подхода можно использовать установку загрузчика для каждой системы в загрузочную запись ее раздела, а ваша главная система просто будет по цепочке загружать загрузочные записи разделов отдельных систем, и в результате при загрузке получим две стадии, каждая со своим меню.

### **Создание самодостаточной загрузочной дискеты.**

И наконец, давайте более пристально посмотрим на настройку GRUB и то как сделать самодостаточную загрузочную дискету, которая сможет предоставить командую строку GRUB, вне зависимости от того, что произошло с вашим жестким диском.

Вспомните все, что мы говорили о цилиндрах жесткого диска. Хотя вы можете считать, что цилиндры в современных жестких дисках это некая фикция, многие части вашей файловой системы о них не забыли. В частности вы обнаружите, что разделы используют целые числа цилиндров и выравнены по границам цилиндров. В разделах многие файловые системы также работают с пространством в единицах цилиндров. Во многих UNIX и Linux системах структура файловой системы хранится в *суперблоке*, являющимся первой единицей размещения файловой системы. Для таких файловых систем как ext2 или ext3 и довольно больших жестких дисков пространство разбивается на несколько секций с копией суперблока в начале каждой секции. Это может помочь при восстановлении, если вы случайно испортили границы раздела в такой программе, как fdisk.

Еще одним плюсом в пользу работы с цилиндрами является то, что в начале диска имеется некоторое пространство, идущее сразу же за MBR. GRUB использует это, размещая в нем загрузчик стадии 1.5 или в другом подобном не используемом месте раздела где это возможно. Загрузчик стадии 1.5 распознает файловую систему раздела, содержащего стадию

2, так что это в чем-то более устойчиво по отношению к проблемам, связанным с перемещением файлов.

Все это хорошо и прекрасно, но как это связано с загрузочной дискетой? Что ж, на дискете не так много места и нет даже понятия цилиндров, поэтому если вы хотите загрузить и стадиу 1 и стадиу 2 GRUB с дискеты, вам потребуется установить стадиу 1, а затем скопировать стадиу 2 в сектора, следующие за загрузочным сектором. Листинг 10 содержит пример того, как это можно сделать. Используйте чистую дискету, поскольку в результате производимых действий все данные на ней будут уничтожены. Вам следует скопировать файлы поставляемые с вашим дистрибутивом grub, а не из вашего каталога /boot/grub, поскольку /boot/grub/stage2 изменяется для работы с разделами вашего жесткого диска. Вы сможете найти оригинальные файлы stage1 и stage2 в подкаталоге /usr/share/grub. В нашем примере они расположены в /usr/share/grub/i386-redhat.

#### Листинг 10. Создание загрузочной дискеты GRUB.

```
[root@lyrebird root]# ls /usr/share/grub  
i386-redhat  
[root@lyrebird root]# cd /usr/share/grub/i386-redhat  
[root@lyrebird i386-redhat]# ls -l st*  
-rw-r--r-- 1 root root 512 Aug 3 2004 stage1  
-rw-r--r-- 1 root root 104092 Aug 3 2004 stage2  
[root@lyrebird i386-redhat]# dd if=stage1 of=/dev/fd0 bs=512 count=1  
1+0 records in  
1+0 records out  
[root@lyrebird i386-redhat]# dd if=stage2 of=/dev/fd0 bs=512 seek=1  
  
203+1 records in  
203+1 records out
```

Если вы отформатировали вашу дискету перед тем как выполнить это, и теперь пытаетесь смонтировать ее, то команда mount выдаст ошибку. Копирование stage2 сразу же вслед за загрузочным сектором дискеты (seek=1) уничтожит файловую систему на ней.

Если теперь вы загрузитесь с этой дискеты, то вы отметите задержку при загрузке второй стадии с дискеты. Вы можете загрузиться с этой дискеты на любом ПК, и наличие на нем системы Linux не обязательно. Когда вы загрузитесь с дискеты вы увидите приглашение командной строки GRUB. Нажмите клавишу TAB чтобы увидеть список доступных команд. Для получения справки о команде с именем *commandname* используйте [help commandname](#). Листинг 11 показывает пример командной строки GRUB.

#### Листинг 11. Командная строка GRUB.

```
GRUB version 0.93 (640K lower / 3072K upper memory)  
  
[ Minimal BASH-like line editing is supported. For the first word, TAB  
lists possible command completions. Anywhere else TAB lists the possible  
completions of a device/filename.]  
  
grub>  
Possible commands are: blocklist boot cat chainloader clear cmp color configfi  
le debug device displayapm displaymem dump embed find fstest geometry halt help  
hide impsprobe initrd install ioprobe kernel lock makeactive map md5crypt modu  
le modulenounzip pager partnew parttype password pause quit read reboot root ro  
otnoverify savedefault serial setkey setup terminal terminfo testload testvbe u
```

```

nhide uppermem vprobe

grub> help rootnoverify
rootnoverify: rootnoverify [DEVICE [HDBIAS]]
    Similar to `root', but don't attempt to mount the partition. This
    is useful for when an OS is outside of the area of the disk that
    GRUB can read, but setting the correct root device is still
    desired. Note that the items mentioned in `root' which derived
    from attempting the mount will NOT work correctly.

grub> find /boot/grub/grub.conf
(hd0,2)
(hd0,6)
(hd0,7)
(hd0,10)
(hd1,7)

grub>

```

В этом примере мы смогли узнать, что на четырех различных разделах первого диска имеется файл конфигурации GRUB и еще один на втором жестком диске. Мы могли бы загрузить меню GRUB с одного из них, использовав команду configfile. Например:

```
configfile (hd0,2)/boot/grub/grub.conf
```

Это приведет к загрузке меню этого конфигурационного файла и мы сможем загрузить систему с этой точки. Вы можете найти эти команды grub в руководстве GRUB. Наберите **info grub** в окне терминала, чтобы открыть руководство.

И последнее, прежде чем мы закончим с GRUB. Мы упоминали, что файл второй стадии GRUB уничтожает файловую систему на дискете. Если вы хотите получить GRUB дискету для восстановления, загружающую файлы GRUB, включая конфигурационный файл, с дискеты, то вы можете сделать это, выполнив следующие шаги:

- используйте команду **mkdosfs** для создания на дискете файловой системы DOS FAT и опцию **-R** для резервирования достаточного количества секторов для файла второй стадии.
- Смонтируйте дискету
- Создайте на дискете каталог `/boot/grub`
- Скопируйте файлы GRUB stage1, stage2 и `grub.conf` в каталог `boot/grub` на дискете. Скопируйте также, если хотите, файл с фоновым изображением (заставкой).
- Отредактируйте файл `grub.conf` на дискете так, чтобы он ссылался на файл с изображением на дискете.
- Размонтируйте дискету
- Используйте командную оболочку **grub** для установки GRUB на дискету, используя команды GRUB `root` и `setup`.

Мы проиллюстрировали это в Листинге 12.

#### **Листинг 12. Установка GRUB на дискету с файловой системой.**

```
[root@lyrebird root]# mkdosfs -R 210 /dev/fd0
mkdosfs 2.8 (28 Feb 2001)
[root@lyrebird root]# mount /dev/fd0 /mnt/floppy
[root@lyrebird root]# mkdir /mnt/floppy/boot
[root@lyrebird root]# mkdir /mnt/floppy/boot/grub
```

```
[root@lyrebird root]# cp /boot/grub/stage1 /mnt/floppy/boot/grub
[root@lyrebird root]# cp /boot/grub/stage2 /mnt/floppy/boot/grub
[root@lyrebird root]# cp /boot/grub/splash* /mnt/floppy/boot/grub
[root@lyrebird root]# cp /boot/grub/grub.conf /mnt/floppy/boot/grub
[root@lyrebird root]# umount /dev/fd0
[root@lyrebird root]# grub
Probing devices to guess BIOS drives. This may take a long time.

GRUB version 0.93 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ]

grub> root (fd0)
Filesystem type is fat, using whole disk

grub> setup (fd0)
Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/fat_stage1_5" exists... no
Running "install /boot/grub/stage1 (fd0) /boot/grub/stage2 p /boot/grub/grub.c
onf "... succeeded
Done.
```

Рассмотренные инструменты это все, что необходимо вам для восстановления после различных ошибок, которые могут возникнуть при использовании загрузчика.

| [предыдущая](#) | [следующая](#)

## Сборка и установка программ

В этом разделе рассматривается материал темы 1.102.3 для экзамена LPI 101  
Администрирование для начинающих (LPIC-1). Рейтинг темы 5.

В этом разделе вы узнаете как компилировать и устанавливать программы из исходных текстов (source) [Прим.пер.: в русскоязычной Linux-среде для обозначения исходных текстов программ часто используются жаргонные слова *исходники* и *сырцы*]. Вы научитесь распаковывать типичные архивы с исходными текстами и настраивать файлы Makefile.

Для чего может понадобиться устанавливать программу из исходных текстов? Среди распространенных причин:

1. Вам требуется программа, не входящая в ваш дистрибутив.
2. Вам нужна программа, которая доступна только в виде исходных текстов.
3. Вам нужна некая функция программы, которая может быть включена только после перекомпиляции программы из исходных текстов.
4. Вы хотите узнать о том как программа работает или принять участие в ее разработке.

Все они являются весьма вескими причинами для установки программы из исходных текстов.

## Загрузка и распаковка

Вне зависимости от вашей мотивации сборки из исходных текстов, прежде чем перейти непосредственно к самой сборке вам потребуется заполучить исходные тексты. Вы можете найти пакет на сайте, разработанном для размещения проектов, вроде сайта SourceForge.net от Open Source Technology Group, или на сайте посвященному только этому проекту.

В этом разделе мы в основном будем рассматривать пакеты распространяемые в виде так называемых *сжатых tar-файлов* (*tarball*). Команда **tar** (от *Tape ARchive* -- ленточный архив) используется для создания *архивов* с файлами из древа каталогов. Несмотря на название, tar-файлы могут располагаться на любом носителе. Хранение их на диске позволяет выполнять такие операции, как например удаление части архива, что невозможно сделать на ленте. Команда tar сама по себе не сжимает данные, она просто соединяет их в один файл в специальном формате, позволяющем сохранить как сами файлы, так и все права доступа, запреты и структуру каталогов. Команда tar может быть использована совместно с программой сжатия, обычно **gzip** или **bzip2** для создания сжатого архива, который экономит дисковое пространство, а также уменьшает время передачи файла. Вот этот получившийся архив и называют *tarball*.

Помимо простых архивов исходные тексты отдельных программ могут быть упакованы для вашего дистрибутива в *пакеты с исходными текстами* (*source package*), такие как RPM с исходными текстами (или SRPM). Мы обсудим управление пакетами в этом учебнике позднее. А сейчас просто не забудьте поискать пакет с исходными текстами для вашего дистрибутива, если вы его найдете, то это обычно значительно облегчает компиляцию программы, поскольку он уже подогнан к структуре файловой системы вашего дистрибутива.

Перед загрузкой узнайте как можно больше о пакете. Если доступна информация по сборке или установке, то просмотрите ее на предмет необходимости других пакетов, чтобы было возможно скомпилировать этот. Часто вам необходимо установить еще и несколько библиотек, а возможно и инструмент разработки, чтобы успешно скомпилировать выбранную программу. Это особенно часто бывает, если ваша программа использует любые графические инструментарии. Иногда только после запуска процесса сборки выясняется, что необходим еще какой-то пакет. Не волнуйтесь - это не редкость. Вам просто нужно найти и установить отсутствующий пакет и продолжать попытки пока все требуемые пакеты не будут установлены.

Обычно для загрузки вы можете использовать ваш браузер или, возможно, программу для ftp. Ваш пакет вероятно будет иметь имя, имеющее одно из перечисленных окончаний: tar, tar.gz, tar.Z, tgz, или tar.bz2. Иногда вы будете загружать пакеты, используя CVS (Concurrent Version System -- Система параллельных версий). Примером может быть GNU GRUB 2 от Free Software Foundation (смотри [Ресурсы](#)). В этом случае загруженные вами исходные тексты будут уже распакованы. Изредка вы можете найти файлы с расширением .zip, говорящим о том, что это файл zip.

## Сжатые tar файлы

Сжатый tar файлы или *tarball* -- это наиболее популярная форма распространения исходных текстов, если не используется система управления пакетами вроде RPM от Red Hat или управление пакетами в Debian. Они создаются при помощи команды **tar**, архивирующей древо каталогов и все файлы из него в один архивный файл. Обычно результат может быть сжат при помощи некой программы сжатия, как правило используется одна из следующих **compress**, **gzip** или **bzip2**. Поскольку архивирование и сжатие это наиболее общие операции, команда GNU tar, имеющаяся во многих дистрибутивах Linux может также самостоятельно применять сжатие и развертывание с использованием compress, gzip или bzip2. Если имеющаяся у вас версия tar не поддерживает указанные типы сжатия, то UNIX и Linux системы весьма хорошо используют конвейер, позволяющий нескольким командам работать над одними и теми же данными по очереди, так что двухступенчатый процесс может выполняться вручную, а tar сделает это для вас в любом случае.

Для иллюстрации предположим, что мы загрузили проект Dr. Geo interactive geometry. Во время создания этого урока текущая версия содержалась в файле drgeo-1.1.0.tar.gz. Расширение gz говорит о том, что этот файл сжат при помощи gzip. Сначала мы покажем, как получить tar файл из сжатого файла, а затем как извлечь оттуда отдельные файлы. Затем мы

покажем вам как распаковать и извлечь файлы одной командой или при помощи конвейера. Для того, чтобы просто извлечь tar-архив мы используем команду **gunzip** как показано в Листинге 13.

### Листинг 13. Распаковка пакета с исходными текстами Dr Geo.

```
[ian@localhost ~]$ ls drgeo*
drgeo-1.1.0.tar.gz
[ian@localhost ~]$ gunzip drgeo-1.1.0.tar.gz
[ian@localhost ~]$ ls drgeo*
drgeo-1.1.0.tar
```

Отметим, что наш файл .tar.gz теперь заменен исходным .tar файлом. Чтобы распаковать tar-файл для других упомянутых расширений вам следует использовать следующие команды (соответственно)

```
uncompress drgeo-1.1.0.tar.Z
gunzip drgeo-1.1.0.tar.Z
gunzip drgeo-1.1.0.tar.gz
gunzip drgeo-1.1.0.tgz
bunzip2 drgeo-1.1.0.tar.bz2
```

Вы можете отметить, что gunzip может работать с .Z, .tar.gz и .tgz. В действительности же, в вашей системе может вообще не быть программ compress и uncompress.

Для извлечения файлов из tar архива используется команда **tar**. Стандартная форма: **tar -xvf имя\_файла.tar**, приведена в Листинге 14. Опционально вы можете ограничить вывод, используя малый фильтр для разбивки вывода на страницы.

### Листинг 14. Извлечение файлов из архива Dr Geo.

```
[ian@localhost ~]$ tar -xvf drgeo-1.1.0.tar |more
drgeo-1.1.0/
drgeo-1.1.0/po/
drgeo-1.1.0/po/ChangeLog
drgeo-1.1.0/po/Makefile.in.in
drgeo-1.1.0/po/POTFILES.in
drgeo-1.1.0/po/drgeo.pot
drgeo-1.1.0/po/az.po
drgeo-1.1.0/po/ca.po
drgeo-1.1.0/po/cs.po
...
```

Опция **-X** говорит tar, что нужно извлечь файлы из архива. Опция **-V** говорит tar, что нужно вывести подробный листинг обрабатываемых файлов. И опция **-f** вместе с именем файла (в данном случае drgeo-1.1.0.tar) говорит tar, что архивные файлы извлекаются из файла.

Правильно сделанный пакет в процессе разархивирования создаст каталог, в котором будут сохранены файлы пакета. В нашем примере это каталог drgeo-1.1.0. Иногда пакет этого не делает, и потому вы можете захотеть посмотреть структуру пакета перед тем, как выполнить его распаковку с риском получить огромное количество файлов прямо в домашнем каталоге. Чтобы выполнить проверку используйте команду tar с опцией **-t**, отображающей таблицу

содержимого, вместо **-X**, производящей разархивирование. Если вы также опустите опцию **-V**, то все равно полученного листинга будет достаточно для того, чтобы понять какие файлы будут созданы и будет ли создан новый каталог или все будет выгружено в текущий.

Теперь, когда мы увидели как разархивировать сжатые архивы tar в два шага, вы возможно удивитесь заявлению, что все это можно сделать за один шаг. Да можно. Если вы добавите опцию **-Z** к команде tar, то она сможет распаковать и разархивировать архивы, сжатые gzip при помощи одной команды. Например:

```
tar -zxvf drgeo-1.1.0.tgz  
или  
tar -zxvf drgeo-1.1.0.tar.Z
```

Для выполнения того же для архивов сжатых при помощи bzip2, вместо опции **-Z** используйте опцию **-j**. Например:

```
tar -jxvf drgeo-1.1.0.tar.bz2
```

Вы можете также использовать опцию **-c** с любой из указанных выше команд сжатия для направления распакованных файлов в стандартный вывод, который затем преобразуется в стандартный вход для команды tar. Отметим, что это оставляет ваш оригинальный файл неизменным, а не распаковывает его в большой .tar файл. Вот несколько примеров:

```
bunzip2 -c drgeo-1.1.0.tar.bz2 | tar -xvf -  
uncompress -c drgeo-1.1.0.tar.Z | tar -xvf -  
gunzip -c drgeo-1.1.0.tar.Z | tar -xvf -  
gunzip -c drgeo-1.1.0.tar.gz | tar -xvf -  
gunzip -c drgeo-1.1.0.tgz | tar -xvf -
```

### Примечания:

1. Значение **-** вместо имени архивного файла заставляет tar использовать стандартный ввод для архивов. Ваша версия tar может делать это по умолчанию, в этом случае вам нет нужды указывать опцию **-f** вообще. Просто опустите в конце вышеперечисленных команд завершающие **f -**.
2. Команда **zcat** выполняет те же функции, что и **gunzip -c**.

### Древо CVS

Иногда код необходимого вам проекта не упакован в архив, а доступен через CVS (Concurrent Version System -- Система Параллельных Версий). В качестве примера можно привести проект GRUB 2, обсуждавшийся нами в разделе [Менеджеры загрузки](#). В Листинге 15 приведен пример.

### Листинг 15. Загрузка GRUB2 через CVS.

```
[ian@attic4 ~]$ export CVS_RSH="ssh"  
[ian@attic4 ~]$ cvs -z3 -d:ext:anoncvs@savannah.gnu.org:/cvsroot/grub co grub2  
cvs server: Updating grub2  
U grub2/.cvsignore  
U grub2/AUTHORS  
U grub2/COPYING  
U grub2/ChangeLog  
U grub2/DISTLIST  
U grub2/INSTALL  
U grub2/Makefile.in  
U grub2/NEWS
```

...

Команда `export` говорит CVS как соединиться с удаленным сервером (используя безопасную оболочку (`secure shell`) или `ssh` в данном случае). Команда `cvs` проверяет (опция `co`) проект `grub2`. Вы найдете все файлы проекта в каталоге `grub2`, который команда `cvs` создаст для вас.

## Zip файлы

Изредка вы можете обнаружить пакеты с исходными текстами в виде `zip` файлов. Это может иметь место для пакетов, которые работают в Windows также как и в Linux системах.

Оригинальная программа PKZIP была разработана для систем DOS компанией PKWARE, Inc., а теперь доступна для нескольких платформ. Многие системы Linux имеют версию, созданную Info-ZIP.

Листинг 16 показывает как использовать команду `unzip` для распаковки исходных текстов хранителя экрана, демонстрирующего выворачивание сферы на изнанку (`sphere eversion`).

### Листинг 16. Распаковка исходных текстов `sphere eversion` из `zip`.

```
[ian@attic4 ~]$ unzip sphereEversion-0.4-src.zip
Archive:  sphereEversion-0.4-src.zip
  creating: sphereEversion-0.4-src/
  inflating: sphereEversion-0.4-src/Camera.h
  inflating: sphereEversion-0.4-src/drawutil2D.h
  inflating: sphereEversion-0.4-src/drawutil.h
  inflating: sphereEversion-0.4-src/fontdata.h
  inflating: sphereEversion-0.4-src/fontDefinition.h
  inflating: sphereEversion-0.4-src/generateGeometry.h
  inflating: sphereEversion-0.4-src/global.h
  inflating: sphereEversion-0.4-src/mathutil.h
  inflating: sphereEversion-0.4-src/Camera.cpp
  inflating: sphereEversion-0.4-src/drawutil2D.cpp
  inflating: sphereEversion-0.4-src/drawutil.cpp
  inflating: sphereEversion-0.4-src/fontdata.cpp
  inflating: sphereEversion-0.4-src/generateGeometry.cpp
  inflating: sphereEversion-0.4-src/main.cpp
  inflating: sphereEversion-0.4-src/mathutil.cpp
  inflating: sphereEversion-0.4-src/README.TXT
  inflating: sphereEversion-0.4-src/Makefile
```

## Сборка (компиляция) программ

Теперь, когда исходные тексты распакованы в древо каталогов, рассмотрим как скомпилировать программу или программы.

### Инспекция исходных текстов

Перед тем, как запустить компиляцию, вы должны просмотреть что было распаковано. В частности просмотреть документацию по установке. Обычно это файл `README` или `INSTALL`, возможно, что и оба, расположенные в каталоге вашего нового проекта. Если пакет разработан для нескольких платформ, вы можете найти файлы, специфичные для некой платформы, такие как `README.linux` или `INSTALL.linux`.

### Конфигурирование

Весьма часто в главном каталоге исходных текстов встречается скрипт `configure`. Это скрипт разработан для настройки `Makefile`, который подгоняется под вашу систему. Он обычно

генерируется разработчиком, с использованием программы GNU autoconf. Скрипт configure проверяет вашу систему на наличие всех нужных инструментов и совместимость. Полученный в результате Makefile или несколько таких файлов, скомпилируют проект на вашей конкретной системе.

Сложный конфигурационный скрипт может проверять множество аспектов вашей системы, включая такие вещи, как тип процессора, является ли он 32-х или 64-х битным и так далее. Простой конфигурационный сценарий не делает почти ничего, кроме создания файлов Makefile.

Если у вас нет файла с именем *configure* в главном каталоге проекта, то просмотрите документацию на предмет другого способа настройки параметров сборки. Если же такой файл у вас есть, то перейдите в главный каталог проекта и выполните.

```
./configure --help
```

Эта команда выдаст справку о доступных опциях конфигурации. Многие из них, такие как *--prefix*, встречаются в большинстве конфигурационных скриптов. Некоторые скорее всего будут специфичны для конкретного компилируемого проекта. Найдите те, который вам нужно изменить.

**Примечание:** Если ваш проект не имеет конфигурационного скрипта, то возможно он имеет файл Makefile, работающий на большинстве платформ, или установочный процесс в какой-то другой форме. Например, пакет, использующий только скрипты Python файлы с данными может не требовать сборки, так что он может иметь просто сценарий для установки.

В учебнике для темы 104 мы рассмотрим стандарт иерархии файловых систем (FHS – Filesystem Hierarchy Standard). А сейчас отметим, что локальные программы должны иметь исполняемые файлы, сохраненные в древе /usr/local в /usr/local/bin и man-страницы в /usr/local/man. Скрипты configure вероятно имеют опцию *--prefix*, указывающую место установки. Если программа не совместима с FHS, то вам может потребоваться указать эту опцию при запуске скрипта configure. Если вы компилируете программу для замены установленной версии, то вам может потребоваться установить ее, указав каталог в /opt или /usr в качестве префиксов.

В добавок к возможному указанию префиксов вы можете обнаружить другие опции, связанные с размещением специфичных компонентов, таких как *--mandir* или *--infodir* для указания расположения man и info страниц соответственно.

После просмотра возможных опций и определения того, что вам следует изменить, выполните скрипт *configure*, добавив все необходимые опции. Не забудьте добавить *.* / перед командой configure, поскольку каталог вашего проекта вероятно не будет указан в переменной path. Например, вы можете выполнить

```
./configure  
или  
./configure --prefix /usr/local
```

После запуска configure обычно вы видите сообщения, рассказывающие о типе используемой вами системы и о том, какие необходимые инструменты установлены, а какие нет. Если все идет хорошо, то к концу процесса конфигурирования вы должны получить созданный Makefile.

### **config.cache**

По завершении выполнения скрипта configure, он сохраняет информацию о конфигурации в файле с названием *config.cache*, расположенном в том же каталоге, что и сам скрипт configure.

Если вам необходимо запустить `./configure` вновь, то убедитесь, что прежде вы удалили файл config.cache (используйте команду `rm`), поскольку configure будет использовать настройки из config.cache, если он существует, не производя повторной проверки вашей системы.

Листинг 17 содержит часть выведенных сообщений выполнения configure для пакета Dr Geo, который мы распаковали ранее.

### Листинг 17. Конфигурирование Dr Geo.

```
[ian@localhost ~]$ cd drgeo-1.1.0
[ian@localhost drgeo-1.1.0]$ ./configure | less
checking for XML::Parser... ok
checking for iconv... /usr/bin/iconv
checking for msgfmt... /usr/bin/msgfmt
checking for msgmerge... /usr/bin/msgmerge
checking for xgettext... /usr/bin/xgettext
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether to enable maintainer-specific portions of Makefiles... no
checking for g++... g++
checking for C++ compiler default output file name... a.out
checking whether the C++ compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
...
checking for guile... /usr/bin/guile
checking for guile-config... no
configure: error: guile-config required but not found
```

Скрипт configure проверяет несколько программ преобразования графики, являющихся частью пакета netpbm. Имеется предупреждение, поскольку одна из необходимых программ преобразования не найдена в системе. Есть также предупреждение о том, что использование префикса /usr/local требует прав суперпользователя (на этапе установки). Поскольку это первый запуск, то выводится несколько ошибок, связанных с файлом Makefile, который еще не существует, но появится если мы выполним configure вновь. И наконец скрипт configure сообщает об успешном завершении.

### Make и файлы Makefile

По завершении конфигурирования, вы должны получить файл в каталоге проекта с именем *Makefile*. Он называется *сборочный файл проекта*, программа с именем *make* используется для его обработки и сборки программы. Может также иметься несколько make-файлов в ваших подкаталогах.

Make-файл содержит *правила*, являющиеся инструкциями, которые сообщают программе make как собирать различные компоненты приложения. Файл также содержит *targets (цели)*, которые сообщают программе make, что именно компилировать. Программа make анализирует make-файл и определяет порядок в котором следует производить компиляцию (сборку). Например, если исполняемый файл создается из трех объектных файлов, то объектные файлы должны быть скомпилированы до того, как они будут объединены в исполняемый код. Make-файл может выполнять как компиляцию, так и инсталляцию

приложения. Назначения (targets) make-файла обычно доступны для нескольких функций, таких как:

#### **make**

без опций просто компилирует программу. Говоря технически, таким образом компилируется *назначение по умолчанию*, что обычно означает просто компиляцию программы из исходных текстов.

#### **make install**

устанавливает скомпилированную программу. Если вы производите установку в /usr/local, то вам могут потребоваться права суперпользователя (root).

#### **make clean**

удаляет файлы, созданные в процессе сборки.

#### **make all**

иногда используется для выполнения всех функций make-файла за раз. .

Обратитесь к документации проекта, чтобы узнать, есть ли дополнительные назначения (targets) или дополнительные элементы, которые могут понадобиться вам.

Теперь, когда ваш главный Makefile создан, используйте команду **make**, обычно без опций, для сборки выполняемых файлов, man-страниц и других частей программы. В зависимости от быстродействия вашего компьютера и сложности процесса, сборка может занять от одной-двух минуты до нескольких часов для сложных проектов.

Иногда сборка может не работать. Наиболее общие причины, это:

- Отсутствие необходимых пакетов
- Не та версия необходимых пакетов
- Не верные значения параметров, которые вы должно быть пропустили в configure или make.
- Отсутствие компилятора.
- Ошибки в скрипте configure или созданном Makefile.
- Ошибки в исходном коде.

В нашем примере с Dr Geo, одна из таких проблем была обнаружена на этапе конфигурирования, но это не обычный случай. По мере накопления опыта работы в Linux, вы сможете определить и исправить эти проблемы. Иногда вам придется обратиться к FAQ [Прим.пер.: Frequently Asking Questions -- ЧАсто задаваемые ВОпросы. В последнее время в Рунете это преобразуется в русскоязычное ЧАВО] или списку рассылки о поддержке данного пакета. В других случаях вам может понадобиться определить, что вы пропустили и установить это.

### **Установка**

Если при сборке все прошло хорошо, то вы готовы к установке. На этапе компиляции были собраны все необходимые файлы, но они все еще расположены не в том месте, чтобы быть готовыми к использованию. Например, бинарные файлы необходимо скопировать в /usr/local/bin, а man страницы в /usr/local/man и т.д.

Если вы не указывали опцию **--prefix**, то несколько файлов и каталогов скорее всего будет скопировано в древо /usr/local. Вам потребуются права суперпользователя для записи в древо /usr/local вашей файловой системы. Если вы вошли не как суперпользователь (root), то используйте команду **SU** для получения прав суперпользователя. Будет запрошен ввод пароля root. Затем используйте команду **make install**, чтобы установить только что собранную программу. Установка занимает от нескольких секунд до минут, в зависимости от размера программы. Мы привели часть выведенного при установке Dr Geo в Листинге 18.

## Листинг 18. Установка Dr Geo.

```
[ian@attic4 drgeo-1.1.0]$ su
Password:
[root@attic4 drgeo-1.1.0]# make install
Making install in po
make[1]: Entering directory `/home/ian/drgeo-1.1.0/po'
if test -n ""; then \
    /usr/local/share; \
else \
    /bin/sh ..../mkinstalldirs /usr/local/share; \
fi
installing az.gmo as /usr/local/share/locale/az/LC_MESSAGES/drgeo.mo
installing ca.gmo as /usr/local/share/locale/ca/LC_MESSAGES/drgeo.mo
installing cs.gmo as /usr/local/share/locale/cs/LC_MESSAGES/drgeo.mo
installing da.gmo as /usr/local/share/locale/da/LC_MESSAGES/drgeo.mo
installing de.gmo as /usr/local/share/locale/de/LC_MESSAGES/drgeo.mo
installing el.gmo as /usr/local/share/locale/el/LC_MESSAGES/drgeo.mo
installing en_CA.gmo as /usr/local/share/locale/en_CA/LC_MESSAGES/drgeo.mo
installing en_GB.gmo as /usr/local/share/locale/en_GB/LC_MESSAGES/drgeo.mo
...
/usr/bin/install -c drgeo /usr/local/bin/drgeo
/bin/sh ..../mkinstalldirs /usr/local/share/applications
/usr/bin/install -c -m 644 drgeo.desktop /usr/local/share/applications/drgeo.desktop
make[2]: Leaving directory `/home/ian/drgeo-1.1.0'
make[1]: Leaving directory `/home/ian/drgeo-1.1.0'
[root@attic4 drgeo-1.1.0]# exit
exit
[ian@attic4 drgeo-1.1.0]$
```

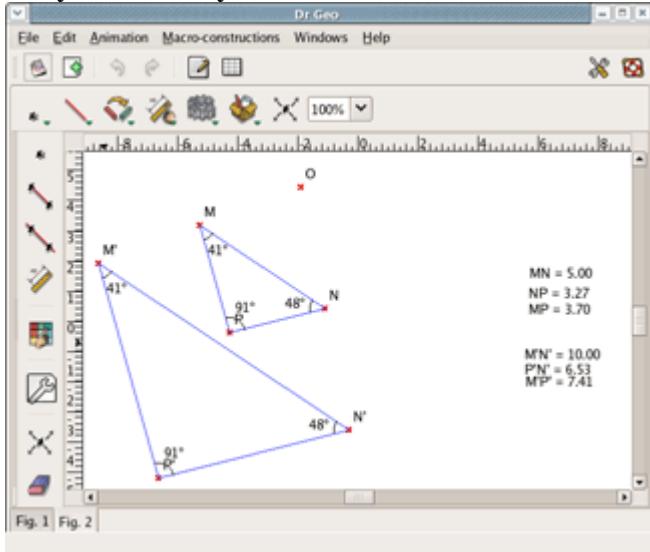
После копирования файлов, команда make install должна также убедиться, что все установленные файлы имеют корректные права доступа и разрешения. После завершения установки программа считается установленной и готовой к использованию, или готовой к предварительной настройке перед использованием.

**Замечание:** Если вы имеете права суперпользователя, то можно очень легко, допустив ошибку, нанести большой вред системе, поэтому не забудьте завершить режим суперпользователя командой **exit** или нажатием **ctrl-d** в командной оболочке bash.

## Запуск программ

Если ваша программа готова к запуску, то вы можете попробовать запустить ее, набрав в командной строке ее имя, для нашего примера **drgeo**. На Рисунке 1 показано окно Dr Geo, отображающее один из примеров, поставляемых вместе с программой.

**Рисунок 1. Запущенный Dr Geo**



Вот что еще вам следует сделать перед запуском программы.

- Прочтите `man` страницы если они есть в пакете. Попытайтесь выполнить `man Имя_программы`.
- Настроить конфигурационные файлы, например в `/etc`.
- Настроить автоматический запуск для такой программы, как демон сервера.

В этом разделе мы рассмотрели процесс установки программы из исходных текстов от самого начала и до конца. В следующих разделах мы поговорим о библиотеках, управлении библиотеками и пакетами, а также о том, как устанавливать их.

| [предыдущая](#) | [следующая](#)

### Управление библиотеками совместного доступа

В этом разделе рассматривается материал темы 1.102.4 экзамена LPI 101 Администрирование для начинающих (LPIC-1). Рейтинг темы 3.

В этом разделе вы узнаете как определить, от каких совместно используемых библиотек зависят программы. Вы узнаете где хранятся системные библиотеки. Установку пакетов, включая библиотеки совместного доступа, мы рассмотрим в следующих разделах этого учебника.

### Статические и динамические исполняемые файлы

В системах Linux имеется два типа исполняемых программ.

1. Исполняемые файлы *статической компоновки* (*Statically linked*) содержат все функции библиотек, которые необходимы им для работы. Все библиотечные функции включены в исполняемый файл. Это полные программы, которым не нужны внешние библиотеки для запуска. Одним из преимуществ статически скомпонованных программ является то, что они будут работать без установки зависимостей.
2. Исполняемые файлы *динамической компоновки* (*Dynamically linked*) намного меньше по размеру, поскольку они не полные, в том смысле, что для запуска им необходимы функции из внешних *совместно используемых* (*shared*) библиотек. Кроме того, что они меньше по размеру, динамическая компоновка позволяет пакету указать от каких библиотек он зависит, без необходимости включения этих библиотек в пакет. Использование динамической компоновки также позволяет многим рабочим программам совместно использовать одну копию библиотеки, вместо того, чтобы занимать память множеством копий одного и того же кода. По этим причинам

большинство программ на сегодняшний день используют динамическую компоновку.

Интересным примером типичной системы Linux является команда `ln` (/bin/ln), создающая связи между файлами или *жесткие (hard)* связи, или *мягкие (soft)* (или *символические (symbolic)*) связи (ссылки). Совместно используемые библиотеки часто используют символические ссылки между универсальным именем и конкретной версией библиотеки. При этом если нарушить некоторые ссылки на динамические библиотеки, то команда `ln` (с помощью которой эти ссылки можно быть бы восстановить) тоже становится неработоспособной. Во избежание таких случаев, системы Linux содержат статически скомпонованную версию программы `ln` под именем `sln` (/sbin/sln). Листинг 19 иллюстрирует большую разницу в размере между двумя этими программами.

### Листинг 19. Размеры sln и ln.

```
[ian@lyrebird ian]$ ls -l /sbin/sln; ls -l /bin/ln
-rwxr-xr-x    1 root      root        457165 Feb 23  2005 /sbin/sln
-rwxr-xr-x    1 root      root        22204 Aug 12  2003 /bin/ln
```

### Команда ldd

Не принимая во внимание то, что статически скомпонованная программа имеет большой размер. Как мы можем определить является ли программа скомпонованой статически? И если она скомпонована динамически, как нам узнать какие библиотеки ей нужны? Ответ на оба вопроса это команда `ldd`, отображающая информацию о требуемых библиотеках для исполняемой программы. В Листинге 20 показан вывод команды `ldd` для исполняемых `ln` и `sln`.

### Листинг 20. Вывод программы ldd для sln и ln.

```
[ian@lyrebird ian]$ ldd /sbin/sln /bin/ln
/sbin/sln:
          not a dynamic executable
/bin/ln:
        libc.so.6 => /lib/tls/libc.so.6 (0x00ebd000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00194000)
```

Поскольку `ldd` в действительности предназначен для динамической компоновки она сообщает нам, что `sln` является статически скомпонованной говоря, что это "not a dynamic executable" (не динамический исполняемый файл), в то время как для `ln` приводятся имена двух совместно используемых библиотек (`libc.so.6` и `ld-linux.so.2`), которые ей необходимы, а также где их можно найти. Отметим, что `.so` указывает на то, что обе они являются *совместно используемыми объектами (shared objects)* или динамическими библиотеками. Для Листинга 21 мы воспользовались командой `ls -l`, чтобы показать, что это действительно символическая ссылка на конкретные версии библиотек.

### Листинг 21. Символические ссылки на библиотеки.

```
[ian@lyrebird ian]$ ls -l /lib/tls/libc.so.6; ls -l /lib/ld-linux.so.2
lrwxrwxrwx    1 root      root        13 May 18 16:24 /lib/tls/libc.so.6 -> libc-2.3.2.so
```

```
lrwxrwxrwx    1 root   root      11 May 18 16:24 /lib/ld-linux.so.2 -> ld-2.3.2.so
```

## Динамическая загрузка

Исходя из предыдущего, вы можете удивиться, узнав, что ld-linux.so, которая выглядит как библиотека совместного использования, на самом деле по своей природе является исполняемым файлом. Это код, отвечающий за динамическую загрузку. Он читает информацию заголовка исполняемого файла, приведенный в формате *Executable and Linking Format (формат ссылок и исполняемых)* или (*ELF*). Из этой информации можно определить какие библиотеки необходимы и какие следует загрузить. Затем он осуществляет динамическое связывание для согласования указателей на адреса в вашем исполняемом файле и загруженных библиотеках, для того, чтобы программа запустилась.

Вы не сможете найти man страницы для ld-linux.so, но вы можете обнаружить их для ld.so, выполнив `man ld.so`. Листинг 22 иллюстрирует использование опции `--list` для ld-linux.so, чтобы показать ту же информацию для команды ln, что мы выводили для команды ldd в Листинге 20.

### Листинг 22. Использование ld-linux.so для отображения требований для библиотеки.

```
[ian@lyrebird ian]$ /lib/ld-linux.so.2 --list /bin/ln
 libc.so.6 => /lib/tls/libc.so.6 (0x00a83000)
 /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00f2c000)
```

Отметим что шестнадцатиричные адреса для двух листингов отличаются. Скорее всего они будут различны даже если вы запустите ldd дважды.

## Настройка динамических библиотек

Итак, как же динамический загрузчик узнает где искать исполняемые файлы? Как и для многого в Linux в /etc имеется конфигурационный файл (configuration file). Фактически два конфигурационных файла: /etc/ld.so.conf и /etc/ld.so.cache. Листинг 23 показывает содержимое /etc/ld.so.conf для двух различных систем. Отметим, что в системе attic4 (под управлением Fedora Core 4), /etc/ld.so.conf указывает, что в подкаталоге ld.so.conf.d должны учитываться все файлы .conf. Реальное содержимое /etc/ld.so.conf для вашей системы может отличаться.

### Листинг 23. Содержимое /etc/ld.so.conf.

```
[ian@lyrebird ian]$ cat /etc/ld.so.conf
/usr/kerberos/lib
/usr/X11R6/lib
/usr/lib/qt-3.1/lib
[
[ian@attic4 ~]$ cat /etc/ld.so.conf
include ld.so.conf.d/*.conf
```

Необходимо, чтобы загрузка программ проходила быстро, поэтому файл ld.so.conf обрабатывается командой `ldconfig` для обработки всех библиотек из ld.so.conf.d, а также из надежных каталогов, /lib и /usr/lib. Динамический загрузчик использует файл ld.conf.cache для определения файлов, которые необходимо динамически загрузить и связать. Если вы

измените ld.so.conf (или добавьте новые включаемые (included) файлы в ld.so.conf.d, то вы должны выполнить команду **ldconfig** (от имени суперпользователя) чтобы перестроить файл ld.conf.cache.

Обычно вы используете команду ldconfig без параметров для перестройки ld.so.cache. Имеется также несколько параметров, которые вы можете указать для переопределения этого поведения по умолчанию. Как всегда выполните **man ldconfig** для получения большей информации. Мы проиллюстрировали использование параметра -p для отображения содержимого ld.so.cache в Листинге 24.

#### Листинг 24. Использование ldconfig для отображения ld.so.cache.

```
[ian@lyrebird ian]$ /sbin/ldconfig -p | more
768 libs found in cache `/etc/ld.so.cache'
    libzvt.so.2 (libc6) => /usr/lib/libzvt.so.2
    libz.so.1 (libc6) => /usr/lib/libz.so.1
    libz.so (libc6) => /usr/lib/libz.so
    libx11globalcomm.so.1 (libc6) => /usr/lib/libx11globalcomm.so.1
    libxsltbreakpoint.so.1 (libc6) => /usr/lib/libxsltbreakpoint.so.1
    libxslt.so.1 (libc6) => /usr/lib/libxslt.so.1
    libxmms.so.1 (libc6) => /usr/lib/libxmms.so.1
    libxml2.so.2 (libc6) => /usr/lib/libxml2.so.2
    libxml2.so (libc6) => /usr/lib/libxml2.so
    libxmltok.so.0 (libc6) => /usr/lib/libxmltok.so.0
    libxmlparse.so.0 (libc6) => /usr/lib/libxmlparse.so.0
    libxml.so.1 (libc6) => /usr/lib/libxml.so.1
    libxerces-c.so.24 (libc6) => /usr/lib/libxerces-c.so.24
    ...
lib-gnu-activation-20030319.so (libc6) => /usr/lib/lib-gnu-activation-20030319.so
ld-linux.so.2 (ELF) => /lib/ld-linux.so.2
```

Если вы запускаете старое приложение, которому требуется некая старая версия библиотеки или если вы разрабатываете новую библиотеку или новую версию библиотеки, то вам может понадобиться переопределить путь для поиска по умолчанию, используемый загрузчиком. Это может также понадобиться скриптам, использующим специфичные для продукта библиотеки, устанавливаемые в древо /opt.

Также как вы устанавливаете переменную PATH, вы можете указать путь поиска для исполняемых файлов, вы можете задать переменную LD\_LIBRARY\_PATH со списком каталогов, разделенных двоеточием, в которых следует искать библиотеки перед тем, как система станет искать их в ld.so.cache. Например, вы можете использовать команду вроде

```
export
LD_LIBRARY_PATH=/usr/lib/oldstuff:/opt/IBM/AgentController
/lib
```

В оставшихся разделах этого учебника мы рассмотрим управление пакетами.

| [предыдущая](#) | [следующая](#)

### Управление пакетами от Debian

Этот раздел посвящен изложению материала темы 1.102.5 для экзамена LPI 101 Администрирование для начинающих (LPIC-1). Рейтинг темы 8.

В предыдущих разделах мы научились устанавливать программы из исходных текстов. В

в этом разделе мы изучим альтернативу, используемую сегодня большинством дистрибутивов, управление пакетами (*package management*), в котором предварительно собранные программы или наборы программ распространяются в качестве *пакетов* (*package*), готовых для установки в конкретном дистрибутиве. В этом и следующем разделах мы рассмотрим управление пакетами, сфокусировавшись на двух наиболее распространенных системах управления пакетами. Это *Advanced Packaging Tool* или *APT*, разработанная *Debian* и *Red Hat Package Manager* или *RPM*, разработанная *Red Hat*.

## Обзор управления пакетами

В примере с Dr Geo предыдущего раздела, наши конфигурационные шаги сначала привели к ошибке, потому что у нас не была установлена требуемая программа. Инструменты управления пакетами формализуют указание требований и версий, стандартизируют их размещение в системе, а также обеспечивают механизм отслеживания, помогающий определить установленные пакеты. В результате получаем облегчение установки программного обеспечения, его поддержки и удаления.

Хотя вы все еще можете иметь желание устанавливать программы из исходных текстов по перечисленными в предыдущем разделе причинам, но вы, вероятно, большую часть поддержки системы и установки программ выполняете с использованием менеджера пакетов, который имеется в вашем дистрибутиве.

С точки зрения пользователя, основные функции управления пакетами обеспечиваются на уровне команд. Поскольку Linux-разработчики стараются сделать использование Linux легче, то основные инструменты снабжаются другими надстройками, включая графический интерфейс, скрывающий сложность базовых средств от конечного пользователя. В этих двух разделах мы сосредоточимся на базовых средствах, хотя и упомянем некоторые другие инструменты, чтобы вы имели стартовую платформу для их изучения.

## Установка пакетов Debian

Давайте вернемся к проблемам, с которыми мы столкнулись при работе с исходными текстами Dr Geo. Так получилось, что эти проблемы возникли в системе Fedora Core 4, использующей управление пакетами RPM. К счастью в этом разделе учебника, я также не досчитался нескольких компонентов guile в системе Ubuntu, основанной на Debian, в которой я пытался установить Dr Geo. Возникшие ошибки для этого случая приведены в Листинге 25.

### Листинг 25. Отсутствие функции guile.

```
ian@attic4:~$ cd drgeo-1.1.0
ian@attic4:~/drgeo-1.1.0$ ./configure
checking for perl... /usr/bin/perl
checking for XML::Parser... ok
checking for iconv... /usr/bin/iconv
checking for msgfmt... /usr/bin/msgfmt
...
checking for guile... no
configure: error: guile required but not found
i
```

Пакет, который необходим нам -- это пакет *guile*. Мы можем установить его используя команду **apt-get**, как показано в Листинге 26. Отметим, что использование команды **sudo** является обычным для Ubuntu способом работать с правами суперпользователя (root).

### Листинг 26. Установка guile с использованием apt-get.

```

ian@attic4:~$ sudo apt-get install guile
Reading package lists... Done
Building dependency tree... Done
Note, selecting guile-1.6 instead of guile
Suggested packages:
  guile-1.6-doc
The following NEW packages will be installed:
  guile-1.6
0 upgraded, 1 newly installed, 0 to remove and 24 not upgraded.
Need to get 31.5kB of archives.
After unpacking 209kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com hoary/main guile-1.6 1.6.7-1lubuntu1 [31.5kB]
Fetched 31.5kB in 0s (37.4kB/s)

Preconfiguring packages ...
Selecting previously deselected package guile-1.6.
(Reading database ... 84435 files and directories currently installed.)
Unpacking guile-1.6 (from .../guile-1.6_1.6.7-1lubuntu1_i386.deb) ...
Setting up guile-1.6 (1.6.7-1lubuntu1) ...
i
```

Из приведенного листинга мы видим, что apt-get откуда-то читает список пакетов (подробнее об этом далее), строит древо зависимостей, определяет, что guile-doc рекомендовано устанавливать вместе с guile, и загружает пакет guile из сети Интернет. Затем пакет guile распаковывается, устанавливается и настраивается. Отметим, что расширение, используемое для пакетов Debian, это .deb. Полное имя нашего пакета guile выглядит так guile-1.6\_1.6.7-1lubuntu1\_i386.deb.

Если apt-get обнаруживает, что пакет, который вы пытаетесь установить зависит от других пакетов, то он автоматически загружает и устанавливает также и их. В нашем примере устанавливается только guile, потому что все зависимости уже удовлетворены. Следуя выведенному совету мы можем установить guile-doc (или guile-1.6.doc).

Предположим, что вместо установки guile-doc, мы хотим узнать зависит ли пакет guile-doc от других пакетов. Мы можем использовать опцию apt-get -s (для симулирования). Имеется и несколько других опций со сходными функциями, такие как **--just-print** и **--dry-run**. Обратитесь к man страницам за подробностями. Не удивительно, что документация для пакета, который мы только что установили не имеет каких-либо зависимостей, поэтому в Листинге 27 мы привели несколько больше полезных примеров симулировав установку пакета ssl-cert, которому требуется пакет openssl.

### Листинг 27. Симуляция или пробный прогон установки ssl-cert.

```

ian@attic4:~$ sudo apt-get -s install ssl-cert
Reading package lists... Done
Building dependency tree... Done
The following extra packages will be installed:
  openssl
Suggested packages:
  ca-certificates
The following NEW packages will be installed:
  openssl ssl-cert
0 upgraded, 2 newly installed, 0 to remove and 24 not upgraded.
Inst openssl (0.9.7e-3 Ubuntu:5.04/hoary)
Inst ssl-cert (1.0-11 Ubuntu:5.04/hoary)
Conf openssl (0.9.7e-3 Ubuntu:5.04/hoary)
```

```
Conf ssl-cert (1.0-11 Ubuntu:5.04/hoary)
```

Мы видим, что требуется еще два новых пакета и вследствии этого они были установлены и настроены.

### Список пакетных ресурсов: apt-setup

Мы говорили, что apt-get откуда-то читает список пакетов. Читает его он из /etc/apt/sources.list. Это список, который вы можете изменять самостоятельно, но вероятно вы предпочтете настроить его, используя команду [apt-setup](#). Команда [apt-setup](#) это интерактивный инструмент, который знает расположение главных АРТ репозиториев. Вы можете иметь доступ к источникам пакетов на CD-ROM, в вашей локальной файловой системе или в сети с использованием http или ftp.

Если ваш дистрибутив установил для вас файл /etc/apt/sources.list, то он может и не содержать ваш CD-ROM в качестве источника пакетов. Это может быть неудобно особенно на начальной стадии экспериментирования с новой системой, когда вы можете захотеть добавить много пакетов, большинство которых еще не обновились. В этом случае вы можете воспользоваться командой

```
apt-cdrom add
```

чтобы добавить ваш CD-ROM к списку источников пакетов.

Apt-get и другие средства о которых мы будем говорить используют локальную базу данных для определения установленных пакетов. Они могут сверять установленные версии с доступными. Для этого информация о доступных версиях загружается с указанного в списке /etc/apt/sources.list источника и сохраняется на компьютере локально. Если вы обновляете ваш файл /etc/apt/sources.list, то вам следует выполнить

```
apt-get update
```

Это приведет сохраненные данные о доступных пакетах в актуальное состояние. Вообще же вы должны всегда делать это перед тем как установить новый пакет.

### Удаление или обновление пакетов.

перед тем как покинуть apt-get, мы рассмотрим две другие полезные опции.

Если вы установили пакет и позднее хотите удалить его используйте опцию apt-get [remove](#). Листинг 28 показывает как удалить пакет guile, который мы установили ранее.

#### Листинг 28. Удаление пакета guile.

```
ian@attic4:~$ sudo apt-get remove guile
Reading package lists... Done
Building dependency tree... Done
Note, selecting guile-1.6 instead of guile
The following packages will be REMOVED:
  guile-1.6
0 upgraded, 0 newly installed, 1 to remove and 24 not upgraded.
Need to get 0B of archives.
After unpacking 209kB disk space will be freed.
Do you want to continue [Y/n]? Y
(Reading database ... 84455 files and directories currently installed.)
Removing guile-1.6 ...
```

Другой опцией, которую мы хотим упомянуть является опция **upgrade** Эта опция обновляет все установленные пакеты в вашей системе до новейших версий. Не путайте ее с опцией **update**, которая просто обновляет информацию о доступных пакетах.

За более подробной информацией о возможностях и опциях apt-get обратитесь к *man* странице.

### Файл apt.conf

Если вы просмотрите *man* страницу apt-get, то обнаружите множество опций. Если вы используете команду apt-get постоянно и поняли, что опции по умолчанию вас не устраивают, то вы можете создать новые умолчания в /etc/apt/apt.conf. Программа **apt-config** доступна в виде скриптов для опроса файла apt.conf. За более подробной информацией обратитесь к *man* страницам для apt.conf и apt-config.

### Информация пакета Debian

Теперь мы обратимся к нескольким инструментам для получения информации о пакете. Некоторые из этих инструментов также выполняют и другие функции, но мы сосредоточимся на информационных аспектах.

### Статус пакета с dpkg

Еще один инструмент являющийся частью системы АРТ это **dpkg**. Это инструмент среднего уровня для управления пакетами, который может устанавливать и удалять пакеты, а также отображать их информацию. Конфигурирование dpkg может выполняться при помощи /etc/dpkg/dpkg.cfg. Отдельные пользователи могут также найти файл .dpkg.cfg в своем собственном домашнем каталоге, что обеспечивает дополнительное конфигурирование. Если у вас нет ни одного из этих файлов, то проверьте, например, /usr/share/doc/dpkg/dpkg.cfg.

Инструмент dpkg использует многие файлы в ветке /var/lib/dpkg вашей файловой системы. В частности, файл /var/lib/dpkg/status содержит статус информации о пакетах вашей системы. Листинг 29 показывает использование **dpkg -s** для отображения статуса пакета guile после его установки. Напомним, что мы действительно установили guile-1.6. Из Листинга 29 мы видим, что нам нужно указывать полное имя, а не сокращенное.

### Листинг 29. Статус пакета guile.

```
ian@attic4:~$ dpkg -s guile
Package `guile' is not installed and no info is available.

Use dpkg --info (= dpkg-deb --info) to examine archive files,
and dpkg --contents (= dpkg-deb --contents) to list their contents.
ian@attic4:~$ dpkg -s guile-1.6
Package: guile-1.6
Status: install ok installed
Priority: optional
Section: interpreters
Installed-Size: 204
Maintainer: Rob Browning <rlb@defaultvalue.org>
Architecture: i386
Version: 1.6.7-1lubuntu1
Provides: guile
Depends: guile-1.6-libs, libc6 (>= 2.3.2.ds1-4), libguile-ltdl-1
Suggests: guile-1.6-doc
Conflicts: libguile-dev (<= 1:1.4-24)
Description: The GNU extension language and Scheme interpreter
Guile is a Scheme implementation designed for real world programming,
providing a rich Unix interface, a module system, an interpreter, and
many extension languages. Guile can be used as a standard #! style
```

```
interpreter, via #!/usr/bin/guile, or as an extension language for  
other applications via libguile.
```

## Пакеты и файлы в них

Часто мы хотим знать, что находится в пакете или к какому пакету принадлежит тот или иной файл. Обе эти задачи для dpkg. Листинг 30 иллюстрирует использование `dpkg -L` для вывода списка файлов (включая каталоги) установленных из пакета guile.

### Листинг 30. Что находится в пакете guile?

```
root@attic4:~# dpkg -L guile-1.6  
/.  
/usr  
/usr/bin  
/usr/bin/guile-1.6-snarf  
/usr/bin/guile-1.6-tools  
/usr/bin/guile-1.6  
/usr/bin/guile-1.6-config  
/usr/share  
/usr/share/guile  
/usr/share/guile/1.6  
/usr/share/guile/1.6/scripts  
/usr/share/guile/1.6/scripts/autofrisk  
/usr/share/guile/1.6/scripts/display-commentary  
/usr/share/guile/1.6/scripts/doc-snarf  
/usr/share/guile/1.6/scripts/frisk  
/usr/share/guile/1.6/scripts/generate-autoload  
/usr/share/guile/1.6/scripts/lint  
/usr/share/guile/1.6/scripts/PROGRAM  
/usr/share/guile/1.6/scripts/punify  
/usr/share/guile/1.6/scripts/read-scheme-source  
/usr/share/guile/1.6/scripts/snarf-check-and-output-texi  
/usr/share/guile/1.6/scripts/snarf-guile-m4-docs  
/usr/share/guile/1.6/scripts/use2dot  
/usr/share/doc  
/usr/share/doc/guile-1.6  
/usr/share/doc/guile-1.6/copyright  
/usr/share/doc/guile-1.6/changelog.Debian.gz  
/usr/lib  
/usr/lib/menu  
/usr/lib/menu/guile-1.6
```

Чтобы найти пакет, содержащий какой-то конкретный файл, используйте опцию `dpkg -S` как показано в Листинге 31. Имя пакета выводится слева.

### Листинг 31. Какой пакет содержит файл?

```
ian@attic4:~$ dpkg -S /usr/share/guile/1.6/scripts/lint  
guile-1.6: /usr/share/guile/1.6/scripts/lint
```

Вы можете отметить, что список в Листинге 30 не содержит `/usr/bin/guile`, тогда как команда `which guile` говорит, что это программа, которая запустится если набрать `guile`. Когда

такое случается вам может понадобиться выполнить некоторую поисковую работу, чтобы найти откуда пришел пакет. Например, на этапе установки могут выполняться такие задачи, как создание символьных ссылок, которые не указываются в качестве составляющих пакета. Последним добавлением в систему Linux является система *alternatives* которая управляется командой **update-alternatives**. В Листинге 32, мы показали как использовать команду **ls**, чтобы увидеть с чем связана команда **guile** при помощи символьной ссылки. Ссылка на каталог /etc/alternatives подсказывает, что мы используем систему *alternatives*, и потому мы используем команду **update-alternatives**, чтобы найти больше информации и в конце концов мы сможем использовать команду **dpkg -S** для подтверждения, что команда **guile** получена из пакета **guile-1.6**. Настройка системы *alternatives* может быть сделана постустановочным скриптом, являющимся частью пакета **guile-1.6**.

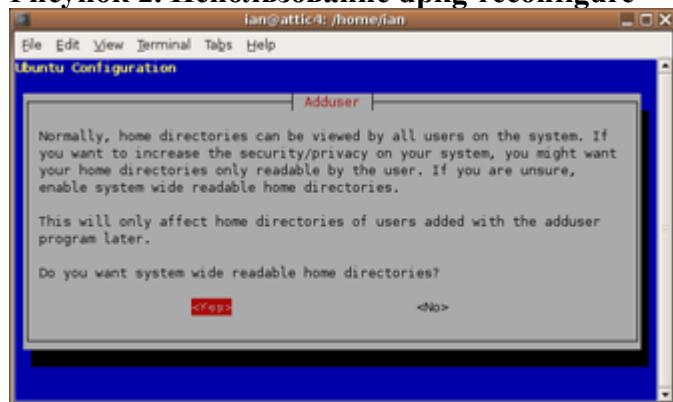
### Листинг 32. Более сложное использование **dpkg -S**

```
ian@attic4:~$ ls -l $(which guile)
lrwxrwxrwx 1 root root 23 2005-09-06 23:38 /usr/bin/guile -> /etc/alternatives/guile
ian@attic4:~$ update-alternatives --display guile
guile - status is auto.
  link currently points to /usr/bin/guile-1.6
/usr/bin/guile-1.6 - priority 160
  slave guile-config: /usr/bin/guile-1.6-config
  slave guile-snarf: /usr/bin/guile-1.6-snarf
  slave guile-tools: /usr/bin/guile-1.6-tools
Current `best' version is /usr/bin/guile-1.6.
ian@attic4:~$ dpkg -S /usr/bin/guile-1.6
guile-1.6: /usr/bin/guile-1.6
```

### Перенастройка пакетов Debian.

APT содержит функцию с именем *debconf*, которая используется для настройки пакетов после их установки. Пакеты, использующие эту функцию (а используют не все) могут быть перенастроены после того, как они уже установлены. Самый легкий способ сделать это -- использовать команду **dpkg-reconfigure**. Например, команда **adduser** может создать домашние каталоги, которые сможет просматривать любой пользователь системы. Вам это может не понравиться по соображениям безопасности. На Рисунке 2 приведены вопросы по настройке, соответствующие пакету **adduser**. Выполните **dpkg-reconfigure adduser** (от имени **root**) для вывода этого окна.

### Рисунок 2. Использование **dpkg-reconfigure**

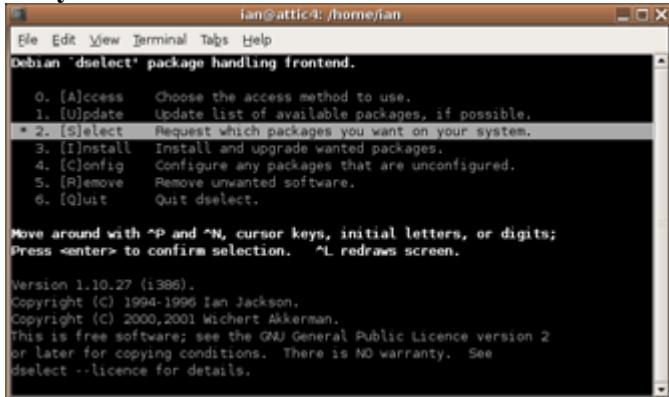


### Использование **dselect**

Ранее мы отмечали, что статус пакетов хранится в **/var/lib/dpkg/status**. Мы также указывали, что **dpkg** может делать больше, чем просто отображать информацию о пакете. Теперь мы

кратко рассмотрим команду **dselect**, предоставляющую текстовый полноэкранный интерфейс (используя ncurses) к функциям dpkg для управления пакетами. Вы можете использовать dselect для установки или удаления пакетов, а также для управления флагами статуса, показывающими должен ли пакет поддерживаться в актуальном состоянии или оставить его в текущем состоянии. Если вы выполните команду **dselect** (от имени root), то вы увидите экран, похожий на приведенный на Рисунке 3.

**Рисунок 3. Работающая dselect**



### Использование режима Select в dselect

Вы можете просматривать и изменять статус каждого пакета выбрав опцию Select подсвеченную на Рисунке 3. Вы увидите окно справки. Для выхода из справки в любой момент нажмите Пробел. Затем вы увидите список пакетов и групп пакетов.

Вы можете искать пакеты, используя / в конце строки поиска. На Рисунке 4 показан пример результата поиска для "guile".

**Рисунок 4. Окно Selection для dselect**



### Состояния выбора пакета

Состояние выбора для каждого пакета может быть определено по скрытому заголовку *EIO.M*. Эти буквы расшифровываются как *Error* (ошибка), *Installed state* (состояние Установлено), *Old mark* (помечен как старый) и *Mark* (помечен). Вы можете использовать клавишу "v" для переключения между краткой формой отображения этой информации и отображением этого в виде слов.

Четвертую колонку, или M, мы рассмотрим повнимательнее. Она описывает что случится после того, как мы закончим работу с окном выделения и перенесемся в окно установки. Пометки имеют следующее значение:

\*

- Установить или обновить до последней версии
- =
- Оставить пакет с текущим статусом и версией
- (дефис)
- Удалить пакет, но оставить его настройки для случая повторной переустановки позднее
- \_ (подчеркивание)
- Удалить пакет вместе с настройками.

Для изменения пометок нажмите соответствующую клавишу, за исключением того, что следует нажимать клавишу "+" чтобы пометить пакет для установки или обновления. По завершении нажмите **Enter** для подтверждения сделанных изменений или нажмите "X" (заглавную X) для отмены изменений без сохранения. Это вернет вас к окну, изображенном на Рисунке 3, с выбранной опцией **Install** (Установка). Нажмите **Enter** для установки или обновления вашей системы.

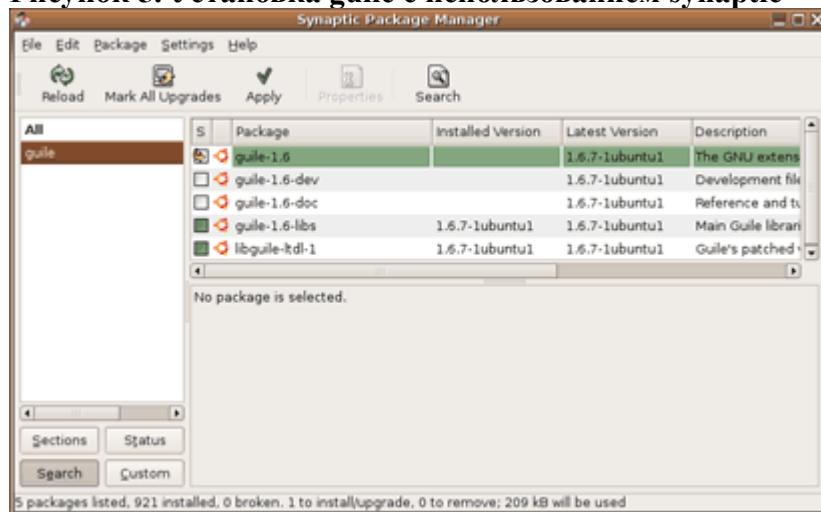
Если вам понадобится помочь, то в любой момент нажмите "?" (знак вопроса). Нажмите Пробел для выхода из справки.

### Обновление Debian другими средствами

Мы увидели, что **dselect** может помочь установить или удалить отдельные пакеты, а также обновить все пакеты вашей системы до последних версий. Если вы хотите сделать это из командной строки, то можете использовать **apt-get dselect-upgrade**, что обработает пометки, которые, как мы видели, настраиваются при помощи **dselect**.

Кроме **dselect** существует несколько других интерактивных интерфейсов управления пакетами для систем **Debian**, включая **aptitude**, **synaptic**, **gnome-apt** и **wajig**. **Synaptic** это графическое приложение для использования в оконной системе **X**. На рисунке 5 показан интерфейс пользователя **synaptic** с нашим старым другом -- пакетом **guile**, помеченный для установки.

**Рисунок 5. Установка **guile** с использованием **synaptic****



Кнопка **Apply** (Применить) установит **guile** и обновит все остальные пакеты, включенные в список на обновление. Кнопка **Reload** (Перезагрузить) обновит список пакетов. Если вы используете графический интерфейс, то возможно **synaptic** будет для вас более удобна в использовании нежели **apt-get**, **dpkg** или **dselect**.

### Поиск пакетов Debian

В нашем последнем пункте об управлении пакетами в **Debian**, мы рассмотрим способы поиска пакетов. Обычно **apt-get** и другие рассмотренные нами инструменты, уже знают все

пакеты, которые могут понадобиться вам из списка доступных пакетов. Команда, которую мы еще не использовали это **apt-cache**, она полезна для поиска информации о пакете в вашей системе. Apt-cache может искать регулярные выражения (о которых мы поговорим подробно в учебнике по теме 103). Предположим вам нужно найти имя пакета, содержащего загрузчик Linux. Листинг 33 показывает как можно сделать это.

### Листинг 33. Поиск загрузчика Linux при помощи apt-cache

```
ian@attic4:~$ apt-cache search "linux loader"
lilo - LInux LOader - The Classic OS loader can load Linux and others
lilo-doc - Documentation for LILO (LInux LOader)
```

Ранее мы видели, что dselect и synaptic также предоставляют средства поиска. Если вы используете synaptic, то заметьте, что в меню поиска имеются опции, при помощи которых вы можете указать область поиска: только имена или также описания пакетов.

Если вы все еще не можете найти пакет, то поищите в списке пакетов на сайте Debian или еще где-то в Интернет.

Если вы выполнили поиск и загрузили .deb файл, то вы можете установить его, используя **dpkg -i**. Например, Dr Geo может быть найден в виде .deb пакета в официальном репозитории пакетов Debian.

### Листинг 34. Установка Dr Geo из .deb пакета

```
ian@attic4:~$ ls drg*.deb
drgeo_1-1.0.0-1_i386.deb
ian@attic4:~$ sudo dpkg -i drgeo_1-1.0.0-1_i386.deb
Password:
Selecting previously deselected package drgeo.
(Reading database ... 84435 files and directories currently installed.)
Unpacking drgeo (from drgeo_1-1.0.0-1_i386.deb) ...
Setting up drgeo (1.0.0-1) ...
```

Заметьте, что архив с исходными текстами имеет большую версию (1.1.0), нежели deb пакет (1.0.0-1). Если вы установили Dr Geo и по некоторым причинам он не работает, то возможно вам придется установить его из исходных текстов

Если ничего не помогло, то существует еще один возможный источник пакетов.

Предположим, что вы нашли программу упакованную в RPM, а не .deb. Вы можете использовать программу **alien**, которая может осуществлять преобразование из одного формата пакетов в другой. Вам следует тщательно прочитать документацию alien поскольку не все возможности систем управления пакетами alien может преобразовать в другой формат.

Существует намного больше систем управления пакетами в Debian, нежели описано здесь. А также в Debian имеется многое кроме системы управления пакетами. Дополнительные ссылки смотри в [Ресурсах](#).

| [предыдущая](#) | [следующая](#)

## Менеджер пакетов Red Hat (RPM)

В этом разделе приводится материал по теме 1.102.6 для экзамена LPI 101

Администрирование для начинающих (LPIC-1). Рейтинг темы 8.

В предыдущем разделе об управлении пакетами в Debian, мы дали вам краткий [обзор управления пакетами](#). В этом разделе мы сосредоточимся на *Red Hat Package Manager* (менеджере пакетов Red Hat) или *RPM*, разработанном Red Hat. RPM и APT во многом схожи. Оба могут устанавливать и удалять пакеты. Оба хранят базу данных установленных пакетов. Оба имеют основные функции для командной строки, а также другие инструменты, предоставляющие более дружественный для пользователя интерфейс. Оба могут загружать пакеты из Интернет. Вообще говоря, RPM-пакетов не так много, как пакетов для APT, хотя команда *rpm* обладает большими возможностями. Другим отличием является то, что RPM не хранит информацию об установленных пакетах в вашей системе в той же расширенной форме, что и *dpkg*.

Red Hat представила RPM в 1995. В настоящее время RPM это система управления пакетами используемая для создания пакетов в Linux Standard Base (LSB). Опции команды *rpm* сгруппированы в три подгруппы для:

- Опроса и проверки пакетов
- Установки, обновления и удаления пакетов
- выполнения других функций.

В этом учебнике мы сосредоточимся на первых двух. Вы можете найти информацию о других функциях в *man* страницах для *rpm*.

Мы также должны отметить, что *rpm* это имя основной команды, используемой с RPM, а *.rpm* это расширение, используемое для файлов RPM. Поэтому " rpm-пакет" или "xxx rpm-пакет" будет означать, вообще говоря, файл RPM, тогда как " rpm" обычно будет означать команду.

### Установка и удаление пакетов RPM

Как и в предыдущем разделе мы рассмотрим проблемы, обнаружившиеся при установке Dr Geo в системе Fedora Core 4 в разделе [Компиляция и установка программ](#). Вы можете вернуться к Листингу 17, в котором мы пропустили команду *guile-config*.

### Введение в rpm

Команда *rpm* может устанавливать пакеты из локальной файловой системы или из Интернет, используя http или ftp. В Листинге 35 показана установка пакета *guile-devel* с использованием команды *rpm -ivh* и сетевого источника пакета.

### Листинг 35. Установка *guile-devel* при помощи *rpm*

```
[root@attic4 ~]# rpm -ivh http://download.fedoraproject.org/pub/fedora/linux/core/4/i386/os/Fedora/RPMS/guile-devel-1.6.7-2.i386.rpm
Retrieving http://download.fedoraproject.org/pub/fedora/linux/core/4/i386/os/Fedora/RPMS/guile-devel-1.6.7-2.i386.rpm
Preparing... ################################ [100%]
1:guile-devel ################################ [100%]
```

Отметим, что опция *-v* предоставляет подробный вывод, а опция *-h* показывает знак "решетка" (#), для индикации прогресса. Если вы хотите перед установкой из сети проверить пакет, то возможно вам придется сначала загрузить его и только потом установить. Мы поговорим о проверке пакетов чуть позже, а сейчас давайте используем команду [wget](#) для получения пакета с его последующей установкой из локальной файловой системы **без** использования опций *-vh*. Вывод показан в Листинге 36.

### Листинг 36. Установка guile-devel из файла

```
[root@attic4 ~]# wget http://download.fedoraproject.org/pub/fedora/linux/core/4/i386/os/Fedora/RPMS/guile-devel-1.6.7-2.i386.rpm
--22:29:58-- http://download.fedoraproject.org/pub/fedora/linux/core/4/i386/os/Fedora/RPMS/guile-devel-1.6.7-2.i386.rpm
              => `guile-devel-1.6.7-2.i386.rpm'
Resolving download.fedoraproject.org... 209.132.176.221
Connecting to download.fedoraproject.org[209.132.176.221]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 481,631 [application/x-rpm]

100%[=====] 481,631          147.12K/s    ETA 00:00
22:30:02 (140.22 KB/s) - `guile-devel-1.6.7-2.i386.rpm' saved [481,631/481,631]

[root@attic4 ~]# ls guile*
guile-devel-1.6.7-2.i386.rpm
[root@attic4 ~]# rpm -i guile-devel-1.6.7-2.i386.rpm
```

**Ни решеток, ни сообщений.**

### Переустановка rpm-пакета

Если вы самостоятельно выполнили указанные выше команды, то на втором шаге вы вероятно увидели ошибку (или на первом, если в вашей системе пакет guile-devel уже установлен), сообщающую, что guile-devel уже установлен. Для того, чтобы обойти это, вам следует использовать опцию **-e** для удаления (или стирания) rpm-пакета перед его повторной установкой, как показано в Листинге 37. Это также следует сделать, если вам необходимо переустановить rpm-пакет, из-за того, что вы случайно удалили его некоторые файлы.

### Листинг 37. Удаление guile-devel

```
[root@attic4 ~]# rpm -e guile-devel
```

### Принудительная установка rpm-пакета

Иногда удаление rpm-пакета не удобно, особенно если имеются некоторые программы, зависящие от него. Например, если вы попытаетесь удалить пакет guile вместо guile-devel, то можете увидеть нечто подобное Листингу 38, где многие установленные пакеты зависят от пакета guile, а потому удаление не разрешается.

### Листинг 38. Попытка удалить guile.

```
[root@attic4 ~]# rpm -q -R guile-devel
/bin/sh
/usr/bin/guile
guile = 5:1.6.7
rpmlib(CompressedFileNames) <= 3.0.4-1
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
[root@attic4 ~]# rpm -e guile
error: Failed dependencies:
        libguile-ltdl.so.1 is needed by (installed) g-wrap-1.3.4-8.i386
```

```
libguile-ltdl.so.1 is needed by (installed) gncash-1.8.11-3.i386
libguile.so.12 is needed by (installed) g-wrap-1.3.4-8.i386
libguile.so.12 is needed by (installed) gncash-1.8.11-3.i386
libqthreads.so.12 is needed by (installed) g-wrap-1.3.4-8.i386
libqthreads.so.12 is needed by (installed) gncash-1.8.11-3.i386
guile is needed by (installed) g-wrap-1.3.4-8.i386
guile = 5:1.6.7 is needed by (installed) guile-devel-1.6.7-2.i386
/usr/bin/guile is needed by (installed) guile-devel-1.6.7-2.i386
```

Не стоит говорить, что в этом случае не нужно удалять все пакеты, зависящие от этого, решением данной проблемы является принудительная установка грт-пакета при помощи опции `--force`. В Листинг 39 мы проиллюстрировали принудительную переустановку `guile-devel` из файла загруженного в Листинге 36.

### Листинг 39. Установка guile-devel с опцией --force.

```
[root@attic4 ~]# rpm -ivh --force guile-devel-1.6.7-2.i386.rpm
Preparing... ################################ [100%]
1:guile-devel ################################ [100%]
```

## Принудительное удаление rpm-пакета

Существует альтернатива принудительной установке при помощи опции `--force`, что может понадобиться в некоторых случаях. Вы можете удалить грт-пакет, используя опцию `--nodeps`, отключающую внутреннюю проверку зависимостей. Вообще вам следует делать это **только** если вы знаете что делаете и **только** если вы собираетесь исправить проблемы с зависимостями путем переустановки пакета. Примером может служить необходимость по некоторым причинам сменить версию пакета на более раннюю и желание убедиться, что все следы поздней версии были удалены. Команда, которую вам следует использовать для удаления пакета `guile` без проверки зависимостей выглядит так

```
rpm -e --nodeps guile
```

Опцию `--nodeps` можно также использовать и при установке npm-пакета. И опять это не рекомендуется, но иногда необходимо:

## Обновление RPM пакетов

Теперь, когда вы знаете как устанавливать и удалять RPM-пакеты, обратим свой взор на обновление до новых версий. Это похоже на установку, за тем исключением, что мы используем опцию `-U` или `-F` вместо опции `-i`. Различие между двумя этими опциями состоит в том, что опция `-U` обновляет существующие пакеты **или** устанавливает пакет, если он не был установлен, тогда как опция `-F` только обновляет или *freshen* (*освежает*) уже установленные пакеты. Вследствии этого опция `-U` используется чаще, особенно когда командная строка содержит список RPM-пакетов. Таким образом, удаленные пакеты устанавливаются, а установленные -- обновляются. Листинг 40 показывает результат попытки обновления `guile-devel` до текущей версии, а затем он удаляется и попытка обновления повторяется (теперь это срабатывает как установка).

#### Листинг 40. Обновление guile-devel.

```
package guile-devel-1.6.7-2 is already installed
[root@attic4 ~]# rpm -e guile-devel
[root@attic4 ~]# rpm -Uvh guile-devel-1.6.7-2.i386.rpm
Preparing...                                          #### [100%]
 1:guile-devel                                     #### [100%]
```

## Опрос RPM-пакетов

Как вы могли понять из приведенных нами примеров установка rpm-пакетов требует указания полного имени файла (или URL), вроде guile-devel-1.6.7-2.i386.rpm. С другой стороны, удаление rpm-пакета требует только имени пакета, вроде guile-devel. Также как и с APT, RPM хранит внутреннюю базу установленных пакетов, позволяющую вам манипулировать установленными пакетами, используя имя пакета. В этой части учебника мы рассмотрим какая же информация из этого хранилища доступна для вас, при помощи опции **-q** (от *query*) для команды rpm.

Базовый запрос просто спрашивает пакет установлен или нет. Добавьте опцию **-i** и вы получите информацию о пакете. Отметим, что для установки, обновления и удаления пакетов вам потребуются права суперпользователя (root), но не-root пользователи могут выполнять запросы к базе данных rpm-пакетов.

### Листинг 41. Отображение информации о guile-devel.

```
[ian@attic4 ~]$ rpm -q guile-devel
guile-devel-1.6.7-2
[ian@attic4 ~]$ rpm -qi guile-devel
Name        : guile-devel                         Relocations: (not relocatable)
Version     : 1.6.7                               Vendor: Red Hat, Inc.
Release     : 2                                  Build Date: Wed 02 Mar 2005 11:04:14 AM EST
Install Date: Thu 08 Sep 2005 08:35:45 AM EDT    Build Host: porky.build.redhat.com
Group       : Development/Libraries               Source RPM: guile-1.6.7-2.src.rpm
Size        : 1635366                            License: GPL
Signature   : DSA/SHA1, Fri 20 May 2005 01:25:07 PM EDT, Key ID b44269d04f2a6fd2
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary     : Libraries and header files for the GUILE extensibility library.
Description :
The guile-devel package includes the libraries, header files, etc.,
that you will need to develop applications that are linked with the
GUILE extensibility library.
```

You need to install the guile-devel package if you want to develop applications that will be linked to GUILE. You also need to install the guile package.

## RPM пакеты и файлы в них

У вас будет часто появляться желание узнать что же находится внутри пакета или к какому пакету принадлежит некоторый файл. Для вывода списка файлов пакета guile-devel, используйте опцию **-ql**, как показано в Листинге 42. В этом пакете много файлов, но мы покажем только часть этого вывода.

### Листинг 42. Отображение информации о guile-devel.

```
[ian@attic4 ~]$ rpm -ql guile-devel
/usr/bin/guile-config
```

```
/usr/bin/guile-snarf
/usr/include/guile
/usr/include/guile/gh.h
/usr/include/guile/srfi
/usr/include/guile/srfi/srfi-13.h
/usr/include/guile/srfi/srfi-14.h
/usr/include/guile/srfi/srfi-4.h
/usr/include/libguile
/usr/include/libguile.h
...
```

Вы можете ограничить список выводимых файлов, до списка настроечных файлов, добавлением опции **-C** к вашему запросу. Подобно этому, опция **-d** ограничивает вывод только файлами документации.

### Опрос файла пакета

Приведенная выше команда это запрос к базе данных RPM установленных пакетов. Если вы только что загрузили пакет и хотите получить ту же информацию, то вы можете воспользоваться опцией **-p** (для *файла пакета*) в вашем запросе с указанием имени **файла** пакета (также как при установке пакета). Листинг 43 повторяет запросы из Листинга 41 к файлу пакета вместо базы данных RPM.

### Листинг 43. Отображение информации для файла пакета guile-devel.

```
[ian@attic4 ~]$ rpm -qp guile-devel-1.6.7-2.i386.rpm
guile-devel-1.6.7-2
[ian@attic4 ~]$ rpm -qpi guile-devel-1.6.7-2.i386.rpm
Name        : guile-devel                               Relocations: (not relocatable)
Version     : 1.6.7                                     Vendor: Red Hat, Inc.
Release     : 2                                         Build Date: Wed 02 Mar 2005 11:04:14 AM EST
Install Date: (not installed)                         Build Host: porky.build.redhat.com
Group       : Development/Libraries                   Source RPM: guile-1.6.7-2.src.rpm
Size        : 1635366                                    License: GPL
Signature   : DSA/SHA1, Fri 20 May 2005 01:25:07 PM EDT, Key ID b44269d04f2a6fd2
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary     : Libraries and header files for the GUILE extensibility library.
Description :
The guile-devel package includes the libraries, header files, etc.,
that you will need to develop applications that are linked with the
GUILE extensibility library.
```

You need to install the guile-devel package if you want to develop applications that will be linked to GUILE. You also need to install the guile package.

### Опрос всех установленных пакетов

Опция **-a** применяет ваш запрос ко всем установленным пакетам. Это приводит к очень большому выводу информации, поэтому обычно его используют вместе с одним или несколькими фильтрами, таким как **sort** для сортировки листинга, **more** или **less** для разбивки на страницы, **wc** для получения количества пакетов или файлов, или **grep** для поиска пакетов имя которых вы знаете не точно. Листинг 44 показывает следующие запросы:

1. Отсортированный список всех пакетов в системе.
2. Количество всех пакетов в системе.

3. Количество всех файлов во всех пакетах системы.
4. Количество всех файлов документации установленных при помощи RPM.
5. Поиск всех пакетов с именем, содержащим "guile" (с учетом регистра).

#### Листинг 44. Запросы ко всем пакетам.

```
[ian@attic4 ~]$ rpm -qa | sort | more
4Suite-1.0-8.b1
a2ps-4.13b-46
acl-2.2.23-8
acpid-1.0.4-1
alchemist-1.0.36-1
alsa-lib-1.0.9rf-2.FC4
alsa-utils-1.0.9rf-2.FC4
...
[ian@attic4 ~]$ rpm -qa | wc -l
874
[ian@attic4 ~]$ rpm -qal | wc -l
195681
[ian@attic4 ~]$ rpm -qald | wc -l
31881
[ian@attic4 ~]$ rpm -qa | grep -i "guile"
guile-devel-1.6.7-2
guile-1.6.7-2
```

Используя `rpm -qa` можно облегчить администрирование нескольких систем. Если вы перенаправите отсортированный листинг в файл на одной из машин, то выполнив то же самое на другой машине, вы, при помощи программы `diff`, сможете найти различия.

#### Поиск владельца файла.

Учитывая, что теперь вы можете вывести список всех пакетов и всех файлов пакета, можно сказать, что теперь у вас есть все, чтобы обнаружить к какому пакету принадлежит некий файл. Однако команда `rpm` предоставляет опцию `-f`, для помощи в обнаружении пакета, являющегося владельцем файла. В нашем примере с Dr Geo в разделе [Компиляция и установка программ](#), нам потребовалась `guile-config`. Теперь у нас есть установленный пакет `guile-devel`, этот исполняемый файл то что нам нужно. В Листинге 45 показано как использовать команду `which` для получения полного пути к команде `guile-config`, и полезный пример использования вывода в качестве входных параметров для команды `rpm -qf`. Отметим, что апострофы, окружающие `which guile-config` наклонены в обратную сторону. Другим способом использования в bash является использование `$(which guile-config)`

#### Листинг 45. К какому пакету принадлежит `guile-config`.

```
[ian@attic4 ~]$ which guile-config
/usr/bin/guile-config
[ian@attic4 ~]$ rpm -qf `which guile-config`
guile-devel-1.6.7-2
[ian@attic4 ~]$ rpm -qf $(which guile-config)
guile-devel-1.6.7-2
```

#### RPM зависимости

Ранее мы видели, что не можем стереть пакет `guile` из-за *зависимостей*. В добавок к файлам

RPM пакет может содержать различные *функции*, от которых могут зависеть другие пакеты. В нашем примере многим другим пакетам требуются функции, предоставляемые пакетом guile. И мы не сможем установить guile-devel, если мы еще не установили в системе guile. А как только guile-devel будет установлен, он станет еще одной причиной почему guile не может быть удален.

Обычно все это работает хорошо. Если вам необходимо установить несколько пакетов одновременно, некоторые из которых могут зависеть от других, просто предоставьте полный список вашей команде `rpm -Uvh` и она проанализирует зависимости и выполнит установку в верном порядке.

Вместо попыток стереть установленный пакет и получения сообщения об ошибке, команда rpm предоставляет опцию для опроса установленных пакетов или файлов пакетов для поиска от чего они зависят или что *требуют*. Это опция `--requires`, которая может быть сокращена до `-R`. В Листинге 46 перечислены функции, необходимые для guile-config. Добавьте опцию `-p` и используйте полные RPM-имена, если хотите опросить файл пакета вместо базы данных RPM.

#### Листинг 46. Требования guile-config.

```
[ian@attic4 ~]$ rpm -qR guile-devel  
/bin/sh  
/usr/bin/guile  
guile = 5:1.6.7  
rpmlib(CompressedFileNames) <= 3.0.4-1  
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
```

В добавок к поиску функциональности, необходимой пакету, вам может потребоваться найти, какому файлу требуется данная функциональность (что делается при попытке удаления пакета). Листинг 47 иллюстрирует это для двух функций, требующихся guile-devel.

Поскольку вывод может включать повторы, мы также показали как можно отфильтровать вывод при помощи `sort` и `uniq` для вывода каждого требуемого пакета только один раз.

#### Листинг 47. Что необходимо /usr/bin/guile и guile.

```
[ian@attic4 ~]$ rpm -q --whatrequires /usr/bin/guile guile  
guile-devel-1.6.7-2  
g-wrap-1.3.4-8  
guile-devel-1.6.7-2  
[ian@attic4 ~]$ rpm -q --whatrequires /usr/bin/guile guile | sort|uniq  
guile-devel-1.6.7-2  
g-wrap-1.3.4-8
```

#### Целостность RPM пакетов.

Чтобы убедиться в целостности RPM-пакетов, они содержат MD5 или SHA1 дайджест, а также могут иметь цифровую подпись. Пакеты с цифровой подписью обычно требуют публичный ключ для проверки. Чтобы проверить целостность файла RPM-пакета используйте опцию rpm `--checksig` (сокращенно `-K`). Обычно вы найдете полезным добавление опции `-V` для подробного вывода.

#### Листинг 48. Проверка целостности файла пакета guile-devel.

```
[ian@attic4 ~]$ rpm --checksig guile-devel-1.6.7-2.i386.rpm  
guile-devel-1.6.7-2.i386.rpm: (sha1) dsa sha1 md5 gpg OK  
[ian@attic4 ~]$ rpm -Kv guile-devel-1.6.7-2.i386.rpm  
guile-devel-1.6.7-2.i386.rpm:  
  Header V3 DSA signature: OK, key ID 4f2a6fd2  
  Header SHA1 digest: OK (b2c61217cef4a72a8d2eddb8db3e140e4e7607a1)  
  MD5 digest: OK (cf47354f2513ba0c2d513329c52bf72a)  
  V3 DSA signature: OK, key ID 4f2a6fd2
```

Вы можете получить строку вывода, похожую на :

V3 DSA signature: NOKEY, key ID 16a61572

Это означает, что этот пакет подписан, но у вас в базе данных RPM нет необходимого публичного ключа. Отметим, что ранние версии RPM могли обеспечивать другую верификацию.

Если пакет подписан и вы хотите проверить соответствует ли он подписи, то вы должны найти файл соответствующей подписи и импортировать его в базу данных RPM. Сначала вам следует загрузить ключ, а затем проверить его отпечатки пальцев перед тем как импортировать его с использованием команды `rpm -i import`. Для более подробной информации смотри Man страницы для rpm. Вы также можете найти большое количество информации об подписанных бинарных файлах на [www.rpm.org](http://www.rpm.org).

### Проверка установленного пакета

Подобно проверке целостности rpm-пакета вы можете захотеть проверить целостность установленных файлов, используя `rpm -V`. Эти шаги необходимы, чтобы убедиться в том, что файлы не изменились с тех пор, как были установлены из rpm-пакета. Как показано в Листинге 49 если пакет все еще не был изменен, то ничего не выводится.

### Листинг 48. Проверка установленного пакета guile-devel.

```
[ian@attic4 ~]$ rpm -V guile-devel
```

Давайте войдем как root и удалим /usr/bin/guile-config вместе и заменим /usr/bin/guile-snarf копией /bin/bash и попытаемся сделать то же самое вновь. Результат приведен в Листинге 49.

### Листинг 49. Подделка пакета guile-devel.

```
[root@attic4 ~]# rm /usr/bin/guile-config  
rm: remove regular file `/usr/bin/guile-config'? y  
[root@attic4 ~]# cp /bin/bash /usr/bin/guile-snarf  
cp: overwrite `/usr/bin/guile-snarf'? y  
[root@attic4 ~]# rpm -V guile-devel  
missing      /usr/bin/guile-config  
S.5....T    /usr/bin/guile-snarf
```

Этот листинг показывает нам, что /usr/bin/guile-snarf имеет не верную MD5 сумму, размер файла и не прошел проверку mtime. Вы можете исправить это стерев пакет и переустановив его, или принудительно переустановив, как было показано ранее. Если вы удаляете пакет, то

ждите сообщения об ошибке, поскольку один из его файлов отсутствует.

## Настройка RPM

RPM редко требует настройки. В старых версиях rpm вы могли производить изменения в /etc/rpmrc для управления операциями во время работы. В последних версиях этот файл был перенесен в /usr/lib/rpm/rpmrc, где он автоматически заменяется, при обновлении rpm-пакета, что приводит к потере любых сделанных вами изменений. Если требуется некая специфичная для системы конфигурация, то она все еще может быть добавлена в /etc/rpmrc, а индивидуальная конфигурация для пользователей должна находиться в .rpmrc.in в домашнем каталоге пользователя. Вы можете найти описание формата этих файлов в книге *Maximizing RPM* (Смотри [Ресурсы](#)).

Если вы хотите просмотреть конфигурацию rpmrc, то для этого имеется специальная опция команды rpm. Используйте команду:

```
rpm --showrc
```

## Репозитории и другие средства

Теперь вы можете удивиться откуда же берутся все эти rpm пакеты, как их можно найти и как управлять обновлением своей системы. Если у вас дистрибутив, основанный на RPM (RPM-based), то скорее всего ваш дистрибутив имеет *репозиторий (repository)* пакетов. Ваш дистрибутив может также предоставлять инструменты для установки пакетов из репозитория или обновления вашей исходной системы. Эти инструменты могут быть как графическими, так и работать в командной строке. Примерами могут быть:

- YaST (SUSE)
- up2date (Red Hat)
- yum - Yellow Dog Updater Modified (Fedora и другие)
- Mandrake Software Management (Mandriva)

Обычно эти инструменты выполняют обновления многих пакетов в автоматическом или полу-автоматическом режиме. Они могут также обеспечивать возможности отображения содержимого репозиториев или поиска пакетов. Обратитесь к документации вашего дистрибутива за детальной информацией.

Если вы не можете найти какой-то особенный RPM при помощи встроенных средств вашей системы, другим прекрасным ресурсом для поиска RPM-пакетов является сервер Rpmfind.Net

Следующий учебник этой серии посвящен теме 103 -- GNU и команды Unix.

## Ресурсы

### Научиться

- Оригинал статьи "[Linux installation and package management](#)".
- В [Программе LPIC](#) вы можете найти список заданий, примерные вопросы и детальное описание требований каждого уровня сертификации по системному администрированию Профессионального института Linux.
- Проект [The Linux documentation project](#) -- это место, где находится большое количество полезной документации о Linux, включая [Linux Partition HOWTO](#) о планировании и создании разделов жестких дисков IDE и SCSI.
- Вы можете узнать больше о FHS на [домашней странице Filesystem Hierarchy Standard \(Стандарт иерархии файловых систем\)](#).

- Дополнительная информация о GRUB и GRUB 2 доступна на [домашней странице GNU GRUB](#). GNU GRUB это менеджер загрузки, название происходит от GRand Unified Bootloader (Большой унифицированный загрузчик).
- [Спецификации интерфейса множественной загрузки](#) позволяют любому совместимому загрузчику загружать соответствующую операционную систему.
- Обратитесь к "[Debian installation guide \(Руководство по установке Debian\)](#)" за дополнительной информацией об установке Debian.
- Просмотреть и найти список пакетов Debian можно в разделе [Debian packages](#).
- Узнайте об утилите управления пакетами APT от Debian в "[APT HOWTO](#)."
- Скачайте [Alien package converter](#).
- Просмотрите "[Guide to creating your own Debian packages \(Руководство по созданию собственного пакета Debian\)](#)."
- Посетите [Ubuntu](#), Linux с человеческим лицом.
- По адресу [www.rpm.org](#) можно найти последнюю информацию об инструментах создания RPM-пакетов и ссылки на информацию об RPM.
- Книга [Maximum RPM](#) предоставляет наиболее полное и систематическое изложение всех аспектов RPM. Она доступна в электронном и бумажном виде.
- Прочитайте [RPM HOWTO](#), чтобы узнать о Red Hat Package Manager (Менеджер пакетов Red Hat), RPM.
- Ищите RPM для вашего дистрибутива в [Rpmfind.Net](#) и [RPM Search](#).
- На странице [LSB Home](#) можно узнать о проектах Linux Standard Base (LSB), Free Standards Group (FSG), разрабатывающих стандартное бинарное окружение.
- Ищите материалы о сертификации в книжном варианте в [LPI Linux Certification in a Nutshell](#) (O'Reilly, 2001).
- Больше ресурсов для разработчиков Linux можно найти в [Linux-разделе developerWorks](#).

## **Получить продукты и технологии**

- Загрузите [System rescue CD-Rom \(Диск восстановления системы\)](#), один из многих инструментов доступный в сети, призванный помочь вам в восстановлении системы после сбоя.
- Просмотрите [SourceForge.net](#) большой репозиторий программ с открытым кодом для различных платформ.
- [Dr. Geo, interactive geometry \(Интерактивная геометрия Доктора Гео\)](#) это исходный проект, который используется в качестве примера в этом учебнике.
- Посетите [Info-ZIP](#) для получения программ Zip и Unzip для Linux и других платформ.
- Программа [Sphere Eversion Program \(выворачивание сферы\)](#) покажет вам как вывернуть сферу наизнанку без ее разрезания или сминания. Не очевидный результат лежащий в основе этой программы описывается при помощи видео файла с именем "Outside In".
- Получите [пакеты Debian](#) на домашней странице Debian.
- Создайте свой следующий проект для Linux с помощью [trial-приложений IBM](#),

доступных для загрузки прямо с developerWorks.

[В начало](#)

# Подготовка к экзамену LPI 101: GNU и UNIX команды

*Junior Level Administration (LPIC-1) тема 103*

Ян Шилдс, Старший программист, EMC

**Описание:** Графические пользовательские интерфейсы облегчают диалог с системой, но, чтобы понять настоящую мощь Linux, необходимо уметь работать с командной строкой. Приготовьтесь к экзамену LPI®101 и узнайте о потоках и фильтрах, файлах и управлению процессами, регулярных выражениях, а также редакторе vi. В третьем из пяти руководств, Ян Шилдс знакомит вас с командной строкой Linux®, а также некоторыми командами GNU и UNIX. К концу этого руководства вы сможете свободно использовать команды на любой Linux системе.

[Больше статей из этой серии](#)

**Дата:** 15.11.2005

**Уровень сложности:** средний

## Прежде чем начать

Посмотрите, чему могут научить эти руководства и как получить из них максимум пользы.

## О руководствах

Linux Professional Institute(LPI) сертифицирует системных администраторов Linux по двум уровням: *уровень начинающий* (также именуемый "certification level 1") и *уровень промежуточный* (также именуемый "certification level 2"). Чтобы достигнуть certification level 1, вы должны сдать экзамены 101 и 102; чтобы достигнуть certification level 2, вы должны сдать экзамены 201 и 202.

*Таблица 1. LPI экзамен 101: Руководства и темы*

тема LPI экзамена 101	руководство developerWorks	Аннотация руководства
Тема 101	<a href="#">Подготовка к LPI экзамену 101 (тема 101): Аппаратное обеспечение и архитектура</a>	Научитесь конфигурировать ваше оборудование в Linux. По окончании этого руководства вы будете знать, как Linux конфигурирует оборудование на современном PC, и что делать в случае возникновения проблем.
Тема 102	<a href="#">Подготовка к LPI экзамену 101: Установка Linux и управление пакетами</a>	Получите представление об установке Linux и управлению пакетами. К концу этого руководства вы будете знать, как Linux использует дисковые разделы, из чего состоит процесс загрузки Linux, и как устанавливать и управлять пакеты с программами.
Тема 103	Подготовка к LPI экзамену 101: GNU и UNIX команды	(Это руководство). Получите представление о наиболее общих командах GNU и UNIX. К концу этого руководства вы будете знать, как использовать команды в командной строке,

		включая команды обработки текста и фильтры, как искать файлы и каталоги, а также как управлять процессами.
Тема 104	Подготовка к LPI экзамену 104: Устройства, файловые системы Linux, Стандарт на структуру файловой системы	Скоро появится
Тема 110	Подготовка к LPI экзамену 110: Система X Window	Скоро появится

Чтобы сдать экзамены 101 и 102 (и получить certification level 1), вы должны уметь:

- Работать в командной строке Linux
- Выполнять простые задачи по поддержке системы: выручать пользователей, добавлять пользователей к системе, создавать резервные копии и пользоваться ими, а также выключать и перезагружать систему
- Установить и сконфигурировать рабочую станцию (включая X) и подсоединить ее к LAN или подсоединить отдельный компьютер через modem в Internet

Чтобы продолжить приготовление на сертификацию уровня 1, смотри [руководства developerWorks для экзамена LPI 101](#). Узнайте больше о [всех руководствах LPI на developerWorks](#).

Linux Professional Institute не одобряет никаких других учебных материалов и техник в частности. За более подробной информацией, пожалуйста, обращайтесь [info@lpi.org](mailto:info@lpi.org).

## Об этом руководстве

Добро пожаловать в "GNU и UNIX команды", третье из пяти руководств, созданное для подготовки вас к сдаче экзамена LPI 101. Из этого руководства вы узнаете, как легко использовать GNU и UNIX команды.

Это руководство организовано в соответствии с задачами LPI для этой темы. Предварительно, ожидайте больше вопросов на экзамене по задачам с более высоким весом.

*Таблица 2. GNU и UNIX команды: Экзаменационные задачи, рассматриваемые в этом руководстве*

Экзаменационная задача LPI	Значимость задачи	Краткое описание задачи
1.103.1 <a href="#">Работа с командной строкой</a>	Вес 5	Взаимодействие с оболочкой и командами с помощью командной строки. Эта задача включает в себя набор правильных команд и последовательностей команд, определения, ссылки и экспорта переменных окружения, использование истории команд и возможностей редактирования, запуска команд, находящихся в переменной окружения PATH так и вне ее, использование подстановки команды, применение команд рекурсивно к дереву каталогов, а также умение использовать man-страницы.

1.103.2	Вес 6	Применение фильтров к текстовым потокам. Цель этой задачи состоит в пересылке текстовых файлов и выходных потоков через утилиты обработки текста с целью модификации выходного потока, используя стандартные команды UNIX, входящие в GNU пакет textutils.
1.103.3	Вес 3	Использование основных команд UNIX для копирования, перемещения и удаления файлов и каталогов. Задания включают сложные операции управления файлами как рекурсивное копирование множества файлов, рекурсивное удаление файлов, перемещение файлов, которые удовлетворяют заданному шаблону. Эта задача включает использование простых и сложных шаблонов, с помощью которых определяется нужный набор файлов, а также использование команды find для обнаружения и осуществления действий над файлами, удовлетворяющими определенному типу, размеру и времени.
1.103.4	Вес 5	Перенаправление потоков и соединение их в определенном порядке для эффективной обработки текста. Задачи включают перенаправление стандартного потока ввода, вывода и потока ошибок, перенаправление выходного потока одной команды на входной поток другой команды, использование вывода одной команды в качестве аргументов для другой команды, а также вывод выходного потока, как на экран, так и в файл.
1.103.5	Вес 5	Управление процессами. Эта задача включает знание о том, как запускать задачи в фоновом и приоритетном режимах, как перевести задачу из фонового режима в приоритетный и наоборот, как запустить процесс, который будет работать без связи с терминалом, а также сигнализировать программе, чтобы она продолжала работать после выхода из системы. Задачи также включают наблюдение за деятельностью активных процессов, выбора и сортировки определенных процессов для отображения на экране, посылку сигналов процессам, уничтожение процессов, а также идентификацию и уничтожение приложений X, которые не уничтожились после завершения X сессии.
1.103.6	Вес 3	Изменение приоритета исполнения процесса. Задача включает запуск программы с высоким или низким приоритетом, определение приоритета процесса, а также изменение приоритета работающего процесса.

1.103.7 <u>Поиск текстовых файлов с помощью регулярных выражений</u>	Вес 3	Манипулирование файлами и текстовыми данными с помощью регулярных выражений. Эта задача включает создание простых регулярных выражений из нескольких элементов. Она также включает использование инструментов с регулярными выражениями для осуществления поиска по файловой системе или содержимому файла.
---	-------	---

1.103.8 <u>Осуществление простых операций редактирования файла с помощью vi</u>	Вес 1	Редактирование текстовых файлов с помощью vi. Эта задача включает перемещение по документу в vi, структурирование документа в vi, вставка, редактирование, удаление, копирование и поиск текста.
--	-------	--

## В начало

### Предварительные замечания

Чтобы извлечь максимум пользы из этого руководства, вы должны иметь базовое представление о Linux, а также компьютер с Linux, на котором вы можете попробовать все команды, рассматриваемые в этом руководстве. Иногда разные версии программ по-разному форматируют свой вывод, поэтому ваши результаты могут не всегда выглядеть в точности как на приведенных листингах и рисунках этого руководства.

## Использование командной строки

Данная глава описывает материал темы 1.103.1, необходимый для сдачи экзамена Junior Level Administration (LPIC-1) 101. Эта тема имеет вес 5.

В этом разделе описываются следующие темы:

- Взаимодействие с командными интерпретаторами и командами
- Команды и последовательности команд
- Определение, использование и экспорт переменных окружения
- История команд и средства редактирования
- Запуск команд, находящихся в переменной окружения PATH так и вне ее
- Использование подстановки команд
- Применение команд рекурсивно к дереву каталогов
- Использование man-страниц (помощи) для поиска информации о командах

Данный раздел дает описание некоторых основных возможностей командного интерпретатора bash. Особый акцент делается на возможностях, необходимых для сертификации. Командный интерпретатор это богатая среда, и мы приветствуем ее дальнейшее самостоятельное изучение. По командным интерпретаторам UNIX и Linux написано много книг и bash в частности.

### Командный интерпретатор bash

Интерпретатор *bash* один из нескольких интерпретаторов, доступных в Linux. Также он называется *Bourne-again shell*, в честь Стивена Борна, создателя ранней версии интерпретатора (*/bin/sh*). Bash по существу совместим с sh, но представляет много улучшений, как в функциональном плане, так и возможностям программирования. Он включает возможности интерпретаторов Korn (ksh) и C (csh), и разрабатывается как POSIX-совместимый интерпретатор.

Прежде, чем мы начнем изучать bash, напомним, что *интерпретатор* -- это программа,

которая принимает и исполняет команды. Он также поддерживает возможности программирования, позволяя составлять сложные конструкции из обычных команд. Эти сложные конструкции или *сценарии* можно сохранить в файлы, которые в свою очередь сами являются новыми командами. Более того, множество команд на типичной Linux системе реализованы как сценарии командного интерпретатора.

Интерпретаторы содержат *встроенные* команды, такие как `cd`, `break` и `exec`. Другие команды являются *внешними*.

Интерпретаторы также используют три стандартных потока ввода/вывода:

- `stdin` это *стандартный поток ввода*, который обеспечивает ввод для команд.
- `stdout` это *стандартный поток вывода*, который обеспечивает отображение результатов выполнения команды в окне терминала.
- `stderr` это *стандартный поток ошибок*, который отображает ошибки, возникающие при работе команд.

Потоки ввода обеспечивают ввод для программ, обычно он связан с клавиатурой терминала. Выходные потоки печатают текстовые символы, обычно на терминал. Терминал изначально был ASCII печатной машинкой или видеотерминалом, но сейчас он часто представляет собой окно на графическом рабочем столе. Более подробно о том, как перенаправлять стандартные потоки ввода/вывода, можно посмотреть дальше в этом руководстве в разделе [Потоки, программные каналы и перенаправление](#). Остальная часть этого раздела сосредоточится на перенаправлении на высоком уровне.

Далее мы будем предполагать, что вы знаете, как попасть в командную строку. Если нет, то статья developerWorks "[Основные задачи начинающих разработчиков Linux](#)" поможет вам разобраться с этой и другими задачами.

Если вы используете Linux систему без графического интерфейса или же вы открыли окно терминала в графическом режиме, то увидите приглашение для ввода команд как в Листинге 1.

### Листинг 1. Примеры типичных пользовательских приглашений

```
[db2inst1@echidna db2inst1]$  
ian@lyrebird:~>  
$
```

Если вы зайдете как пользователь `root` (или суперпользователь), то ваше приглашение может выглядеть, как показано в Листинге 2.

### Листинг 2. Примеры приглашений для пользователя `root` или суперпользователя

```
[root@echidna root]#  
lyrebird:~ #  
#
```

Пользователь `root` имеет значительную власть, поэтому пользуйтесь им с осторожностью. Если у вас привилегии пользователя `root`, то большинство приглашений начинаются со знака решетки (`#`). Приглашение для обычного пользователя, как правило, начинается с другого символа, обычно это знак доллара (`$`). Ваше приглашение может отличаться от того, что написано в примерах данного руководства. Ваше приглашение может включать ваше

пользовательское имя, имя машины, текущий каталог, дату или время, когда было напечатано приглашение, и так далее.

## Некоторые соглашения этого руководства

Руководства developerWorks по экзаменам LPI 101 и 102 включают код примеров из реальных Linux систем, с использованием приглашений по умолчанию для этих систем. В нашем случае приглашение пользователя root начинается с #, так что вы можете отличить его от приглашений обычных пользователей, которые начинаются со знака \$. Это соглашение совпадает с тем, которое используется в книгах по данному предмету. Внимательно смотрите приглашение командного интерпретатора в каждом примере.

## Команды и последовательности

Вы находитесь в командном интерпретаторе, посмотрим, что вы можете теперь сделать. Основная функция командных интерпретаторов состоит в том, что он исполняет ваши команды, посредством которых вы взаимодействуете с Linux системой. В системах Linux (и UNIX) команды состоят из *имени команды, опций и параметров*. У некоторых команд нет ни опций, ни параметров, у других есть опции, но нет параметров, в то время как у третьих нет ни опций, ни параметров.

Если строка содержит символ #, то все последующие символы в ней игнорируются. Таким образом, символ # может означать как начало комментария, так и приглашение пользователя. Дальнейшая интерпретация будет очевидна из контекста.

## Команда Echo

Команда **echo** выводит на терминал список своих аргументов как показано в Листинге 3.

### Листинг 3. Примеры команды echo

```
[ian@echidna ian]$ echo Слово  
Слово  
[ian@echidna ian]$ echo И предложение  
И предложение  
[ian@echidna ian]$ echo Куда подевались пробелы?  
Куда подевались пробелы?  
[ian@echidna ian]$ echo "А вот и пробелы." # и комментарий  
А вот и пробелы.
```

В третьем примере Листинга 3 все промежутки между словами на выходе команды стали одного размера в один пробел. Чтобы этого избежать вам потребуется заключить строку в кавычки, используя или двойные кавычки ("") или одинарные (''). Bash использует *символы разделители*, как пробелы, символы табуляции и символы новой строки для разбиения входной строки на *токены*, которые затем передаются вашей команде. Заключение строки в кавычки подавляет ее разделение и таким образом она является единым токеном. В приведенном выше примере каждый токен после имени команды является параметром, таким образом, у нас получается соответственно 1, 2, 4 и 1 параметр.

У команды echo есть несколько опций. Обычно echo добавляет после своего вывода символ новой строки. Используйте опцию **-n** чтобы она не добавляла символ новой строки. Используйте опцию **-e**, чтобы команда интерпретировала escape-последовательности. Некоторые из них представлены в Таблице 3.

Таблица 3. Echo escape-последовательности

Escape последовательность	Значение
---------------------------	----------

\a	Звонок
\b	Забой последнего символа
\c	Не добавлять символ новой строки (тоже самое, что и опция -n)
\f	Перевод страницы (очищает экран на видео дисплее)
\n	Новая строка
\r	Перевод каретки
\t	Горизонтальная табуляция

### Escape-последовательности и перенос строки

Существует небольшая проблема при использовании обратного слеша в bash. Когда символ обратного слеша () не заключен в кавычки, то он сам служит escape-последовательностью для bash, предохраняя значение следующего символа. Это необходимо для особых метасимволов, которые мы рассмотрим чуть позже. Существует одно исключение из этого правила: обратный слеш, за которым следует перевод строки, заставляет bash проглотить оба символа и считать последовательность как запрос на продолжение строки. Это может быть полезным при разбиении длинных строк, особенно применительно к сценариям.

Чтобы последовательности, описанные выше, правильно обрабатывались командой echo или одной из многих других команд, которые используют похожие escape символы управления, вы должны заключить escape последовательности в кавычки или же включить их в строку, заключенную в кавычки, либо использовать еще один обратный слеш для верной интерпретации символов. Листинг 4 содержит примеры различных вариантов использования \.

### Листинг 4. Примеры использования echo

```
[ian@echidna ian]$ echo -n Нет новой строки
Нет новой строки[ian@echidna ian]$ echo -e "Нет новой строки\c"
Нет новой строки[ian@echidna ian]$ echo "Строка в которой нажали
> клавишу Enter"
Строка в которой нажали
клавишу Enter
[ian@echidna ian]$ echo -e "Строка с escape символом\nновой строки"
Строка с escape символом
новой строки
[ian@echidna ian]$ echo "Строка с escape символом\nновой строки, но без опции -e"
Строка с escape символом\nновой строки, но без опции -e
[ian@echidna ian]$ echo -e Метасимволы с двойным\\n\\tобратным слешем
Метасимволы с двойным
обратным слешем
[ian@echidna ian]$ echo Обратный слеш \
> за которым следует Enter \
> служит как запрос на продолжение строки.
Обратный слеш за которым следует Enter служит как запрос на продолжение строки.
```

Заметим, что bash отображает специальное приглашение (>), когда вы набрали строку с незавершенными кавычками. Ваша входная строка переносится на вторую строку и в неё включает символ новой строки.

### Метасимволы Bash и операторы управления

Bash включает несколько *символов*, которые, будучи не заключенными в кавычки, также служат для разделения входной строки на слова. Кроме пробела такими символами являются '|', '&', ';', '(', ')', '<', и '>'. Некоторые из этих символов мы обсудим более подробно в других

разделах этого руководства. А сейчас заметим, что если вы хотите включить метасимвол как часть вашего текста, то он должен быть заключен в кавычки или же ему должен предшествовать обратный слеш (\) как в Листинге 4.

Новая строка и соответствующие метасимволы или пары метасимволов также служат как *операторы управления*. Такими символами являются '|', '&&', '&', ';', ';;', '||' '(', и ')'. Некоторые из этих операторов управления позволяют вам создавать *последовательности* или *списки* команд.

Простейшая последовательность команд состоит из двух команд, разделенных точкой с запятой (;). Каждая следующая команда исполняется после предыдущей. В любой среде программирования команды возвращают код, свидетельствующий о нормальном или неудачном завершении программы; команды Linux обычно возвращают 0 в случае успешного завершения и ненулевое значение в случае неуспеха. Вы можете осуществлять обработку по условию, используя управляющие операторы && и ||. Если вы разделите две команды управляющим оператором &&, то вторая команда будет выполняться только в том случае, если первая вернула на выходе ноль. Если вы разделили команды с помощью ||, то вторая команда будет исполняться, только если первая вернула ненулевое значение. Листинг 5 содержит некоторые последовательности команд с использованием команды echo. Эти примеры не очень интересны, так как echo возвращает 0, но мы рассмотрим больше примеров, когда научимся использовать большее число команд.

## Листинг 5. Последовательности команд

```
[ian@echidna ian]$ echo line 1;echo line 2; echo line 3
line 1
line 2
line 3
[ian@echidna ian]$ echo line 1&&echo line 2&&echo line 3
line 1
line 2
line 3
[ian@echidna ian]$ echo line 1||echo line 2; echo line 3
line 1
line 3
```

## Выход

Вы можете выйти из командного интерпретатора с помощью команды **exit**. Дополнительно в качестве параметра вы можете задать код выхода. Если вы работаете с командным интерпретатором в терминальном окне в графическом режиме, то в этом случае оно просто закроется. Аналогично, если вы подсоединены к удаленной системе с помощью ssh или telnet (например), то соединение завершится. В интерпретаторе bash вы также можете нажать клавишу **Ctrl** и **d** для выхода.

Давайте рассмотрим еще один оператор управления. Если вы заключите команду или список команд в круглые скобки, то команда или последовательность команд будет выполняться в своей копии командного интерпретатора, таким образом, команда exit выходит из копии командного интерпретатора, а не из того интерпретатора, в котором вы работаете в данный момент. Листинг 6 содержит простые примеры совместно с использованием && и ||.

## Листинг 6. Командные интерпретаторы и последовательности команд

```
[ian@echidna ian]$ (echo В копии интерпретатора; exit 0) && echo OK || echo Bad exit
В копии интерпретатора
```

```
OK
[ian@echidna ian]$ (echo В копии интерпретатора; exit 4) && echo OK || echo Bad exit
В копии интерпретатора
Bad exit
```

## Переменные окружения

При работе в bash, вас окружает совокупность параметров, составляющих вашу *среду*, например, формат вашего приглашения, имя домашнего каталога, ваш рабочий каталог, название вашего интерпретатора, файлы, которые вы открыли, определенные вами функции и так далее. Ваша среда включает множество *переменных* которые можете устанавливать как вы, так и bash. Bash также позволяет вам создавать *переменные оболочки*, которые вы можете экспортить в свою среду для использования другими процессами, запущенными в интерпретаторе или другими интерпретаторами, которые вы можете запустить из текущего интерпретатора.

Как у переменных окружения, так и у переменных оболочки есть *имя*. Ссыльаться на значение переменной можно, поставив перед именем переменной знак '\$'. Некоторые наиболее общие переменные среды bash приведены в Таблице 4.

Таблица 4.Некоторые наиболее общие переменные среды bash

Имя	Значение
USER	Имя зашедшего в систему пользователя
UID	Цифровой идентификатор зашедшего в систему пользователя
HOME	Домашний каталог пользователя
PWD	Текущий рабочий каталог
SHELL	Имя командного интерпретатора
\$	Идентификатор процесса (или <i>PID</i> ) bash (или другого процесса)
PPID	Идентификатор процесса, который породил данный процесс (то есть идентификатор родительского процесса)
?	Код выхода последней команды

На Листинге 7 можно видеть некоторые переменные bash.

## Листинг 7. Переменные среды и shell

```
[ian@echidna ian]$ echo $USER $UID
ian 500
[ian@echidna ian]$ echo $SHELL $HOME $PWD
/bin/bash /home/ian /home/ian
[ian@echidna ian]$ (exit 0);echo $?;(exit 4);echo $?
0
4
[ian@echidna ian]$ echo $$ $PPID
30576 30575
```

## Не используете bash?

Интерпретатор bash принят по умолчанию во многих дистрибутивах Linux. Если вы работаете не с bash, то можете рассмотреть следующие способы, чтобы попрактиковаться в работе с bash.

- Используйте команду  
`chsh -s /bin/bash`  
чтобы изменить интерпретатор по умолчанию. Изменения вступят в силу во время вашего следующего захода в систему.
- Команда  
`su - $USER -s /bin/bash`  
создаст другой процесс, который будет являться дочерним по отношению к вашему текущему интерпретатору. Новый процесс запустит процесс входа в систему с командным интерпретатором bash.
- Создайте пользователя с командным интерпретатором bash для того, чтобы подготовиться к сдаче экзамена LPI.

Вы можете создать или *установить* переменную оболочки, набрав сразу за именем переменной знак равно (=). Переменные чувствительны к регистру, таким образом, var1 и VAR1 -- это две разные переменные. По соглашению переменные, особенно экспортируемые переменные, пишутся в верхнем регистре, но это не обязательное требование. Формально, \$\$ и \$? являются *параметрами* оболочки, а не переменными. Вы можете на них ссылаться, но не присваивать значения.

Когда вы создаете переменную оболочки, то часто захотите *экспортировать* ее в среду так, чтобы она стала доступна другим процессам, которые вы запускаете из интерпретатора. Переменные, которые вы экспортируете **не** доступны родительским интерпретаторам. Для экспортации переменной используется команда `export`. Для удобства вы можете присвоить значение и экспортовать переменную за один шаг.

Чтобы проиллюстрировать присваивание и экспорт, создадим еще один bash из текущего bash интерпретатора, а затем запустим интерпретатор Korn из (ksh) созданного bash. Мы будем использовать команду `ps` для отображения информации о работающих процессах. Более подробно о команде `ps` мы узнаем, когда изучим понятие [статус процесса](#) далее в этом руководстве.

## Листинг 8. Переменные среды и shell

```
[ian@echidna ian]$ ps -p $$ -o "pid ppid cmd"
  PID  PPID CMD
30576 30575 -bash
[ian@echidna ian]$ bash
[ian@echidna ian]$ ps -p $$ -o "pid ppid cmd"
  PID  PPID CMD
16353 30576 bash
[ian@echidna ian]$ VAR1=var1
[ian@echidna ian]$ VAR2=var2
[ian@echidna ian]$ export VAR2
[ian@echidna ian]$ export VAR3=var3
[ian@echidna ian]$ echo $VAR1 $VAR2 $VAR3
var1 var2 var3
[ian@echidna ian]$ echo $VAR1 $VAR2 $VAR3 $SHELL
var1 var2 var3 /bin/bash
[ian@echidna ian]$ ksh
$ ps -p $$ -o "pid ppid cmd"
  PID  PPID CMD
16448 16353 ksh
$ export VAR4=var4
$ echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL
var2 var3 var4 /bin/bash
$ exit
$ [ian@echidna ian]$ echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL
```

```
var1 var2 var3 /bin/bash
[ian@echidna ian]$ ps -p $$ -o "pid ppid cmd"
  PID  PPID CMD
16353 30576 bash
[ian@echidna ian]$ exit
[ian@echidna ian]$ ps -p $$ -o "pid ppid cmd"
  PID  PPID CMD
30576 30575 -bash
[ian@echidna ian]$ echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL
/bin/bash
```

### Примечание:

1. В начале этой последовательности у интерпретатора bash был PID 30576 .
2. У второго интерпретатора bash PID 16353, а его родительский PID 30576, то есть изначальный bash.
3. Мы создали переменные VAR1, VAR2, и VAR3 во втором экземпляре bash, но экспортировали только VAR2 и VAR3.
4. В интерпретаторе Korn, мы создали VAR4. Команда echo отображает значения только переменных VAR2, VAR3 и VAR4, и подтвердила, что VAR1 не была экспортирована. Вы не были удивлены, когда значение переменной SHELL не изменилось, хотя изменилось приглашение ввода? Вы не можете всегда полагаться на SHELL, чтобы определить в каком интерпретаторе идет работа, но команда ps позволит точно определить, что к чему. Заметим, что ps ставит дефис (-) перед первым экземпляром bash, чтобы дать нам понять, что это *исходный командный интерпретатор*.
5. Во втором экземпляре bash мы можем просмотреть VAR1, VAR2 и VAR3.
6. Наконец, когда мы возвращаемся в исходный интерпретатор, ни одна переменная в нем не существует.

Ранее мы обсуждали возможность использования кавычек как одинарных, так и двойных. Между ними есть существенная разница. Интерпретатор осуществляет подстановку shell переменных, находящиеся между двойными кавычками ("quot;), но не осуществляет подстановку, если используются одинарные ('). В предыдущем примере, мы создали новый экземпляр интерпретатора из другого и получили новый идентификатор процесса. Используя опцию -C вы можете передать команду в другой интерпретатор, который исполнит команду и произведет возврат. Если вы передаете строку в качестве команды в одинарных кавычках, то второй экземпляр интерпретатора их снимет и обработает строку. При использовании двойных кавычек подстановка переменных происходит **до того** как осуществляется передача строки, поэтому результаты могут отличаться от того, что вы хотели ожидать. Интерпретатор и команда породят процесс, у которого будет свой PID. Листинг 9 иллюстрирует эти концепции. PID изначального интерпретатора bash выделен другим шрифтом.

### Листинг 9. Кавычки и shell переменные

```
[ian@echidna ian]$ echo "$SHELL" '$SHELL' $$ '$$'
/bin/bash $SHELL 19244 $$ 
[ian@echidna ian]$ bash -c "echo Expand in parent $$ $PPID"
Expand in parent 19244 19243
[ian@echidna ian]$ bash -c 'echo Expand in child $$ $PPID'
Expand in child 19297 19244
```

До сих пор все наши переменные заканчивались пробелом, таким образом, было понятно, где

заканчивается имя переменной. На самом деле имя переменной может только состоять из букв, цифр или символа подчеркивания. Интерпретатор знает, что имя переменной заканчивается, как только встречается другой символ. Иногда необходимо использовать переменные в выражениях, где их значение может быть двусмысленным. В таких случаях вы можете использовать фигурные скобки, чтобы отделить имя переменной как показано в Листинге 10.

### Листинг 10. Использование фигурных скобок с именами переменных

```
[ian@echidna ian]$ echo "-$HOME/abc-"  
-/home/ian/abc-  
[ian@echidna ian]$ echo "-$HOME_abc-"  
--  
[ian@echidna ian]$ echo "-${HOME}_abc-"  
-/home/ian_abc-
```

### Команда env

Команда **env** без каких-либо опций или параметров отображает текущие переменные среды. Вы также можете использовать ее, чтобы выполнить команду в предопределенной среде. Опция **-i** (или просто **-**) очищает текущую среду до того как выполнить команду, в то время как опция **-u** обнуляет переменные среды, которые вы не хотите передавать.

Листинг 11 содержит частичный вывод команды **env** без каких-либо параметров, а затем три примера, запускающие разные интерпретаторы без родительской среды. Внимательно их просмотрите прежде, чем мы их обсудим.

### Листинг 11. Команда env

```
[ian@echidna ian]$ env  
HOSTNAME=echidna  
TERM=xterm  
SHELL=/bin/bash  
HISTSIZE=1000  
SSH_CLIENT=9.27.89.137 4339 22  
SSH_TTY=/dev/pts/2  
USER=ian  
...  
_=bin/env  
OLDPWD=/usr/src  
[ian@echidna ian]$ env -i bash -c 'echo $SHELL; env'  
/bin/bash  
PWD=/home/ian  
SHLVL=1  
_=bin/env  
[ian@echidna ian]$ env -i ksh -c 'echo $SHELL; env'  
_=bin/env  
PATH=/bin:/usr/bin  
[ian@echidna ian]$ env -i tcsh -c 'echo $SHELL; env'  
SHELL: Undefined variable.
```

Заметим, что **bash** установил переменную **SHELL**, но не экспорттировал ее в среду, несмотря на то, что **bash** создал в среде три других переменных. В примере с **ksh** у нас содержится две

переменных окружения, но наша попытка выдать на экран значение переменной SHELL приводит к появлению пустой строки. Наконец, tcsh не создал никаких переменных среды и выдал ошибку, когда мы пытались получить значение переменной SHELL.

### Установка и обнуление переменных

Листинг 11 показал, как ведут себя интерпретаторы при обработке переменных и сред. Хотя это руководство уделяет внимание bash, следует знать, что не все интерпретаторы ведут себя одинаково. Более того, интерпретаторы ведут себя по-разному в зависимости от того, являются ли они *исходными командными интерпретаторами* или нет. Сейчас мы просто скажем, что исходный командный интерпретатор это интерпретатор, который вы получаете при входе в систему; вы можете запустить другие командные интерпретаторы так, что они будут вести себя как исходные если пожелаете. Три интерпретатора в примере выше, запущенные с помощью команды `env -i` не являются исходными интерпретаторами. Попытайтесь передать опцию `-l`, чтобы увидеть разницу при запуске исходного командного интерпретатора.

Давайте рассмотрим нашу попытку отобразить значение переменной SHELL в этих командных интерпретаторах:

1. Когда bash запустился, он установил переменную SHELL, но не экспорттировал ее автоматически в среду.
2. Когда запустился ksh, он не установил переменную SHELL. Однако ссылка на неопределенную переменную среды эквивалентно ссылке на пустое значение.
3. Когда запустился tcsh, то он не установил значение переменной SHELL. В этом случае поведение по умолчанию отлично от ksh (и bash) и в результате сгенерировалась ошибка, когда мы пытались получить доступ к переменной.

Вы можете использовать команду `unset` для обнуления переменной и удаления ее из списка shell переменных. Если переменная была экспорттирована в среду, то она также будет удалена и из среды. Вы можете использовать команду `set` для управления поведением работы bash (или других интерпретаторов). Set является встроенной командой в интерпретаторе, поэтому опции зависят от конкретного интерпретатора. В bash опция `-u` сообщает bash, чтобы он не генерировал ошибку при ссылке на неопределенные переменные, а работал с ними как с пустыми значениями. Вы можете добавить различные опции к `set` с помощью `-` и отключить их с помощью `+`. Вы можете отобразить текущий список опций set с помощью `echo $-`.

### Листинг 12. Unset и set

```
[ian@echidna ian]$ echo $-
himBH
[ian@echidna ian]$ echo $VAR1

[ian@echidna ian]$ set -u;echo $-
himuBH
[ian@echidna ian]$ echo $VAR1
bash: VAR1: unbound variable
[ian@echidna ian]$ VAR1=v1
[ian@echidna ian]$ VAR1=v1;echo $VAR1
v1
[ian@echidna ian]$ unset VAR1;echo $VAR1
bash: VAR1: unbound variable
[ian@echidna ian]$ set +u;echo $VAR1;echo $-
himBH
```

Вы можете использовать команду `set` без каких-либо опций, которая отобразит все ваши shell переменные и их значения (если есть). Есть также другая команда, `declare`, с помощью которой вы можете создавать, экспорттировать и отображать значения shell переменных. О других опциях команд `set` и `declare` вы можете узнать из man-страниц. Мы рассмотрим [man-страницы](#) далее в этом разделе.

## Команда exec

Последняя команда, которую мы рассмотрим в этом разделе это `exec`. Вы можете использовать команду `exec`, чтобы запустить другую команду, которая заместит текущий интерпретатор. В Листинге 13 порождается экземпляр bash, а затем используется `exec`, чтобы заместить его на интерпретатор Korn. После выхода из интерпретатора Korn, вы оказываетесь в исходном интерпретаторе bash (в этом примере PID 22985).

### Листинг 13. Использование exec

```
й
[ian@echidna ian]$ echo $$
22985
[ian@echidna ian]$ bash
[ian@echidna ian]$ echo $$
25063
[ian@echidna ian]$ exec ksh
$ echo $$
25063
$ exit
[ian@echidna ian]$ echo $$
22985
```

## История команд

Если вы набирали команды, по мере того как читали руководство, то могли заметить, что часто используются почти одни и те же команды. Хорошая новость состоит в том, что bash может хранить *историю* ваших команд. По умолчанию история включена. Вы можете отключить ее с помощью команды `set +o history` и включить с помощью команды `set -o history`. Переменная среды HISTSIZE сообщает bash о том, сколько надо хранить строк. Набор других свойств определяет поведение и работу истории. Подробностисмотрите в man-страницах bash.

Вот некоторые команды, которые вы можете использовать для работы с историей:

### `history`

Отображает всю историю

### `historyN`

Отображает последние *N* строк вашей истории

### `history -dN`

Удаляет строку *N* из вашей истории; это можно использовать, если, например, вы хотите удалить строку, содержащую пароль

### `!!`

Ваша последняя введенная команда

### `!N`

*N*яя команда истории

### `!-N`

Команда, отстоящая на *N* шагов от текущей в истории (!-1 эквивалентно !!)

### `!#`

Текущая команда, которую вы набираете

**!string**

Самая недавняя команда, которая начинается со строки *string*

**?string?**

Самая последняя команда, содержащая строку *string*

Вы можете использовать двоеточие (:), за которым следует определенное значение, чтобы получить доступ или изменить команду в истории. Листинг 14 показывает некоторые возможности истории.

#### Листинг 14. Управление историей

```
[ian@echidna ian]$ echo $$  
22985  
[ian@echidna ian]$ env -i bash -c 'echo $$'  
1542  
[ian@echidna ian]$ !!  
env -i bash -c 'echo $$'  
1555  
[ian@echidna ian]$ !ec  
echo $$  
22985  
[ian@echidna ian]$ !en:s/$$/${PPID}/  
env -i bash -c 'echo ${PPID}'  
22985  
[ian@echidna ian]$ history 6  
1097 echo $$  
1098 env -i bash -c 'echo $$'  
1099 env -i bash -c 'echo $$'  
1100 echo $$  
1101 env -i bash -c 'echo ${PPID}'  
1102 history 6  
[ian@echidna ian]$ history -d1100
```

Команды в Листинге 14 делают следующее:

1. Вывод PID текущего интерпретатора
2. Запуск команды echo в новом экземпляре интерпретатора и вывод его PID
3. Запустить последнюю команду
4. Перезапустить команду, начинающуюся с 'ec'; произойдет запуск первой команды в этом примере
5. Запустить последнюю команду, начинающуюся с 'en', но заменить '\${PPID}' на '\$\$', поэтому на самом деле отобразится родительский PID
6. Отобразить последние 6 команд истории
7. Удалить команду под номером 1100, последняя команда echo

Вы можете редактировать истории в интерактивном режиме. Интерпретатор bash использует библиотеку readline для управления редактированием команд и истории. По умолчанию, клавиши и комбинации клавиш, которые используются для перемещения по истории или редактированию строк соответствуют тем, что используются в редакторе GNU Emacs. В Emacs комбинации клавиш обычно обозначаются как **C-x** или **M-x**, где **x** это обычная клавиша, а **C** и **M** это *Control* и *Meta* клавиши соответственно. На типичном PC клавиша **Ctrl** соответствует клавише Emacs Control, а клавиша **Alt** соответствует клавише Meta. В Таблице 5 содержатся некоторые доступные функции редактирования истории.

Кроме комбинаций клавиш, показанных в Таблице 5, клавиши курсора, а также Home и End клавиши используются естественным образом для работы с историей. Дополнительные функции, а также возможности настройки опций с помощью файла инициализации readline (обычно это `inputrc` в вашем домашнем каталоге) можно найти в `man`-страницах.

Таблица 5. Редактирование истории с помощью команд `emacs`

Команда	Клавиатура PC	Описание
C-f	Стрелка вправо	Перейти на один знак вправо
C-b	Стрелка влево	Перейти на один знак влево
M-f	Alt-f	Перейти в начало следующего слова; В графических средах эта комбинация приводит к открытию меню <b>File</b> текущего окна
M-b	Alt-b	Перейти к началу предыдущего слова
C-a	Home	Перейти к началу строки
C-e	End	Перейти к концу строки
Backspace	Backspace	Удалить символ перед курсором
C-d	Del	Удалить символ, стоящий сразу после курсора (функции Del и Backspace можно поменять местами)
C-k	Ctrl-k	Удалить (убить) все до конца строки и сохранить для последующего использования
M-d	Alt-d	Удалить (убить) до конца слова и сохранить текст для последующего использования
C-y	Ctrl-y	Вставить текст, удаленный командой убить

Если вы предпочитаете управлять историей в режиме `vi`, то используйте команду `set -o vi`, чтобы переключиться в режим `vi`. Можете переключиться обратно в режим `emacs` с помощью команды `set -o emacs`. Когда вы извлекаете команду в режиме `vi`, то находитесь изначально в режиме вставки `vi`. Более подробно о редакторе `vi` смотрите в разделе [Редактирование файлов в vi](#).

## Пути

Одни команды `bash` являются встроенными, другие же наоборот внешними. Давайте теперь рассмотрим внешние команды и как их запускать, а также как отличить внутреннюю команду.

## Где интерпретатор ищет команды?

Внешние команды представляют собой файлы в файловой системе. Дальше раздел [Простое управление файлами](#) этого руководства и руководства для Темы 104 раскрывают необходимые подробности. В системах Linux и UNIX все файлы являются частью огромного дерева, конем которого является `/`. В рассматриваемых выше примерах нашим текущим каталогом был домашний каталог пользователя. У обычных пользователей домашние каталоги находятся в `/home` каталоге, то есть `/home/ian`, в моем случае. Домашний каталог `root` находится в `/root`. После того как вы набрали команду, `bash` ищет ее в списке *каталогов поиска по умолчанию*, который представляет собой список каталогов, разделенных двоеточием и хранящийся в переменной окружения `PATH`.

Если вы хотите знать какая команда будет выполнена, если вы напечатаете определенную строку, то используйте команду `which` или `type`. В Листинге 15 показан мой путь по умолчанию, а также расположение нескольких команд.

## Листинг 15. Поиск месторасположения команд

```
[ian@echidna ian]$ echo $PATH  
/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/ian/bin
```

```
[ian@echidna ian]$ which bash env zip xclock echo set ls
alias ls='ls --color=tty'
/bin/ls
/bin/bash
/bin/env
/usr/bin/zip
/usr/X11R6/bin/xclock
/bin/echo
/usr/bin/which: no set in (/usr/local/bin:/bin:/usr/bin:/usr/X11R6/b
in:/home/ian/bin)
[ian@echidna ian]$ type bash env zip xclock echo set ls
bash is /bin/bash
env is /bin/env
zip is /usr/bin/zip
xclock is /usr/X11R6/bin/xclock
echo is a shell builtin
set is a shell builtin
ls is aliased to `ls --color=tty'
```

Заметим, что все каталоги в пути заканчиваются на `/bin`. Это общепринятое соглашение, но не требование. Команда `which` доложила нам, что команда `ls` является псевдонимом и что команда `set` не может быть найдена. В этом случае это можно интерпретировать, что команды либо не существует, либо она является встроенной. Команда `type` сообщила нам, что команда `ls` на самом деле является псевдонимом, а также она определила, что команда `set` является встроенной командой интерпретатора; также она сообщила, что есть встроенная команда `echo`, а также есть такая команда в `/bin`, которую мы нашли с помощью команды `which`. Эти две команды по-разному осуществляют свой вывод.

Мы видели, что команда `ls`, используемая для просмотра содержимого каталогов, на самом деле является псевдонимом. Псевдонимы представляют удобный способ использования команд с различными наборами опций или же просто для альтернативного именования команды. В нашем примере опция `--color=tty` заставляет подсвечивать список файлов каталога в зависимости от типа файлов и каталогов. Попробуйте запустить `dircolors --print-database`, чтобы увидеть коды цветов, а также, какие цвета используются для конкретного типа файла.

У каждой из этих команд есть дополнительные опции. В зависимости от ваших требований вы можете использовать ту или иную команду. Я предлагаю использовать `which`, когда уверен, что найду исполняемый файл и мне просто нужен его полный путь. Команда `type` выдает мне более точную информацию, которая мне иногда необходима в сценариях.

### Запуск других команд

В Листинге 15 мы видели, что полный путь исполняемых файлов начинается с корневого каталога `/`. Например, программа `xclock` это на самом деле `/usr/X11R6/bin/xclock`, файл, расположенный в каталоге `/usr/X11R6/bin`. Если программа **не находится** в переменной `PATH`, то вы можете запустить ее, указав полный путь к программе и саму программу. Существует два вида путей, которые вы можете использовать:

- *Абсолютные* пути, которые начинаются с `/`, такие как мы видели в Листинге 15 (`/bin/bash`, `/bin/env` и так далее).
- *Относительные* пути эти пути относительно вашего *текущего рабочего каталога*, имя которого можно получить с помощью команды `pwd`. Такие команды не начинаются с `/`, но по крайней мере содержат один символ `/`.

Вы можете использовать абсолютные пути в независимости от вашего текущего рабочего каталога, но возможно будете использовать относительные пути, когда команда находится

недалеко от текущего каталога. Предположим, что вы разрабатываете новую версию классической программы "Hello World!" в подкаталоге mytestbin вашего домашнего каталога. Возможно вы захотите использовать относительный путь и запустить команду как `mytestbin/hello`. Существует два специальных имени, которые вы можете использовать в указании пути; простая точка (.) ссылается на текущий каталог, и пара точек (..), которые ссылаются на родительский каталог текущего каталога. Так как ваш домашний каталог не находится в переменной окружения PATH (и так и должно быть), то вам понадобится указать явный путь к файлу, который вы хотите запустить из своего домашнего каталога. Например, если у вас есть копия программы hello в вашем домашнем каталоге, то для ее запуска можете просто использовать команду `./hello`. Вы можете использовать как. так и .. как часть абсолютного пути, хотя одинарная . не очень полезна в данном случае. Вы можете использовать тильду (~) для ссылки на свой домашний каталог и `~username` для ссылки на домашний каталог пользователя `username`. Некоторые примеры приведены в Листинге 16.

## Листинг 16. Абсолютные и относительные пути

```
[ian@echidna ian]$ /bin/echo Use echo command rather than builtin
Use echo command rather than builtin
[ian@echidna ian]$ /usr/../bin/echo Include parent dir in path
Include parent dir in path
[ian@echidna ian]$ /bin/../../echo Add a couple of useless path components
Add a couple of useless path components
[ian@echidna ian]$ pwd # See where we are
/home/ian
[ian@echidna ian]$ ../../bin/echo Use a relative path to echo
Use a relative path to echo
[ian@echidna ian]$ myprogs/hello # Use a relative path with no dots
-bash: myprogs/hello: No such file or directory
[ian@echidna ian]$ mytestbin/hello # Use a relative path with no dots
Hello World!
[ian@echidna ian]$ ./hello # Run program in current directory
Hello World!
[ian@echidna mytestbin]$ ~/mytestbin/hello # run hello using ~
Hello World!
[ian@echidna ian]$ ../hello # Try running hello from parent
-bash: ../hello: No such file or directory
```

## Смена рабочего каталога

Также как вы можете исполнять программы из различных каталогов, вы можете изменять ваш текущий рабочий каталог, используя команду `cd`. Аргументом для `cd` должен быть абсолютный или относительный путь до каталога. В команде при указании путей вы также можете использовать ., .., ~, и `~username`. Если вы наберете `cd` без параметров, то перейдете в домашний каталог. Передача в качестве параметра одиночного (-) означает переход в предыдущий рабочий каталог. Домашний каталог хранится в переменной окружения HOME, а предыдущий каталог хранится в переменной OLDPWD, поэтому `cd` эквивалентно `cd $HOME`, а `cd -` эквивалентно `cd $OLPWD`. Обычно мы коротко говорим о *смене каталога* вместо полной *смены текущего рабочего каталога*.

Что касается команд, существует переменная среды CDPATH, которая содержит список каталогов, разделенных двоеточием, в которых должен происходить поиск (в дополнение к текущему рабочему каталогу), при разрешении относительных путей. Если решение использует путь из CDPATH, то `cd` напечатает на выходе полный путь найденного каталога. Обычно удачная смена каталога сопровождается появлением нового приглашения или немного модифицированного приглашения. Некоторые примеры показаны в Листинге 17.

## Листинг 17. Смена каталогов

```
[ian@echidna home]$ cd /;pwd  
/  
[ian@echidna /]$ cd /usr/X11R6;pwd  
/usr/X11R6  
[ian@echidna X11R6]$ cd ;pwd  
/home/ian  
[ian@echidna ian]$ cd -;pwd  
/usr/X11R6  
/usr/X11R6  
[ian@echidna X11R6]$ cd ~ian/..;pwd  
/home  
[ian@echidna home]$ cd ~;pwd  
/home/ian  
[ian@echidna ian]$ export CDPATH=~  
[ian@echidna mytestbin]$ cd /;pwd  
/  
[ian@echidna /]$ cd mytestbin  
/home/ian/mytestbin
```

## Рекурсивное применение команд

Многие Linux команды можно применять рекурсивно ко всем файлам в дереве каталогов. Например, у команды `ls` есть опция `-R` для рекурсивной выдачи списка каталогов, а у команд `cp`, `mv`, `rm`, и `diff` есть опция `-r` для рекурсивного применения. Раздел [Простое управление файлами](#) рассматривает рекурсию более подробно.

## Подстановка команд

У bash есть чрезвычайно мощная возможность передачи результата одной программы на вход другой; это называется *подстановкой команды*. Это можно сделать, заключив команду, результаты которой вам нужны, в апострофы (''). При использовании множественных вложенных команд можно заключать команду между \$( и ) .

В предыдущем руководстве "[Подготовка к экзамену LPI 101 \(тема 102\): Установка Linux и управление пакетами](#)" мы видели, что команда `rpm` может сказать какому пакету принадлежит какая команда; здесь было удобно применять подстановку команды. Теперь вы знаете, что мы действительно это делали.

Подстановка команды является бесценным инструментом при написании сценариев, а также при использовании в командной строке. В Листинге 18 показан пример, как получить абсолютный путь каталога из относительного, как найти пакет, который предоставляет команду `/bin/echo`, и как (будучи root) просмотреть метки трех разделов на жестком диске. Последний использует команду `seq` для создания последовательности целых чисел.

## Листинг 18. Подстановка команды

```
[ian@echidna ian]$ echo '../../usr/bin' dir is $(cd ../../usr/bin;pwd)  
../../usr/bin dir is /usr/bin  
[ian@echidna ian]$ which echo  
/bin/echo  
[ian@echidna ian]$ rpm -qf `which echo`  
sh-utils-2.0.12-3  
[ian@echidna ian]$ su -  
Password:  
[root@echidna root]# for n in $(seq 7 9); do echo p$n `e2label /dev/hda$n`;done  
p7 RH73
```

## Man-страницы

В последней теме раздела этого руководства рассмотрим, как получить справку по командам Linux с помощью man-страниц и других видов документации.

### Man-страницы и разделы

Главный (и традиционный) источник документации -- это *man-страницы*, доступ к которым можно получить с помощью команды **man**. На Рисунке 1 показана man-страница для команды **man**. Используйте команду **man man** для получения этой информации.

**Рисунок 1. Man-страница для команды man**

The screenshot shows a terminal window titled 'ian@echidna-' with the command 'man(1)' entered. The window displays the man(1) manual page. The page is divided into sections:

- 1 NAME**: Describes 'man' as a command to format and display online manual pages and 'manpath' as a command to determine user's search path for man pages.
- 2 SYNOPSIS**: Shows the command line syntax: man [-acdfFhkKtwW] [-path] [-m system] [-p string] [-C config\_file] [-M pathlist] [-P pager] [-S section\_list] [section] name ...
- 3 DESCRIPTION**: Provides a detailed description of the 'man' command, explaining how it formats and displays manual pages. It notes that if a section is specified, 'man' only looks in that section of the manual. The 'name' parameter is typically the name of a command, function, or file. If it contains a slash (/), 'man' interprets it as a file specification, such as ./foo.5 or even man /cd/foo/bar.1.gz. It also describes where 'man' looks for manual page files.
- 4 OPTIONS**: Lists options and their descriptions:
  - C config\_file**: Specifies the configuration file to use; the default is /etc/man.config. (See man.conf(5).)
  - M path**: Specifies the list of directories to search for man pages. Separate the directories with colons. An empty list is the same as not specifying -M at all. See SEARCH PATH FOR MANUAL PAGES.
  - P pager**: Specifies which pager to use. This option overrides the MANPAGER environment variable, which in turn overrides the PAGER variable. By default, 'man' uses /usr/bin/less -isr.

На Рисунке 1 представлены некоторые типичные пункты man-страниц:

- Заголовок с именем команды, за которым в скобках следует номер раздела
- Имя команды и другие похожие команды, которые обсуждаются в этой man-странице
- Список опций и параметров примених к команде
- Короткое описание команды
- Подробное описание каждой опции

Также вы можете найти разделы по использованию, как сообщать ошибки, информацию об авторе, а также список других команд. Например, man-страница для **man** сообщает, что существуют дополнительные команды (и их руководства):

apropos(1), whatis(1), less(1), groff(1) и man.conf(5).

Обычно общими для man-страниц являются 8 разделов. Большинство страниц обычно ставится при установке пакета, поэтому если пакет не установлен, то почти наверняка у вас будут отсутствовать его man-страницы. Кроме того, некоторые разделы страниц могут быть пустыми или почти пустыми. Наиболее общие разделы man-страниц это:

1. Команды пользователя (env, ls, echo, mkdir, tty)
2. Системные вызовы или функции ядра (link, sethostname, mkdir)
3. Библиотечные функции (acosh, asctime, btree, locale, XML::Parser)
4. Информация по оборудованию (isdn\_audio, mouse, tty, zero)
5. Описание формата файлов (keymaps, motd, wvdial.conf)
6. Игры (заметим, что многие игры теперь работают в графическом режиме, поэтому могут иметь собственную систему помощи, а не man-страницы)
7. Разное (arp, boot, regex, unix utf8)
8. Системное администрирование (debugfs, fdisk, fsck, mount, renice, rpm)

Другие разделы могут включать *9* для документации по ядру Linux, *пд* для новой документации, *о* для старой документации и *l* для локальной документации.

Некоторые записи могут встречаться в нескольких разделах. Наши примеры показали, что `mkdir` содержится в разделах 1 и 2, а `tty` в разделах 1 и 4. Вы можете определить определенный раздел, например, `man 4 tty` или `man 2 mkdir`, или вы можете использовать опцию `-a` для получения списка всех разделов man-страниц.

Вы заметили на рисунке, что у `man` много опций, которые вы можете сами посмотреть. Сейчас давайте быстро взглянем на раздел команд "See also", имеющих отношение к `man`.

### See also

Две важнейших команды, имеющих отношение к `man`, это `whatis` и `apropos`. Команда `whatis` ищет man-страницы для указанного вами имени и отображает информации об имени из соответствующих man-страниц. Команда `apropos` осуществляет поиск по ключевым словам в man-страниц и выводит те, которые содержат ваше слово. В Листинге 19 эти команды представлены.

### Листинг 19. Примеры команд `whatis` и `apropos`

```
[ian@lyrebird ian]$ whatis man
man      (1) - format and display the on-line manual pages
man      (7) - macros to format man pages
man [manpath]  (1) - format and display
                  the on-line manual pages
                  man.conf [man]      (5) - configuration data for man
[ian@lyrebird ian]$ whatis mkdir
                  mkdir          (1) - make directories
                  mkdir          (2) - create a directory
[ian@lyrebird ian]$ apropos mkdir
                  mkdir          (1) - make directories
                  mkdir          (2) - create a directory
                  mkdirhier     (1x) - makes a directory hierarchy
```

Между прочим если вы не можете найти man-страницу для `man.conf`, то попробуйтесь запустить `man man.conf ig`.

Вывод на экран команды `man` осуществляется специальная программа постраничного вывода. На большинстве Linux систем такой программой будет `less`. Другим вариантом может быть более старая программа `more`. Если вы хотите напечатать страницу, то определите опцию `-t` для форматирования страницы и печати, используя программу `groff` или `troff`.

У программы вывода `less` есть несколько команд, облегчающих поиск строк в отображаемом тексте. Используйте команду `man less`, чтобы узнать больше о `/`(поиск вперед), `?(поиск назад)` и `n` (для последнего произведенного поиска), а также о многих других командах.

## Другие источники документации

В дополнение к man-страницам, доступным из командной строки, фонд Free Software Foundation создал большое число *info* файлов, которые обрабатываются программой *info*. Она обладает большими возможностями навигации, в том числе и возможностью перехода в другую секцию. Наберите **man info** или **info info**, чтобы получить больше информации. Не все команды документированы в *info*, поэтому вы можете использовать как man-страницы так и *info*.

Существует несколько графических интерфейсов к man-страницам, как например **xman** (из проекта XFree86) и **yelp** (браузер помощи Gnome 2.0).

Если вы не можете найти справки по команде, попытайтесь запустить команду с опцией **--help**. Так вы, возможно, узнаете то что хотели или получите подсказку, где ещё можно поискать.

Следующий раздел посвящен обработки текстовых потоков с помощью фильтров.

| [предыдущая](#) | [следующая](#)

## Текстовые потоки и фильтры

Этот раздел описывает материал темы 1.103.2, необходимый для экзамена Junior Level Administration (LPIC-1) 101. Тема имеет вес 6.

В этом разделе вы узнаете о следующих темах:

- Посылке текстовых файлов и выходных потоков в фильтр с целью модификации вывода
- Использование стандартных UNIX команд из пакета GNU textutils

## Фильтрация текста

Фильтрация текста -- это процесс преобразований над входным потоком текста до того как он будет выдан в выходной поток. Хотя как входной, так и выходной поток могут поступать из файла, в системах Linux и UNIX фильтрация преимущественно осуществляется через *конвойер* команд, когда вывод одной команды *связывается* или *перенаправляется* на ввод следующей команды. Программные каналы и перенаправления более подробно рассмотрены в разделе [Потоки, программные каналы и перенаправления](#), но сейчас давайте взглянем на конвойер и простое перенаправление вывода с помощью операторов **|** и **>**.

## Конвойер с помощью **|**

Вспомним из предыдущего раздела, что интерпретатор оперирует с тремя стандартными потоками ввода/вывода:

- *stdin* это *стандартный поток ввода*, через который поступает ввод командам.
- *stdout* это *стандартный выходной поток*, через который команды выводят свой выход.
- *stderr* это *стандартный поток ошибок*, через который выводятся ошибки в командах.

До сих пор, в этом руководстве, ввод представлял собой параметры, которые мы передавали командам, а вывод отображался на терминал. Многие команды обработки текста (фильтры) могут принимать входной поток, как из стандартного ввода, так и из файла. Чтобы использовать выход команды *command1*, как входной фильтр *command2* вы должны соединить команды с помощью операции конвойеризации (**|**), как показано в Листинге 20.

## Листинг 20. Связывание выхода command 1 со входом command2

```
command1 | command2
```

У любой из команд могут быть опции или аргументы, как вы увидите далее в этом разделе. Вы можете также использовать | для перенаправления вывода command2 в этом конвейере на вход другой команде, command3. Конструируя длинные конвейеры из команд, каждая из которых выполняет свою задачу, можно понять философию выполнения задач в Linux и UNIX. Также иногда вы будете видеть знак дефиса (-) вместо имени файла в качестве аргумента команды, в том значении, что ввод будет поступать из stdin, а не из файла.

### Перенаправление вывода с помощью >

Приятно создавать конвейеры из нескольких команд и наблюдать результат на терминале, но бывают случаи, когда надо сохранить вывод в файл. Это можно сделать с помощью оператора перенаправления вывода (>).

В этой части руководства мы будем использовать небольшие файлы, поэтому давайте создадим каталог lpi103 и перейдем в него. Мы будем использовать > для перенаправления вывода echo в файл text1. Все это показано в Листинге 21. Заметим, что вывод не отображается на терминале, потому что он был перенаправлен в файл.

## Листинг 21. Перенаправление вывода command 1 в файл

```
[ian@echidna ian]$ mkdir lpi103  
[ian@echidna ian]$ cd lpi103  
[ian@echidna lpi103]$ echo -e "1 apple\n2 pear\n3 banana">>text1
```

Теперь, имея в арсенале простые инструменты для создания конвейера и перенаправления, взглянем на наиболее распространенные утилиты обработки текста в UNIX и Linux. Этот раздел описывает некоторые простые возможности; чтобы узнать больше, смотрите соответствующие страницы руководств по этим командам.

### Cat, tac, od и split

Вы создали файл test1, теперь вы захотите посмотреть его содержимое. Используйте команду cat (сокращение от *catenate*), чтобы отобразить содержимое файла на stdout. Листинг 22 проверяет содержимое файла, созданного выше.

## Листинг 22. Вывод содержимого файла с помощью cat

```
[ian@echidna lpi103]$ cat text1  
1 apple  
2 pear  
3 banana
```

Команда cat принимает ввод из stdin, если вы не определите имя файла (или если напишите - как имя файла). Давайте используем эту возможность, а также перенаправление вывода, чтобы создать еще один текстовый файл как в Листинге 23.

### Листинг 23. Создание текстового файла с помощью cat

```
[ian@echidna lpi103]$ cat>text2
9      plum
3      banana
10     apple
```

В Листинге 23 **cat** продолжает читать из `stdin` до конца файла. Используйте комбинацию **Ctrl-d** (нажмите **Ctrl**, а затем нажмите **d**), чтобы послать сигнал конца файла. Такая же комбинация клавиш используется для выхода из `bash`. Заметим, что клавиша `tab` позволяет выровнить в столбец имена фруктов.

Случайно вам захотелось отобразить файл в обратном порядке. Разумеется, для этого тоже существует текстовый фильтр под названием **tac** (перестановка букв в **cat**). Листинг 24 отображает как новый файл `text2`, так и старый `text1` в обратном порядке. Заметим, как просто соединились два файла.

### Листинг 24. Реверсивное отображение с помощью tac

```
[ian@echidna lpi103]$ tac text2 text1
10     apple
3      banana
9      plum
3 banana
2 pear
1 apple
```

Теперь положим, что вы отобразили два текстовых файла с помощью `cat` и `tac` и заметили разницу в выравнивании. Чтобы понять, почему это так, необходимо взглянуть на управляющие символы в файле. Так как они не имеют графического представления, то нам необходимо создать *дамп* файла в формате, который позволит вам найти и интерпретировать эти особые символы. Пакет текстовых утилит GNU включает команду **od** (или *OctalDump*) специально для этой цели.

У команды **od** есть несколько опций, как например **-A** для управления основанием смещений файла и **-t** для управления формой отображения содержимого файла. Основание может быть **o**, (восьмиричное - по умолчанию), **d** (десятичное), **x** (шестнадцатиричное) или **n** (смещения не отображаются). Вы можете отобразить файл в виде восьмиричном, шестнадцатиричном, десятичном, с плавающей точкой, ASCII с escape последовательностями или именованными символами (**nl** для новой строки, **ht** для горизонтальной табуляции и так далее). В Листинге 25 представлены некоторые доступные форматы дампа файла `text2`.

### Листинг 25. Дамп файлов с помощью od

```
[ian@echidna lpi103]$ od text2
0000000 004471 066160 066565 031412 061011 067141 067141 005141
0000020 030061 060411 070160 062554 000012
0000031
[ian@echidna lpi103]$ od -A d -t c text2
0000000 9 \t p l u m \n 3 \t b a n a n a \n
0000016 1 0 \t a p p l e \n
0000025
[ian@echidna lpi103]$ od -A n -t a text2
```

```
9 ht p l u m nl 3 ht b a n a n a nl
1 0 ht a p p l e nl
```

### Замечание:

1. Опция **-A** утилиты **cat** предоставляет альтернативный способ увидеть, где завершаются строки и символы табуляции. Для получения подробной информации смотрите man-страницы.
2. Если у вас есть знания об устройстве ЭВМ, то возможно вас заинтересует утилита **hexdump**, которая является частью другого набора утилит. Она здесь не рассматривается, поэтому обратитесь к man-страницам.

Наши тестовые файлы очень малы, но иногда требуется разбить большие файлы на несколько небольших. Например, вы хотите разбить большой файл на куски объемом с CD, чтобы их записать да CD и отправить по почте кому-нибудь, кто может создать для вас DVD. Команда **split** сделает это таким образом, что **cat** можно будет использовать для простого воссоздания файла. По умолчанию, файлы на выходе команды **split** имеют префикс 'x' в имени, за которым следует суффикс 'aa', 'ab', 'ac', ..., 'ba', 'bb' и так далее. Существуют опции, позволяющие изменять эти умолчания. Вы также можете управлять размером выходных файлов, как в строках, так и в байтах. В Листинге 26 происходит разделение двух текстовых файлов с разными префиксами в именах выходных файлов. Мы разделим **text1** на файлы, содержащие не более двух строк, а **text2** на файлы размером не более 18 байт. Затем мы используем **cat** для отображения различных частей, а также для отображения всего файла, используя *подстановку*, которая рассмотрена в разделе [шаблоны и подстановки](#) позже в этом руководстве.

### Листинг 26. Разделение и воссоединение с помощью **split** и **cat**

```
[ian@echidna lpi103]$ split -l 2 text1
[ian@echidna lpi103]$ split -b 18 text2 y
[ian@echidna lpi103]$ cat yaa
9      plum
3      banana
10[ian@echidna lpi103]$ cat yab
     apple
[ian@echidna lpi103]$ cat y*
9      plum
3      banana
10     apple
```

Заметим, что получившийся файл **yab** не содержит символ новой строки, поэтому наше приглашение было смещено, когда мы использовали **cat** для его отображения.

### **Wc, head и tail**

**Cat** и **tac** отображают файл целиком. Для маленьких файлов, с которыми имеем дело мы, это нормально, но предположим, что у вас большой файл. Для начала вы можете использовать команду **wc** (*Word Count*), чтобы посмотреть размер файла. Команда **wc** отображает число строк, слов и байт в файле. Вы также можете узнать число байт с помощью **ls -l**. Листинг 27 показывает длинный формат списка каталогов для двух файлов, а также вывод команды **wc**.

### Листинг 27. Использование **wc** с текстовыми файлами

```
[ian@echidna lpi103]$ ls -l text*
-rw-rw-r-- 1 ian ian 24 Sep 23 12:27 text1
-rw-rw-r-- 1 ian ian 25 Sep 23 13:39 text2
[ian@echidna lpi103]$ wc text*
      3       6      24 text1
      3       6      25 text2
      6      12      49 total
```

Различные опции позволяют вам контролировать вывод команды **wc** или отображать другую информацию, как например максимальная длина строки. Более подробно написано в man-страницах.

Две команды позволяют вам отобразить как начало (*head*) так и конец файла (*tail*). Это команды соответственно **head** и **tail**. Их можно использовать как фильтры, или же они могут принимать имя файла как аргумент. По умолчанию они отображают первые (или последние) 10 строк файла или потока. В Листинге 28 использована команда **dmesg** для отображения сообщений о загрузке совместно с **wc**, **tail** и **head** чтобы узнать, что всего имеется 177 сообщений, затем отображаются последние 10 строк, и, наконец, отображаются шесть сообщений, начиная с 15 от конца. Некоторые строки были обрезаны в этом выводе (об этом свидетельствует ...).

#### Листинг 28. Использование **wc**, **head** и **tail** для отображения загрузочных сообщений

```
[ian@echidna lpi103]$
[ian@echidna lpi103]$ dmesg | wc
      177     1164    8366
[ian@echidna lpi103]$ dmesg | tail
i810: Intel ICH2 found at IO 0x1880 and 0x1c00, MEM 0x0000 and ...
i810_audio: Audio Controller supports 6 channels.
i810_audio: Defaulting to base 2 channel mode.
i810_audio: Resetting connection 0
ac97_codec: AC97 Audio codec, id: ADS98 (Unknown)
i810_audio: AC'97 codec 0 Unable to map surround DAC's (or ...
i810_audio: setting clocking to 41319
Attached scsi CD-ROM sr0 at scsi0, channel 0, id 0, lun 0
sr0: scsi3-mmc drive: 0x/32x writer cd/rw xa/form2 cdda tray
Uniform CD-ROM driver Revision: 3.12
[ian@echidna lpi103]$ dmesg | tail -n15 | head -n 6
agpgart: Maximum main memory to use for agp memory: 941M
agpgart: Detected Intel i845 chipset
agpgart: AGP aperture is 64M @ 0xf4000000
Intel 810 + AC97 Audio, version 0.24, 13:01:43 Dec 18 2003
PCI: Setting latency timer of device 00:1f.5 to 64
i810: Intel ICH2 found at IO 0x1880 and 0x1c00, MEM 0x0000 and ...
```

Другим популярным использованием **tail** является *слежение* за файлом с помощью опции **-f**, обычно построчно. Это может полезным, если у вас есть фоновый процесс, который генерирует вывод в файл, и вы хотите проверить, что он сделал. В этом режиме **tail** будет работать, пока вы не прекратите его работу (с помощью **Ctrl-c**), отображая строки по мере того, как они будут поступать в файл.

## Expand, unexpand и tr

Когда мы создали файлы text1 и text2, то использовали в text2 символы табуляции. Иногда требуется заменить символы табуляции на другие символы и наоборот. Команды **expand** и **unexpand** этим и занимаются. Опция **-t** в обеих командах позволяет установить шаг табуляции. Таким образом, каждый символ табуляции заменяется на этот шаг. В Листинге 29 показано, как заменить символы табуляции в text2 на пробелы, а также странная последовательность **expand** и **unexpand** которая переупорядочивает текст в text2.

### Листинг 29. Использование expand и unexpand

```
[ian@echidna lpi103]$ expand -t 1 text2
9 plum
3 banana
10 apple
[ian@echidna lpi103]$ expand -t8 text2|unexpand -a -t2|expand -t3
9      plum
3      banana
10     apple
```

К сожалению, вы не можете использовать **unexpand**, чтобы заменить пробелы в text1 на символы табуляции, так как **unexpand** необходимо по крайней мере два пробела для преобразования их в символ табуляции. Однако вы можете использовать команду **tr** которая переводит символы из одного набора (*set1*) в соответствующие символы другого набора(*set2*). В Листинге 30 показано, как использовать **tr**, чтобы преобразовать пробелы в символы табуляции. Так как **tr** это фильтр, то входные данные для него вы генерируете с помощью команды **cat**. Этот пример также иллюстрирует применение **-** для обозначения стандартного ввода в **cat**.

### Листинг 30. Использование expand и unexpand

```
[ian@echidna lpi103]$ cat text1 |tr ' ' '\t'|cat - text2
1      apple
2      pear
3      banana
9      plum
3      banana
10     apple
```

Если вы не уверены в том, что происходит в последних двух примерах, то наберите **od**, чтобы проверить каждую стадию конвейера; например  
**cat text1 |tr ' ' '\t' | od -tc**

## Pr, nl и fmt

Команда **pr** используется для форматирования файлов перед печатью. Заголовок по умолчанию включает имя файла и даты и время создания файла, а также номер страницы и двух пустых строк сносок. Когда выходной поток создается из нескольких файлов или из входного потока, то текущая дата и время появляются вместо имени файла и даты создания. Вы можете напечатать файлы параллельно в столбцах и управлять с помощью опций различными возможностями форматирования. Как обычно, смотрите man-страницы, чтобы узнать подробности.

Команда **nl** нумерует строки, что может быть полезно при печати файлов. Вы также можете нумеровать строки с помощью опции **-n** команды **cat**. На Листинге 31 показано как напечатать наш файл text 1, а затем как пронумеровать text2 и напечатать его параллельно с text1.

### Листинг 31. Нумерация и форматирование для печати

```
[ian@echidna lpi103]$ pr text1 | head
```

2005-09-23 12:27

text1

Page 1

```
1 apple
2 pear
3 banana
```

```
[ian@echidna lpi103]$ nl text2 | pr -m - text1 | head
```

2005-09-26 11:48

Page 1

1 9	plum	1 apple
2 3	banana	2 pear
3 10	apple	3 banana

Другой полезной командой форматирования текста является **fmt**, которая форматирует текст так, что он подходит по определенным размерам. Вы можете соединить несколько коротких строк, а также разделить на несколько строк. В Листинге 32 мы создаем файл text3 с одной большой строкой текста, используя возможности истории с помощью **!#:\***, чтобы не печатать наше предложение четыре раза. Мы также создадим файл text4, содержащий по одному слову на строке. Затем мы будем использовать **cat**, чтобы отобразить их в неформатированном виде, включая символ '\$' для обозначения конца строк. Наконец, мы используем **fmt** для форматирования их с максимальной шириной строкой в 60 символов. Снова, обращайтесь к man-страницам за более подробной информацией.

### Листинг 32. Форматирование по максимальной длине строки

```
[ian@echidna lpi103]$ echo "This is a sentence. " !#:* !#:1-$>text3
echo "This is a sentence. " "This is a sentence. " "This is a sentenc
e. " "This is a sentence. ">text3
[ian@echidna lpi103]$ echo -e "This\nis\nanother\nsentence.">text4
[ian@echidna lpi103]$ cat -et text3 text4
This is a sentence. This is a sentence. This is a sentence. This i
s a sentence. $
This$#
is$#
another$#
sentence.$#
[ian@echidna lpi103]$ fmt -w 60 text3 text4
This is a sentence. This is a sentence. This is a
sentence. This is a sentence.
```

This is another sentence.

## Sort и uniq

Команда **sort** сортирует ввод, согласно схеме локали (LC\_COLLATE) в системе. Команда **sort** также может соединять файлы и проверять, является ли файл отсортированным или нет.

Листинг 33 иллюстрирует применение команды **sort** для сортировки двух файлов, после того как мы преобразовали пробелы в символы табуляции в `text1`. Так как порядок сортировки происходит по символам, то результаты могут вас удивить. К счастью команда **sort** может сортировать как по числовым значениям, так и по символам. Порядок сортировки можно определить как для целой записи, так для каждого *поля*. Пока вы не укажите другой разделитель, поля будут разделяться пробелами и табуляторами. Второй пример в Листинге 33 показывает сортировку первого поля по цифрам, а второго поля по буквам с помощью схемы упорядочивания (в алфавитном порядке). Он также иллюстрирует применение опции **-u** для уничтожения любых дублируемых строк.

### Листинг 33. Сортировка по символам и числам

```
[ian@echidna lpi103]$ cat text1 | tr ' ' '\t' | sort - text2
10      apple
1      apple
2      pear
3      banana
3      banana
9      plum
[ian@echidna lpi103]$ cat text1|tr ' ' '\t'|sort -u -k1n -k2 - text2
1      apple
2      pear
3      banana
9      plum
10     apple
```

Заметим, что у нас все еще есть две строчки, содержащие фрукт "apple". Другая команда **uniq** дает нам дополнительный контроль над выявлением дублирующих строк. Команда **uniq** обычно работает с отсортированными файлами, но удаляет **последовательные** одинаковые строки из любого файла, отсортированного или нет. Команда **uniq** также может игнорировать некоторые поля. В Листинге 34 показана сортировка двух файлов по второму полю (имени фрукта), а затем уничтожение идентичных по второму полю строк.

### Листинг 34. Использование uniq

```
[ian@echidna lpi103]$ cat text1|tr ' ' '\t'|sort -k2 - text2|uniq -f1
10      apple
3      banana
2      pear
9      plum
```

Сортировка производилась согласно схеме упорядочивания, поэтому **uniq** выдало результат "10 apple", а не "1 apple". Попытайтесь добавить сортировку первого поля по числам, чтобы увидеть изменения.

## Cut, paste и join

Давайте рассмотрим еще три команды, которые работают с полями в текстовых данных. Особенно эти команды полезны при работе с табулированными данными. Первая команда это **cut**, которая извлекает поля из текста. По умолчанию символом разделителем является табулятор. Листинг 35 использует **cut** для разделения двух столбцов `text2`, а затем использует пробел как выходной разделитель, что является довольно специфичным способом преобразования символов табуляции в пробелы.

### Листинг 35. Использование cut

```
[ian@echidna lpi103]$ cut -f1-2 --output-delimiter=' ' text2
9 plum
3 banana
10 apple
```

Команда **paste** вставляет строки из двух или более файлов параллельно, подобно тому, как команда **pr** объединяет два файла с помощью опции **-m**. Листинг 36 демонстрирует результат вставки двух текстовых файлов.

### Листинг 36. Вставка файлов

```
[ian@echidna lpi103]$ paste text1 text2
1 apple 9      plum
2 pear   3      banana
3 banana     10    apple
```

Эти примеры показывают простую вставку, но **paste** может вставлять данные из одного или нескольких файлов различными способами. За подробностями обращайтесь к `man`-страницам.

Последняя команда манипулирования с полями это **join**, которая объединяет файлы на основе соответствия полей. Эти файлы должны быть отсортированы по объединяемому полю. Так как `text2` не отсортирован по числовому порядку, мы можем отсортировать его, а затем объединить две строки, которые имеют одинаковое поле (а данном случае 3). Давайте также создадим новый файл, `text5`, отсортировав `text1` по второму полю (имени фрукта), а затем заменив пробелы на символы табуляции. Если мы затем отсортируем `text2` и объединим его с `text5` по второму полю, то получим два совпадения (яблоко и банан). Листинг 37 иллюстрирует эти примеры.

### Листинг 37. Объединение файлов по полям

```
[ian@echidna lpi103]$ sort -n text2|join -1 1 -2 1 text1 -
3 banana banana
[ian@echidna lpi103]$ sort -k2 text1|tr ' ' '\t'>text5
[ian@echidna lpi103]$ sort -k2 text2 | join -1 2 -2 2 text5 -
apple 1 10
banana 3 3
```

Поле, используемое для объединения, указывается отдельно для каждого файла. Вы можете, например, объединить файл по полю 3 с файлом по полю 10.

## Sed

Sed это *streameditor* (потоковый редактор). Несколько статей developerWorks, а также много книг и глав посвящено sed (смотри [Ресурсы](#)). Sed чрезвычайно мощный инструмент, а задачи, которые можно выполнить с его помощью, ограничены лишь вашим воображением. Это небольшое введение должно пробудить у вас интерес к sed, но оно не является полным или расширенным.

Как и другие команды, которые мы рассмотрели, sed может работать как фильтр или получать ввод из файла. Вывод осуществляется на стандартный поток вывода. Sed загружает строки из ввода в *пространство шаблонов*, применяет команды редактирования sed к содержимому пространства шаблонов, а затем осуществляет на стандартный вывод пространства шаблонов. Sed может скомпоновать несколько строк в пространстве шаблонов, и может результат записать в файл, записать только частичный вывод, или же вообще ничего не записывать.

Sed использует синтаксис регулярных выражений (смотри [Поиск с помощью регулярных выражений](#) далее в этом руководстве) для поиска и избирательной замены текста в пространстве шаблонов, а также выбора тех строк, над которыми необходимо провести набор команд редактирования. Специальный *буфер* предоставляет временное хранилище для текста. Буфер может заменить пространство шаблонов, может быть добавлен к пространству шаблонов или же вести обмен с пространством шаблонов. Sed имеет ограниченный набор команд, но при использованию синтаксиса регулярных выражений и буфера может предоставлять потрясающие возможности. Набор команд sed обычно называется *sed-сценарием*.

В Листинге 38 представлено три простых sed-сценария. В первом мы используем команду **s** (замещения) буквы 'a' в нижнем регистре на верхний в каждой строке. Этот пример замещает только первые вхождения 'a', поэтому во втором примере мы добавили флаг 'g' (от глобальный) для замещения всех вхождений буквы. В третьем сценарии мы рассматриваем команду **d** (удалить) для удаления строки. В нашем примере мы используем адрес второй строки, чтобы показать, что только ее необходимо удалить. Мы разделяем команды точкой с запятой (;) а затем используем глобальное замещение, которое мы использовали во втором сценарии для замены 'a' на 'A'.

### Листинг 38. Начало работы с sed-сценариями

```
[ian@echidna lpi103]$ sed 's/a/A/' text1
1 Apple
2 peAr
3 bAnana
[ian@echidna lpi103]$ sed 's/a/A/g' text1
1 Apple
2 peAr
3 bAnAnA
[ian@echidna lpi103]$ sed '2d;$s/a/A/g' text1
1 apple
3 bAnAnA
```

В дополнение работе с одиночными строками, sed может работать с группой строк. Начало и конец диапазона разделяется запятой (,) и это может быть, как и номер строки, каретка (^), означающая начало файла, так и знак доллара (\$), означающий конец файла. Зная адрес или

диапазон адресов, вы можете сгруппировать несколько команд и заключить их в фигурные скобки ({ и }) так, что эти команды будут выполнены только на определенном диапазоне строк. Листинг 39 иллюстрирует два способа глобальной подстановки, применимой только к двум последним строкам файла. Он также иллюстрирует применение опции **-e** для исполнения команд в пространстве шаблонов. При использовании круглых скобок команды необходимо разделять запятыми.

### Листинг 39. Sed с использованием адресации

```
[ian@echidna lpi103]$ sed -e '2,${' -e 's/a/A/g' -e '}' text1
1 apple
2 peAr
3 bAnAnA
[ian@echidna lpi103]$ sed -e '/pear/,/bana/{' -e 's/a/A/g' -e '}' text1
1 apple
2 peAr
3 bAnAnA
```

Sed-сценарии могут также храниться в файлах. На самом деле вы захотите так сделать для часто используемых сценариев. Помните, раньше мы использовали команду **tr** для замены пробелов в text1 на символы табуляции. Давайте теперь сделаем это с помощью sed-сценария, хранящегося в файле. Мы будем использовать команду **echo** для создания файла. Результаты представлены в Листинге 40.

### Листинг 40. Использование sed-сценария

```
[ian@echidna lpi103]$ echo -e "s/ /\t/g">sedtab
[ian@echidna lpi103]$ cat sedtab
s/ /    /g
[ian@echidna lpi103]$ sed -f sedtab text1
1      apple
2      pear
3      banana
```

Существует множество таких удобных сценариев как в Листинге 40. Смотри [Ресурсы](#), чтобы узнать дополнительную информацию.

Наш последний пример sed использует команду **=** для вывода номеров строк, а затем фильтрации вывода опять через sed для имитации эффекта команды **nl** для нумерации строк. Листинг 41 использует **=** для вывода номеров строк, а затем использует команду **N** для помещения каждой второй строки в пространство шаблонов в продолжение первой, и наконец удаляет символ новой строки (\n) между двумя строками в пространстве шаблонов.

### Листинг 41. Нумерация строк в sed

```
[ian@echidna lpi103]$ sed '=' text2
1
9      plum
2
3      banana
3
10     apple
[ian@echidna lpi103]$ sed '=' text2|sed 'N;s/\n//'
```

```
19      plum
23      banana
310     apple
```

Не совсем то, что мы хотели! Мы хотели, чтобы нумерация была выровнена по столбцу, а также, чтобы номера и строки файла отделяли несколько пробелов. В Листинге 42 мы вводим несколько строк команд (заметим второе приглашение >). Изучите пример и посмотрите объяснение ниже.

#### Листинг 42. Нумерация строк с sed - второй подход

```
[ian@echidna lpi103]$ cat text1 text2 text1 text2>text6
[ian@echidna lpi103]$ ht=$(echo -en "\t")
[ian@echidna lpi103]$ sed '=' text6|sed "N
> s/^/    /
> s/^.*\(\.\.\.\.\)\n/\1$ht/"
1 1 apple
2 2 pear
3 3 banana
4 9  plum
5 3  banana
6 10 apple
7 1 apple
8 2 pear
9 3 banana
10 9  plum
11 3  banana
12 10 apple
```

Шаги, которые мы предприняли:

1. Мы использовали `cat`, чтобы создать файл из 12 строк с помощью двух копий `text1` и `text2`. Нет никакого интереса в форматировании чисел, если все они одного порядка.
2. Bash использует клавишу tab для автозавершения команды, поэтому полезно иметь уже заготовленный символ табуляции, чтобы использовать его, когда он действительно понадобится. Мы используем команду `echo`, чтобы вывести и сохранить его в shell переменной 'ht'.
3. Мы создаем поток, содержащий номера строк и данные как мы делали до этого и фильтруем его через вторую копию `sed`.
4. Мы вторые строки вместе с первыми в пространство шаблонов
5. Мы предваряем нашу строку с нумерацией в начале пространства шаблонов (обозначаемого '^') шестью пробелами.
6. Затем замещаем строку с пробелами и символом новой строки символом табуляции. Заметим, что левая часть команды 's' использует '(' и ')' для обозначения символов, которые мы хотим использовать в правой части. В правой части мы ссылаемся на первое такое найденное множество (и только такое в данном примере) как \1. Заметим, что наша команда находится между двойных кавычек ("), так что данная подстановка будет определена для \$ht.

Последняя версия (версия 4) `sed` содержит документацию в формате `info` и включает много превосходных примеров. Их нет в старой версии 3.02. GNU `sed` примет команду `sed --version` и отобразит свою версию.

## Простое управление файлами

Эта глава содержит материал темы 1.103.3 подготовки к экзамену Junior Level Administration (LPIC-1) 101. Тема имеет вес 3.

В этом разделе вы изучите следующие вопросы:

- Просмотр содержимого каталогов
- Копирование, перемещение и удаление файлов и каталогов
- Рекурсивное выполнение команд над файлами и каталогами
- Использование шаблонов для управления файлами
- Использование команды `find` для поиска файлов по типу, размеру или времени

### Просмотр каталогов

Как мы сказали ранее, при обсуждении путей в разделе [об использовании командной строки](#), все файлы в системах Linux и UNIX® являются частью большого дерева файловой системы, корнем которого является `/`.

### Просмотр записей каталогов

Если вы проработали предыдущий раздел, то создали каталог `lpi103` в своем домашнем каталоге. Имена файлов и каталогов могут быть как *абсолютными*, что значит, что они начинаются с `/` так и *относительными* к *текущему рабочему каталогу*, что значит, что они не начинаются с `/`. Абсолютное имя файла или каталога состоит из `/`, за которым следует последовательность из 0 или больше имен каталогов, каждый из которых отделяется `/`, а затем конечного имени. Если вы знаете относительное имя файла или каталога, то просто соедините абсолютное имя текущего рабочего каталога, `/` и относительное имя. Например, каталог `lpi103`, который мы создали в прошлом разделе, был создан в моем домашнем каталоге `/home/ian`, поэтому его полный или абсолютный путь `/home/ian/lpi103`. Листинг 43 показывает три разных способа использования команды `ls` для просмотра списка файлов каталога.

### Листинг 43. Просмотр записей каталога

```
[ian@echidna lpi103]$ pwd
/home/ian/lpi103
[ian@echidna lpi103]$ echo $PWD
/home/ian/lpi103
[ian@echidna lpi103]$ ls
sedtab  text2  text4  text6  xab  yab
text1   text3  text5  xaa   yaa
[ian@echidna lpi103]$ ls "$PWD"
sedtab  text2  text4  text6  xab  yab
text1   text3  text5  xaa   yaa
[ian@echidna lpi103]$ ls /home/ian/lpi103
sedtab  text2  text4  text6  xab  yab
text1   text3  text5  xaa   yaa
```

Как вы видите, можно передать имя каталога в качестве параметра команде `ls` и она выдаст содержимое этого каталога.

### Подробности о списке файлов

На устройстве хранения файл или каталог представляет собой коллекцию *блоков*. Информация о файле содержится в *inode*, который хранит информацию о владельце, последнем времени доступа, размере, файл это или каталог, права доступа. Номер inode

также известный как *последовательный номер файла* упомянут в переделах определенной файловой системы. Мы можем использовать опцию `-l` (или `--format=long`) для отображения некоторой информации, хранящейся в inode.

По умолчанию команда `ls` не выводит специальные файлы, чьи имена начинаются с точки (.). У каждого каталога, кроме корневого, есть две специальных записи -- это сам каталог (.) и родительский каталог (..). У корневого каталога отсутствует родительский каталог.

Листинг 44 использует опции `-l` и `-a` для отображения длинного формата списка всех файлов, включая записи. и .. .

#### Листинг 44. Расширенный формат вывода

```
[ian@echidna lpi103]$ ls -al
total 56
drwxrwxr-x    2 ian      ian          4096 Sep 30 15:01 .
drwxr-xr-x   94 ian      ian          8192 Sep 27 12:57 ..
-rw-rw-r--    1 ian      ian           8 Sep 26 15:24 sedtab
-rw-rw-r--    1 ian      ian          24 Sep 23 12:27 text1
-rw-rw-r--    1 ian      ian          25 Sep 23 13:39 text2
-rw-rw-r--    1 ian      ian          84 Sep 25 17:47 text3
-rw-rw-r--    1 ian      ian          26 Sep 25 22:28 text4
-rw-rw-r--    1 ian      ian          24 Sep 26 12:46 text5
-rw-rw-r--    1 ian      ian          98 Sep 26 16:09 text6
-rw-rw-r--    1 ian      ian          15 Sep 23 14:11 xaa
-rw-rw-r--    1 ian      ian           9 Sep 23 14:11 xab
-rw-rw-r--    1 ian      ian          18 Sep 23 14:11 yaa
-rw-rw-r--    1 ian      ian           7 Sep 23 14:11 yab
```

В Листинге 44, первая строка показывает общее число дисковых блоков (56), занимаемых выведенным на экран файлами. Оставшиеся поля расскажут о файле.

- Первое поле (drwxrwxr-x или -rw-rw-r-- в этом случае) говорит нам о том, является ли запись обычным файлом (-) или каталогом (d). Также вы можете увидеть символьные ссылки (l), о которых мы узнаем чуть позже или другие значения для специальных файлов (как например, для файлов в файловой системе /dev). За типом файла следует три набора прав доступа (как rwx или r--) для владельца, членов группы владельца и всех остальных. Три значения соответственно указывают пользователю, группе и всем остальным есть ли у них право на чтение (r), запись (w) или исполнение (x). Другие возможности использования как setuid будут рассказаны позже.
- Следующее числовое поле говорит нам число *жестких ссылок* на файл. Мы говорили, что inode содержит информацию о файле. Запись в каталоге о файле содержит жесткую ссылку (или указатель) на inode для этого файла, так, что у каждой записи в каталоге есть по крайней мере одна жесткая ссылка. У записей каталога есть дополнительная ссылка на запись . и по одной записи .. для каждого подкаталога. Из приведенного выше листинга видно, что у моего домашнего каталога есть несколько подкаталогов.
- Следующие два поля представляют владельца файла и группу, к которой он принадлежит. Некоторые системы, такие как Red Hat, по умолчанию каждому пользователю предоставляют свою группу. В других системах все пользователи могут быть в одной или нескольких группах.
- Следующее поле указывает размер файла.
- Предпоследнее поле указывает время последней модификации.
- Последнее поле содержит имя файла или каталога.

Опция `-i` команды `ls` отобразит для вас номера inode. Мы поговорим о них чуть позже, а также когда будем обсуждать жесткие и символьные ссылки в руководстве для темы 104.

## Множество файлов

Вы также можете определить несколько параметров команде `ls`, каждый из которых будет или именем файла или каталога. Если имя представляет собой каталог, то команда `ls` отобразит содержимое этого каталога, а не саму запись о каталоге. В нашем примере, предположим мы хотим получить информацию о каталоге `lpi103`. Команда `ls -l ./lpi103` отобразит нам список как в предыдущем примере. Листинг 45 покажет, как использовать `ls -ld` и как отобразить список записей для нескольких файлов и каталогов.

### Листинг 45. Использование ls -d

```
[ian@echidna lpi103]$ ls -ld ./lpi103 sedtab xaa
drwxrwxr-x    2 ian      ian          4096 Oct  2 18:49 ./lpi103
-rw-rw-r--    1 ian      ian           8 Sep 26 15:24 sedtab
-rw-rw-r--    1 ian      ian          15 Sep 23 14:11 xaa
```

Заметим, что время модификации `lpi103` отличается от того, чтобы было в предыдущем листинге. Также как и в предыдущем листинге, время доступа файлов и каталогов изменено. Вы этого ожидали? Думаю, нет. Однако при разработке этого руководства я создал несколько дополнительных примеров, а затем удалил их, поэтому дата доступа каталога отражает этот факт. Мы поговорим об этом чуть позже, когда будем обсуждать [поиск файлов](#).

## Сортировка вывода

По умолчанию команда `ls` выводит файлы в алфавитном порядке. Для сортировки вывода существует множество опций. Например, `ls -t` отсортирует по дате модификации (от самой последней до самой старой), в то время как `ls -lS` отсортирует список по размеру (от самых больших к маленьким). Добавление `-r` приводит к сортировке в обратном порядке. Например, используем `ls -lrt`, чтобы вывести длинный список, отсортированный в обратном порядке по дате модификации. За подробностями обратитесь к man-страницам.

## Копирование, перемещение и удаление

Мы научились создавать файлы, но предположим, что хотим сделать их копию или же переименовать их, или переместить в другое место файловой системы, или же удалить совсем. Для этих целей мы будем использовать три команды.

### cp

используется для копирования одного или более файлов. Вы **должны** передать ей по крайней мере два аргумента, один (или более) *источников*, а второй имя *цели*. Если вы определите два имени файла, то первый будет скопирован во второй. Как источник, так и цель могут включать в себя пути. Если вы определите каталог как последнее имя, то можете определить множество файлов, которое будет в него скопировано. Все файлы будут скопированы из существующих месторасположений, а копии будут иметь те же имена, что и исходные файлы. Заметим, что здесь нет никаких предположений по умолчанию о том, является ли цель текущим каталогом как в операционных системах DOS и Windows.

### mv

используется для *перемещения* или *переименования* одного или более файлов или каталогов. В общем случае имена, которые вы можете использовать, следуют тем же правилам, что и для копирования с помощью команды `Cp`; вы можете переименовать один файл или переместить набор файлов в новый каталог. Так как имя это только

запись в каталоге, которая связана с inode, то вас не должно удивлять, что значение inode не меняется **за исключением**, когда файл перемещается в другую файловую систему, в этом случае перемещение похоже на копирование, за которым следует удаление оригинала.

## rm

используется для **удаления** одного или нескольких файлов. Мы коротко рассмотрим, как удалять каталоги.

Листинг 46 иллюстрирует применение **cp** и **mv** для создания резервных копий текстовых файлов. Мы также используем **ls -i** чтобы показать inode некоторых файлов.

1. Сначала скопируем text1 в text1.bkp.
2. Затем создадим подкаталог с помощью команды **mkdir**
3. Затем создадим вторую копию text1, на этот раз в созданном каталоге и покажем, что все три файла имеют разные inode.
4. Затем мы перемещаем text1.bkp в созданный каталог и после этого переименовываем. Хотя мы могли бы сделать с помощью одной команды, мы для примера используем две.
5. Мы проверяем снова значения inode, чтобы подтвердить, что text1.bkp с inode 2129019 больше не находится в каталоге lpi103, но, что это inode файла text1.bkp.1 в каталоге для резервных копий.

## Листинг 46. Копирование и перемещение файлов

```
[ian@echidna lpi103]$ cp text1 text1.bkp
[ian@echidna lpi103]$ mkdir backup
[ian@echidna lpi103]$ cp text1 backup/text1.bkp.2
[ian@echidna lpi103]$ ls -i text1 text1.bkp backup
2128984 text1 2129019 text1.bkp

backup:
1564497 text1.bkp.2
[ian@echidna lpi103]$ mv text1.bkp backup
[ian@echidna lpi103]$ mv backup/text1.bkp backup/text1.bkp.1
[ian@echidna lpi103]$ ls -i text1 text1.bkp backup
ls: text1.bkp: No such file or directory
2128984 text1

backup:
2129019 text1.bkp.1 1564497 text1.bkp.2
```

Обычно **cp** перезапишет существующий файл, если он существует и в него можно писать. С другой стороны команда **mv** не переместит или переименует файл, если цель существует. Существует несколько опций, позволяющих менять поведение **cp** и **mv**.

### -f или --force

заставит cp попытаться удалить существующую цель, если в нее нельзя записывать.

### -i или --interactive

попросит интерактивно подтвердить попытку замещения существующего файла

### -b или --backup

сделает резервную копию файлов, которые будут замещены.

Как обычно обращайтесь к тан-страницам за более подробной информацией.

В Листинге 47 мы продемонстрируем копирование с резервированием, а затем удаление файлов.

### Листинг 47. Резервирование копий и удаление файлов

```
[ian@echidna lpi103]$ cp text2 backup
[ian@echidna lpi103]$ cp --backup=t text2 backup
[ian@echidna lpi103]$ ls backup
text1.bkp.1 text1.bkp.2 text2 text2.~1~
[ian@echidna lpi103]$ rm backup/text2
[ian@echidna lpi103]$ ls backup
text1.bkp.1 text1.bkp.2
```

Заметим, что команда `rm` также принимает опции `-i` и `-f`. Как только вы удалил файл с помощью `rm`, у файловой системы к нему больше нет доступа. На некоторых системах по умолчанию есть псевдоним `alias rm='rm -i'` для пользователя root, чтобы помочь предотвратить случайное удаление файла. Это также полезная мысль, если вы боитесь случайно что-либо удалить.

Прежде чем мы закончим обсуждение, следует сказать, что команда `cp` по умолчанию создает новую дату создания для новых файлов. Владелец и группа нового файла такие же, как и у оригинала. Опция `-p` может использоваться для сохранения выбранных атрибутов. Заметим, что только root может сохранять права владения. Смотри тап-страницы для получения подробной информации.

### Mkdir и rmdir

Мы уже видели, как создать каталог с помощью `mkdir`. Теперь пойдем дальше и рассмотрим команду для удаления каталогов `rmdir`.

### Mkdir

Положим, вы находитесь в каталоге lpi103 и хотите создать подкаталоги dir1 и dir2. Команда `mkdir`, как и другие рассмотренные команды, может создать несколько каталогов за один раз как показано в Листинге 48.

### Листинг 48. Создание нескольких каталогов

```
[ian@echidna lpi103]$ mkdir dir1 dir2
```

Команда не выводит подтверждения об успешном выполнении, но вы можете использовать `echo $?`, чтобы проверить, что код выхода действительно 0.

Если вы хотите создать вложенные каталоги, как например d1/d2/d3, то ничего не выйдет, так ни d1 ни d2 не существуют. К счастью у `mkdir` есть опция `-p`, которая позволяет создать любое число родительских каталогов. Листинг 49 иллюстрирует это.

### Листинг 49. Создание родительских каталогов

```
[ian@echidna lpi103]$ mkdir d1/d2/d3
mkdir: cannot create directory `d1/d2/d3': No such file or directory
[ian@echidna lpi103]$ echo $?
1
[ian@echidna lpi103]$ mkdir -p d1/d2/d3
[ian@echidna lpi103]$ echo $?
0
```

## Rmdir

Удаление каталогов происходит с помощью команды `rmdir`. Снова, существует опция `-p`, позволяющая удалять родительские каталоги. С помощью `rmdir` вы можете удалить каталоги, только если они пустые. Мы рассмотрим другой способ сделать это, когда будем рассматривать [рекурсивное манипулирование](#). Один раз, выучив как использовать команду `rmdir`, вы вряд ли будете часто применять ее в командной строке, но знать о ней, тем не менее, стоит.

Чтобы проиллюстрировать удаление каталогов, мы скопировали файл `text1` в каталог `d1/d2`, так что теперь он больше не пуст. Затем мы использовали `rmdir`, чтобы удалить все каталоги, которые мы создали с помощью `mkdir`. Как вы видите, `d1` и `d2` не удаляются, потому как `d2` не пуст. Другие же каталоги были удалены. Как только мы удалим копию `text1` из `d2`, мы сможем удалить `d1` и `d2` простой командой `rmdir -p`.

### Листинг 50. Удаление каталогов

```
[ian@echidna lpi103]$ cp text1 d1/d2
[ian@echidna lpi103]$ rmdir -p d1/d2/d3 dir1 dir2
rmdir: `d1/d2': Directory not empty
[ian@echidna lpi103]$ ls . d1/d2
.:
backup  sedtab  text2  text4  text6  xab  yab
d1      text1  text3  text5  xaa    yaa

d1/d2:
text1
[ian@echidna lpi103]$ rm d1/d2/text1
[ian@echidna lpi103]$ rmdir -p d1/d2
```

## Рекурсивное манипулирование

В оставшихся нескольких частях этого раздела мы рассмотрим различные операции обработки над множеством файлов, а также рекурсивном применении команд к дереву каталогов.

### Рекурсивный просмотр

У команды `ls` есть опция `-R` (заглавная буква 'R') для просмотра списка каталога и его подкаталогов. Опция рекурсии применима только к именам каталогов; она не будет искать все файлы, положим '`text1`', в дереве каталогов. Мы можем использовать другие опции, которые мы рассмотрели совместно с `-R`. Рекурсивный список каталога `lpi103`, включая значения `inode`, показан в Листинге 51.

### Листинг 51. Рекурсивный просмотр каталогов

```
[ian@echidna lpi103]$ ls -iR ~/lpi103
/home/ian/lpi103:
1564496 backup  2128985 text2  2128982 text5  2128987 xab
2128991 sedtab  2128990 text3  2128995 text6  2128988 yaa
2128984 text1  2128992 text4  2128986 xaa    2128989 yab

/home/ian/lpi103/backup:
2129019 text1.bkp.1  1564497 text1.bkp.2
```

## Рекурсивное копирование

Вы можете использовать опцию `-r` (или `-R` или `--recursive`), чтобы сообщить команде `cp` о прохождении по всему дереву каталога источника и рекурсивно скопировать его содержимое. Чтобы предотвратить бесконечную рекурсию, сам исходный каталог может быть не скопирован. Листинг 52 показывает, как скопировать все в нашем каталоге `lpi103` в подкаталог `copy1`. Мы используем `ls -R`, чтобы показать полученное дерево каталогов.

### Листинг 52. Рекурсивное копирование

```
[ian@echidna lpi103]$ cp -pR . copy1
cp: cannot copy a directory, '.', into itself, `copy1'
[ian@echidna lpi103]$ ls -R
.:
backup  sedtab  text2  text4  text6  xab  yab
copy1   text1   text3  text5  xaa    yaa

./backup:
text1.bkp.1  text1.bkp.2

./copy1:
backup  text1  text3  text5  xaa  yaa
sedtab  text2  text4  text6  xab  yab

./copy1/backup:
text1.bkp.1  text1.bkp.2
```

## Рекурсивное удаление

Мы упомянули ранее, что команда `rmdir` удаляет только пустые каталоги. Мы можем использовать опцию `-r` (или `-R` или `--recursive`), чтобы заставить команду `rm` удалить как файлы так и каталоги, как показано в Листинге 53, где мы удаляем каталог `copy1`, который только что создали, а также его содержимое, включая каталог с резервными копиями и его содержимое.

### Листинг 53. Рекурсивное удаление

```
[ian@echidna lpi103]$ rm -r copy1
[ian@echidna lpi103]$ ls -R
.:
backup  text1  text3  text5  xaa  yaa
sedtab  text2  text4  text6  xab  yab

./backup:
text1.bkp.1  text1.bkp.2
```

Если файлы не доступны вам для записи, то вы можете добавить опцию `-f`, чтобы принудительно удалить их. Так часто поступает пользователь `root`, при чистке системы, но будьте внимательны, так как можете потерять ценные данные, если будете невнимательны.

## Шаблоны и подстановки

Часто вам необходимо произвести единую операцию над множеством объектов файловой системы, не оперируя с деревом, как мы поступали с рекурсивными операциями. Например,

вы захотите найти время модификации всех текстовых файлов, которые мы создали в каталоге lpi103, не выводя разделенных файлов. Хотя это легко сделать в нашем небольшом каталоге, это задача становится непосильной в большой файловой системе.

Чтобы решить эту проблему, используйте поддержку шаблонов, которая встроена в bash. Эта поддержка, также называемая "globbing" (потому что изначально она была реализована программой /etc/glob), позволит вам определить множество файлов с помощью шаблона.

Строка, содержащая любой из символов '?', '\*' или '[', является *шаблоном*. Подстановка это процесс, в котором интерпретатор (или возможно другая программа) заменяет эти шаблоны на список путей, соответствующих шаблону. Соответствие осуществляется следующим образом.

?

означает один любой символ

\*

соответствует любой строке, включая пустую строку.

[

представляет *класс символов*. Класс символов это непустая строка, завершенная ']'. Соответствие означает совпадение с любым символом, заключенным в скобках. Есть несколько особых случаев.

- Символы '\*' и '?' означают сами себя. Если вы используете их в именах файлах, то вам следует заботиться об использовании кавычек или escape-последовательностей.
- Так как строка должна быть непустой, и завершается знаком ']', то вы должны помещать символ ']' **первым** в строке, если хотите найти его соответствие.
- Символ '-' между двумя другими означает диапазон, который включает как эти два символа, так и все те, что находятся между ними. Например, [0-9a-fA-F] представляет собой множество шестнадцатеричных цифр, написанных в верхнем или нижнем регистрах. Чтобы найти соответствие для '-', вы можете поместить его первым или последним в диапазоне.
- Символ '!' означает первый символ диапазона, который дополняет диапазон таким образом, что они не пересекаются. Например, [!0-9] означает любой символ кроме чисел от 0 до 9. Символ '!' соответствует себе, если он не стоит в первой позиции. Помните, что '!' также используется в истории интерпретатора, поэтому будьте внимательны.

Подстановка применяется отдельно к каждому компоненту имени пути. Вы не можете использовать '/' для совпадения или включения его в диапазон. Вы можете использовать его в любом месте, чтобы определить множество файлов или имен каталогов, например, в командах `ls`, `cp`, `mv` или `rm`. В Листинге 54, мы сначала создаем несколько файлов со странными именами, а затем используем команды `ls` и `rm` с шаблонами.

#### Листинг 54. Примеры использования шаблонов

```
[ian@echidna lpi103]$ echo odd1>'text[*?!1]'
[ian@echidna lpi103]$ echo odd2>'text[2*?!]'
[ian@echidna lpi103]$ ls
backup  text1      text2      text3  text5  xaa  yaa
sedtab  text[*?!1]  text[2*?!]  text4  text6  xab  yab
[ian@echidna lpi103]$ ls text[2-4]
text2  text3  text4
[ian@echidna lpi103]$ ls text[!2-4]
text1  text5  text6
[ian@echidna lpi103]$ ls text*[2-4]*
```

```

text2 text[2*?!] text3 text4
[ian@echidna lpi103]$ ls text*[^2-4]* # Surprise!
text1 text[*?!] text[2*?!] text5 text6
[ian@echidna lpi103]$ ls text*[^2-4] # More surprise!
text1 text[*?!] text[2*?!] text5 text6
[ian@echidna lpi103]$ echo text*>text10
[ian@echidna lpi103]$ ls *\!*
text[*?!] text[2*?!]
[ian@echidna lpi103]$ ls *[x\!]*
text1 text2 text3 text5 xaa
text[*?!] text[2*?!] text4 text6 xab
[ian@echidna lpi103]$ ls *[y\!]*
text[*?!] text[2*?!] yaa yab
[ian@echidna lpi103]$ ls tex?[]*
text[*?!] text[2*?!]
[ian@echidna lpi103]$ rm tex?[]*
[ian@echidna lpi103]$ ls *b*
sedtab xab yab

backup:
text1.bkp.1 text1.bkp.2
[ian@echidna lpi103]$ ls backup/*2
backup/text1.bkp.2
[ian@echidna lpi103]$ ls -d .*
...

```

Примечания:

1. Дополнение совместно с '\*' может привести к неожиданным сюрпризам. Шаблон '\*[!2-4]' соответствует длиннейшей части имени, за которым не следуют символы 2, 3 или 4, что соответствует **как** `text[*?!]` так и `text[2*?!]`. Теперь оба сюрприза станут понятны.
2. Что касается ранних примеров `ls`, то если подстановка является именем каталога, а опция `-d` не указана, тогда будет выдано содержимое этого каталога (как в примере выше для шаблона `*b*`).
3. Если имя файла начинается с точки (.), то необходимо явно его указать для поиска соответствия. Заметим, что только последняя команда `ls` отобразила две специальных записи (.) и (..).

Помните, что любая последовательность шаблонных символов может быть интерпретирована интерпретатором и привести к неожиданным результатам. Более того, если вы определите шаблон, который не соответствует никаким объектам в файловой системе, то POSIX требует, чтобы этот шаблон был передан в команду. Иллюстрируем это в Листинге 55. Некоторые ранние версии передавали пустой список в команду, поэтому вы можете столкнуться со старыми сценариями, которые ведут себя необычно. Проиллюстрируем это в Листинге 55.

### Листинг 55. Сюрпризы при использовании шаблонов

```

[ian@echidna lpi103]$ echo text*
text1 text2 text3 text4 text5 text6
[ian@echidna lpi103]$ echo "text*"
text*
[ian@echidna lpi103]$ echo text[!\?]z??
text[!\?]z??

```

Более подробно смотрите в [man 7 glob](#). Вам требуется именно этот раздел, так как есть информация о `glob` в разделе 3. Лучший способ понять теорию это практика, поэтому используйте при любом случае шаблоны. Не забывайте проверять с помощью `ls`, что шаблон делает то, что нужно, до того как применять `cp`, `mv` или `rm` и не получать неожиданных результатов.

## Использование `touch`

Сейчас рассмотрим команду `touch`, которая может изменить время доступа и модификации или же создать пустой файл. В следующей части мы рассмотрим, как использовать эту информацию для поиска файлов и каталогов. Мы будем использовать каталог `lpi103`, который создали ранее в этом руководстве.

### `touch`

Команда `touch` без опций принимает один или более файлов как параметры и изменяет дату **модификации** файлов. Обычно это то время, которое отображается при длинном формате вывода списка каталогов. В Листинге 56 мы используем старого друга `echo` для создания простого файла `f1`, а затем используем длинный формат вывода списка каталога для отображения времени модификации (или *mtime*). В этом случае текущее время и будет временем создания файла. Затем мы используем команду `sleep`, чтобы подождать 60 секунд и снова запустим `ls`. Заметим, что время у файла изменилось на минуту.

### Листинг 56. Изменение времени модификации с помощью `touch`

```
[ian@echidna lpi103]$ echo xxx>f1; ls -l f1; sleep 60; touch f1; ls -l f1
-rw-rw-r-- 1 ian      ian          4 Nov  4 15:57 f1
-rw-rw-r-- 1 ian      ian          4 Nov  4 15:58 f1
```

Если вы определите имя файла, которого не существует, то `touch` просто создаст пустой файл, кроме случая, если вы не определите опцию `-C` или `--no-create`. Листинг 57 демонстрирует обе этих команды. Заметим, что создается только `f2`.

### Листинг 57. Создание пустых файлов с помощью `touch`

```
[ian@echidna lpi103]$ touch f2; touch -c f3; ls -l f*
-rw-rw-r-- 1 ian      ian          4 Nov  4 15:58 f1
-rw-rw-r-- 1 ian      ian          0 Nov  4 16:12 f2
```

Команда `touch` может также изменить *mtime* файла на определенную дату и время с помощью опций `-d` или `-t`. Опция `-d` очень гибка в плане понимания различных форматов даты и времени, в то время как опция `-t` требует по крайней мере формата MMDDhhmm времени, а также опционального указания значений года и секунд. Листинг 58 содержит такие примеры.

### Листинг 58. Установка *mtime* с помощью `touch`

```
[ian@echidna lpi103]$ touch -t 200511051510.59 f3
[ian@echidna lpi103]$ touch -d 11am f4
[ian@echidna lpi103]$ touch -d "last fortnight" f5
[ian@echidna lpi103]$ touch -d "yesterday 6am" f6
[ian@echidna lpi103]$ touch -d "2 days ago 12:00" f7
```

```
[ian@echidna lpi103]$ touch -d "tomorrow 02:00" f8
[ian@echidna lpi103]$ touch -d "5 Nov" f9
[ian@echidna lpi103]$ ls -lrt f*
-rw-rw-r-- 1 ian ian 0 Oct 24 12:32 f5
-rw-rw-r-- 1 ian ian 4 Nov 4 15:58 f1
-rw-rw-r-- 1 ian ian 0 Nov 4 16:12 f2
-rw-rw-r-- 1 ian ian 0 Nov 5 00:00 f9
-rw-rw-r-- 1 ian ian 0 Nov 5 12:00 f7
-rw-rw-r-- 1 ian ian 0 Nov 5 15:10 f3
-rw-rw-r-- 1 ian ian 0 Nov 6 06:00 f6
-rw-rw-r-- 1 ian ian 0 Nov 7 11:00 f4
-rw-rw-r-- 1 ian ian 0 Nov 8 2005 f8
```

Если вы затрудняетесь с определением даты, то можете использовать команду [date](#). У нее также есть опция [-d](#) которая понимает те же форматы даты, что и [touch](#).

Вы можете использовать опцию [-r](#) (или [--reference](#)) вместе с *именем файла*, чтобы сигнализировать, что [touch](#) (или [date](#)) следует использовать дату модификации существующего файла. В Листинге 59 содержатся некоторые примеры.

### Листинг 59. Использование даты модификации существующего файла

```
[ian@echidna lpi103]$ date
Mon Nov 7 12:40:11 EST 2005
[ian@echidna lpi103]$ date -r f1
Fri Nov 4 15:58:27 EST 2005
[ian@echidna lpi103]$ touch -r f1 f1a
[ian@echidna lpi103]$ ls -l f1*
-rw-rw-r-- 1 ian ian 4 Nov 4 15:58 f1
-rw-rw-r-- 1 ian ian 0 Nov 4 15:58 f1a
```

Система Linux записывает как время *модификации* файла, так и время *доступа*. Обе отметки совпадают по значению, когда файл создается, и обе сбрасываются, когда он изменяется. Время доступа изменяется, даже если файл не модифицировался. В нашем последнем примере с [touch](#), мы рассмотрим время *доступа*. Опция [-a](#) (или [--time=atime](#), [--time=access](#) или [--time=use](#)) определяет, что время доступа должно быть изменено. Листинг 60 использует команду [cat](#), чтобы прочесть файл f1 и отобразить его содержимое. Затем мы используем [ls -l](#) и [ls -lu](#), чтобы отобразить время модификации и доступа соответственно для файлов f1 и f1a, который мы создали, используя f1 как ссылку. Затем мы сбрасываем время доступа f1 до f1a, используя [touch -a](#).

### Листинг 60. Время доступа и модификации

```
[ian@echidna lpi103]$ cat f1
xxx
[ian@echidna lpi103]$ ls -lu f1*
-rw-rw-r-- 1 ian ian 4 Nov 7 14:13 f1
-rw-rw-r-- 1 ian ian 0 Nov 4 15:58 f1a
[ian@echidna lpi103]$ ls -l f1*
-rw-rw-r-- 1 ian ian 4 Nov 4 15:58 f1
-rw-rw-r-- 1 ian ian 0 Nov 4 15:58 f1a
[ian@echidna lpi103]$ cat f1
xxx
[ian@echidna lpi103]$ ls -l f1*
```

```

-rw-rw-r--    1 ian      ian          4 Nov  4 15:58 f1
-rw-rw-r--    1 ian      ian          0 Nov  4 15:58 f1a
[ian@echidna lpi103]$ ls -lu f1*
-rw-rw-r--    1 ian      ian          4 Nov  7 14:13 f1
-rw-rw-r--    1 ian      ian          0 Nov  4 15:58 f1a
[ian@echidna lpi103]$ touch -a -r f1a f1
[ian@echidna lpi103]$ ls -lu f1*
-rw-rw-r--    1 ian      ian          4 Nov  4 15:58 f1
-rw-rw-r--    1 ian      ian          0 Nov  4 15:58 f1a

```

За более полной информацией о принимаемых форматах времени и даты, смотри `man` или `info` страницы команд `touch` и `date`.

## Поиск файлов

Последней темой этой части руководства будет команда `find`, которая используется для поиска файлов в одном или более дереве каталогов, на основе таких признаков как имя, дата модификации или размер. Снова, мы будем использовать каталог `lpi103`, который создали ранее в этом руководстве.

### `find`

Команда `find` осуществляет поиск файлов или каталогов, используя все имя или его часть, или же другие критерии как размер, тип, владелец файла, дата создания или дата последнего доступа. Простейший поиск осуществляется по имени или его части. Листинг 61 демонстрирует каталог `lpi103`, в котором мы сначала ищем все файлы, имеющие в своем имени букву '1' или 'k', а затем производим поиск пути, который мы разъясним чуть ниже.

### Листинг 61. Поиск файлов по имени

```

[ian@echidna lpi103]$ find . -name "*[1k]*"
./text1
./f1
./backup
./backup/text1.bkp.2
./backup/text1.bkp.1
./f1a
[ian@echidna lpi103]$ find . -ipath "*ACK*1"
./backup/text1.bkp.1
[ian@echidna lpi103]$ find . -ipath "*ACK*/*1"
./backup/text1.bkp.1

```

### Замечания:

- Шаблоны, которые вы можете использовать такие же, как мы видели в разделе ранее [Шаблоны и подстановки](#).
- Вы можете использовать `-path` вместо `-name`, чтобы находить полные пути, а не просто имена файлов. В этом случае шаблон **может** изменять компоненты пути.
- Если вам нужен регистронезависимый поиск, как в примере с использованием `ipath`, предваряйте опции `find`, которые осуществляют по строке или шаблону, символом 'i'
- Если вы хотите найти файл или каталог, чье имя начинается с точки, как например `.bashrc` или текущий каталог `(.)`, то тогда вы **должны** определить ведущей точку, как часть шаблона. В противном случае поиск будет игнорировать эти файлы и каталоги.

В первом примере выше мы искали как файлы, так и каталоги (`./backup`). Используйте

параметр **-type**, а также однобуквенный тип, чтобы ограничить поиск. Используйте 'f' для регулярных файлов, 'd' для каталогов и 'l' для символьных ссылок. Смотри тап-страницу **find**, чтобы узнать о других типах. На Листинге 62 представлен результат поиска в каталогах (**-type d**).

### Листинг 62. Поиск файлов по типу

```
[ian@echidna lpi103]$ find . -type d  
./  
./backup  
[ian@echidna lpi103]$ find . -type d -name "*"  
./backup
```

Заметим, что **-type d**, без указания какой-либо спецификации отобразит список каталогов, у которых в имени ведущая точка (только текущий каталог в этом случае).

Мы также можем искать файлы по размеру, как определенному (n) так и файлы, размер которых больше (+n) или меньше (-n) определенного значения. Используя как верхнюю, так и нижнюю границы, можно найти файлы, чьи размеры попадают в этот диапазон. По умолчанию опция **-size** команды **find** предполагает файл из 'b' 512-байт блоков. Среди других, можно выбрать 'c' для байт, 'k' для килобайт. В Листинге 63 мы ищем сначала все файлы размера 0, а затем все файлы, чей размер от 24 до 25 байт. Заметим, что определение **-empty** вместо **-size 0** также заставляет искать пустые файлы.

### Листинг 63. Поиск файлов по размеру

```
[ian@echidna lpi103]$ find . -size 0  
./f2  
./f3  
./f4  
./f5  
./f6  
./f7  
./f8  
./f9  
./f1a  
[ian@echidna lpi103]$ find . -size -26c -size +23c -print  
./text1  
./text2  
./text5  
./backup/text1.bkp.2  
./backup/text1.bkp.1
```

В Листинге 63 рассматривается опция **-print**, которая является примером *действия*, которое может быть выполнено на результатом поиска. В bash shell это действие выполняется по умолчанию, если не определено иное. В некоторых системах и интерпретаторах действие обязательно, в противном случае вывода не будет.

Другие действия включают в себя **-ls**, которое печатает информацию о файле, аналогичную команде **ls -lids** или **-exec**, которое выполняет команду для каждого файла. Действие **-exec** должно заканчиваться точкой с запятой в виде escape-последовательности. Также используйте {}, если вы хотите, чтобы возвращаемый файл использовался в команде. Как мы видели раньше, фигурные скобки также интерпретируются, поэтому их надо либо заключать

в кавычки, либо писать как escape-последовательность. Листинг 64 показывает как опции **-ls** и **-exec** могут использоваться для выдачи информации о файлах.

#### Листинг 64. Поиск файлов и действия над ними

```
[ian@echidna lpi103]$ find . -size -26c -size +23c -ls  
2128984 4 -rw-rw-r-- 1 ian ian 24 Sep 23 12:27 ./text1  
2128985 4 -rw-rw-r-- 1 ian ian 25 Sep 23 13:39 ./text2  
2128982 4 -rw-rw-r-- 1 ian ian 24 Sep 26 12:46 ./text5  
1564497 4 -rw-rw-r-- 1 ian ian 24 Oct 4 09:45 ./backup/text1.bkp.2  
2129019 4 -rw-rw-r-- 1 ian ian 24 Oct 4 09:43 ./backup/text1.bkp.1  
[ian@echidna lpi103]$ find . -size -26c -size +23c -exec ls -l '{}' '\;  
-rw-rw-r-- 1 ian ian 24 Sep 23 12:27 ./text1  
-rw-rw-r-- 1 ian ian 25 Sep 23 13:39 ./text2  
-rw-rw-r-- 1 ian ian 24 Sep 26 12:46 ./text5  
-rw-rw-r-- 1 ian ian 24 Oct 4 09:45 ./backup/text1.bkp.2  
-rw-rw-r-- 1 ian ian 24 Oct 4 09:43 ./backup/text1.bkp.1
```

Опция **-exec** используется для самых различных задач, ограниченных вашей фантазией. Например:

```
find . -empty -exec rm '{}' '\;
```

удаляет все пустые файлы в дереве каталогов, в то время как

```
find . -name "*.htm" -exec mv '{}' '{}.html' '\;
```

переименует все файлы .htm в .html.

В наших последних примерах мы используем отметки времени модификации, описанные в команде **touch**, чтобы найти все файлы с определенной датой модификации. В Листинге 65 показаны три примера:

1. При использовании **-mtime -2** команда **find** ищет все файлы, которые были модифицированы в течение последних двух дней. День в данном случае это 24 часовой период, отсчитывающийся от текущей даты и времени. Заметим, что вам следует использовать **-atime**, если вы хотите осуществить поиск файлов на основе времени доступа, а не времени модификации.
2. Указание опции **-daystart** гарантирует, что мы хотим рассматривать календарные дни, начало которых отсчитывается от полуночи. В этом случае файл f3 в результат не попадает.
3. Наконец, мы покажем, как использовать диапазон в минутах, а не днях, чтобы найти файлы, модифицированные между одним часом (60 минут) и 10 часами (600 минут) ранее.

#### Листинг 65. Поиск файлов по временным отметкам

```
[ian@echidna lpi103]$ find . -mtime -2 -type f -exec ls -l '{}' '\;  
-rw-rw-r-- 1 ian ian 0 Nov 5 15:10 ./f3  
-rw-rw-r-- 1 ian ian 0 Nov 7 11:00 ./f4  
-rw-rw-r-- 1 ian ian 0 Nov 6 06:00 ./f6  
-rw-rw-r-- 1 ian ian 0 Nov 8 2005 ./f8  
[ian@echidna lpi103]$ find . -daystart -mtime -2 -type f -exec ls -l '{}' '\;  
-rw-rw-r-- 1 ian ian 0 Nov 7 11:00 ./f4
```

```
-rw-rw-r--    1 ian      ian          0 Nov  6 06:00 ./f6
-rw-rw-r--    1 ian      ian          0 Nov  8 2005 ./f8
[ian@echidna lpi103]$ find . -mmin -600 -mmin +60 -type f -exec ls -l '{}' \;
-rw-rw-r--    1 ian      ian          0 Nov  7 11:00 ./f4
```

Man-страницы команды **find** помогут вам изучить расширенный набор опций, который мы не можем рассмотреть в этом кратком введении.

| [предыдущая](#) | [следующая](#)

## Потоки, программные каналы и перенаправления

Этот раздел описывает материал темы 1.103.4 для подготовки к экзамену Junior Level Administration (LPIC-1) 101. Тема имеет вес 5.

В этом разделе вы изучите следующие темы:

- Перенаправление стандартных потоков ввода/вывода: стандартный ввод, стандартный вывод и стандартный поток ошибок
- Соединение вывода одной команды с входом другой
- Выдача вывода как на `stdout`, так и в файл
- Использование вывода одной команды как аргумент для другой команды

Перенаправление стандартного ввода/вывода

Напомним, что интерпретатор работает с тремя стандартными *потоками*.

1. *stdout* это *стандартный поток вывода*, который обеспечивает вывод команды. Его дескриптор равен 1.
2. *stderr* это *стандартный поток ошибок*, который выводит ошибки команд. Его дескриптор равен 2.
3. *stdin* это *стандартный поток ввода*, который обеспечивает ввод командам. Его дескриптор равен 0.

Входные потоки обеспечивают ввод командам, который обычно поступает от клавиш терминала. Потоки вывода печатают символы текста, обычно на терминал. Изначально терминал представлял собой печатную машинку ASCII или же дисплейный терминал, сейчас в основном это окно на рабочем столе графической среды.

В разделе [Текстовые потоки и фильтры](#) мы видели, как перенаправлять стандартный вывод в файл или на стандартный ввод другой команды, а также можем перенаправить стандартный ввод из файла или из вывода другой команды.

### Перенаправление вывода

Существует два способа перенаправить вывод:

**n>**

перенаправляет вывод из файлового дескриптора *n* в файл. У вас должно быть право записи в файл. Если файла не существует, то он будет создан. Если файл существует, его содержимое будет утеряно без предупреждения.

**n>>**

также перенаправляет вывод из файлового дескриптора *n* в файл. Снова, у вас должно быть право на запись в файл. Если файл не существует, то будет создан. Если он существует, то вывод команды добавится к существующему файлу.

Символы *n>* или *n>>* являются *дескрипторами файла*. Если его не написать, то по

умолчанию предполагается стандартный вывод. В Листинге 66 приведено перенаправление стандартного вывода и стандартного потока ошибок команды `ls`, используя файлы, созданные ранее в каталоге lpi103. Мы также продемонстрируем добавление вывода в существующий файл.

### Листинг 66. Перенаправление вывода

```
[ian@echidna lpi103]$ ls x* z*
ls: z*: No such file or directory
xaa xab
[ian@echidna lpi103]$ ls x* z* >stdout.txt 2>stderr.txt
[ian@echidna lpi103]$ ls w* y*
ls: w*: No such file or directory
yaa yab
[ian@echidna lpi103]$ ls w* y* >>stdout.txt 2>>stderr.txt
[ian@echidna lpi103]$ cat stdout.txt
xaa
xab
yaa
yab
[ian@echidna lpi103]$ cat stderr.txt
ls: z*: No such file or directory
ls: w*: No such file or directory
```

Мы сказали, что перенаправление с помощью `n>` обычно переписывает существующий файл. Вы можете контролировать это с помощью опции `noclobber` встроенной команды `set`. Если она определена, то вы можете ее подавить с помощью опции `n>|` как показано в Листинге 67.

### Листинг 67. Перенаправление вывода с помощью `noclobber`

```
[ian@echidna lpi103]$ set -o noclobber
[ian@echidna lpi103]$ ls x* z* >stdout.txt 2>stderr.txt
-bash: stdout.txt: cannot overwrite existing file
[ian@echidna lpi103]$ ls x* z* >|stdout.txt 2>|stderr.txt
[ian@echidna lpi103]$ cat stdout.txt
xaa
xab
[ian@echidna lpi103]$ cat stderr.txt
ls: z*: No such file or directory
[ian@echidna lpi103]$ set +o noclobber #восстановлением
изначальное значение параметра noclobber
```

Иногда вам требуется перенаправить как стандартный вывод так поток ошибок в файл. Это часто делается для автоматизированных процессов или фоновых задач, так что вы можете рассмотреть вывод позже. Используйте `&>` или `&>>`, чтобы перенаправить стандартный вывод и поток ошибок в одно и тоже место. Другой способ состоит в перенаправлении файлового дескриптора *n*, а затем перенаправлении файлового дескриптора *m* в одно и тоже место с помощью конструкции *m>&n* или *m>>&n*. Важен порядок, в котором осуществляется перенаправление вывода. Например,  
`command 2>&1 >output.txt`  
не тоже самое, что  
`command >output.txt 2>&1`

Проиллюстрируем эти концепции в Листинге 68. Заметим, что в последней команде стандартный вывод был перенаправлен после потока ошибок, таким образом стандартный вывод все еще выводится на терминал.

### Листинг 68. Перенаправление двух потоков в один файл

```
[ian@echidna lpi103]$ ls x* z* &>output.txt
[ian@echidna lpi103]$ cat output.txt
ls: z*: No such file or directory
xaa
xab
[ian@echidna lpi103]$ ls x* z* >output.txt 2>&1
[ian@echidna lpi103]$ cat output.txt
ls: z*: No such file or directory
xaa
xab
[ian@echidna lpi103]$ ls x* z* 2>&1 >output.txt
ls: z*: No such file or directory
[ian@echidna lpi103]$ cat output.txt
xaa
xab
```

В другие разы вам потребуется проигнорировать полностью стандартный вывод или поток ошибок. В этом случае перенаправьте нужный поток в /dev/null. В Листинге 69 мы покажем как игнорировать поток ошибок команды `ls`.

### Листинг 69. Игнорирование вывода с помощью /dev/null

```
[ian@echidna lpi103]$ ls x* z* 2>/dev/null
xaa  xab
[ian@echidna lpi103]$ cat /dev/null
```

## Перенаправление ввода

Также как мы перенаправляли потоки `stdout` и `stderr`, вы тоже можете перенаправлять `stdin` из файла, используя оператор `<`. Если вы вспомните в обсуждении [sort и uniq](#) мы использовали команду `tr` для замены пробелов в файле `text1` на символы табуляции. В том примере мы использовали вывод команды `cat`, чтобы создать стандартный ввод для команды `tr`. Вместо бессмысленного вызова `cat`, мы можем использовать перенаправление ввода для преобразования пробелов символы табуляции, как показано в Листинге 70.

### Листинг 70. Перенаправление ввода

```
[ian@echidna lpi103]$ tr ' ' '\t'<text1
1      apple
2      pear
3      banana
```

У интерпретаторов, включая `bash`, также есть концепция *документа*, которая является другой формой перенаправления ввода. Она включает в себя использование `<<` и слова, такого как `END`, в качестве маркера или сигнала конца ввода. Проиллюстрируем это на примере Листинга 71.

## Листинг 71. Перенаправление ввода, используя концепцию документа

```
[ian@echidna lpi103]$ sort -k2 <<END
> 1 apple
> 2 pear
> 3 banana
> END
1 apple
3 banana
2 pear
```

Помните, как мы создали файл `text2` в [Листинге 23](#)? Вы можете удивиться, почему бы просто не набрать `sort -k2`, ввести свои данные, а затем нажать `Ctrl-d`, чтобы сигнализировать окончание ввода. Короткий ответ состоит в том, что вы можете, но в таком случае вы не узнали бы о концепции документа. В действительности, документы очень часто используются в сценариях (которые рассмотрены в руководстве по теме 109 об интерпретаторах, сценариях, программировании и компиляции). В сценарии нет другого способа сигнализировать о том, какие строки необходимо воспринимать как ввод. Так как сценарии интенсивно используют табуляцию для выравнивания информации, то существует другой прием работы с документами. Если вы используете `<<` вместо `<<`, тогда ведущие символы табуляции будут удалены. В Листинге 72 мы использовали подобную технику для создания символа табуляции, который затем использовали в [Листинге 42](#). Затем мы создадим очень маленький сценарий, содержащий две команды `cat` каждый из которых будет считывать из документа. Наконец, мы используем команду `.` (точку), чтобы *обнаружить* сценарий в текущем каталоге и запустить его.

## Листинг 72. Перенаправление ввода с помощью документа

```
[ian@echidna lpi103]$ ht=$(echo -en "\t")
[ian@echidna lpi103]$ cat<<EOF>ex-here.sh
> cat <<-EOF
> apple
> EOF
> ${ht}cat <<-EOF
> ${ht}pear
> ${ht}EOF
> END
[ian@echidna lpi103]$ cat ex-here.sh
cat <<-EOF
apple
EOF
cat <<-EOF
pear
EOF
[ian@echidna lpi103]$ . ex-here.sh
apple
pear
```

## Конвейеры

В разделе [Текстовые потоки и фильтры](#) мы обсуждали *фильтрацию* текста, как процесс взятия входного потока, совершения преобразования над текстом и отсылки его в выходной поток. Мы также сказали, что фильтрация часто осуществляется с помощью *конвейера* команд, в котором выход одной команды *соединяется* или *перенаправляется* на вход другой

команды. Подобное использование конвейеров не ограничивается только текстовыми потоками, хотя именно для они используются чаще всего.

### Соединение stdout с stdin

Как мы уже видели, мы используем оператор | (конвейера) между двумя командами для перенаправления stdout первой команды на stdin второй команды. Мы конструируем длинные конвейеры из нескольких команд в Листинге 73.

### Листинг 73. Конвейер из нескольких команд

```
command1 | command2 | command3
```

Следует заметить, что конвейеры перенаправляют **только** stdout на stdin. Вы не можете использовать 2|, чтобы перенаправить stderr, по крайней мере, используя наши сейчас знания. Если stderr был перенаправлен на stdout, тогда оба потока будут соединены. Иллюстрируем этот подход в Листинге 74, в котором используем конвейер для сортировки сообщений об ошибках и нормальных сообщений команды [ls](#) с четырьмя шаблонами, расположенными не по алфавиту.

### Листинг 74. Конвейер из двух выходных потоков

```
[ian@echidna lpi103]$ ls y* x* z* u* q* 2>&1 |sort
ls: q*: No such file or directory
ls: u*: No such file or directory
ls: z*: No such file or directory
xaa
xab
yaa
yab
```

У любой из команд могут быть опции или аргументы. Многие команды используют дефис (-) вместо имени файла как аргумент, чтобы сообщить, что ввод будет из stdin, а не из файла. Подробнее смотрите в [ман-страницах](#). Конструирование конвейера из команд, каждая из которых решает свою задачу, составляет философию решения задач в Linux UNIX.

Одним из преимуществ конвейеров в системах Linux и UNIX является то, что в отличие от других популярных операционных систем, конвейеры не используют промежуточных файлов. Stdout первой команды **не** пишется в файл, который затем считывается второй командой. Если ваша конкретная версия [tar](#) не поддерживает разжатие файлов с помощью [bzip2](#), не беспокойтесь. Как мы видели в руководстве по Теме 102, вы можете просто использовать конвейер как, например

```
bunzip2 -c drgeo-1.1.0.tar.bz2 | tar -xvf -
```

### Вывод в качестве аргументов

В разделе [Использование командной строки](#) мы изучили подстановку команд и как использовать вывод одной команды как часть другой. В предыдущем разделе [Управление файлами](#) мы узнали как использовать опцию [-i](#) команды [find](#), чтобы использовать вывод команды [find](#) как ввод для другой команды. В Листинге 75 представлено три способа отображения содержимого файлов text1 и text2.

### Листинг 75. Использование вывода как аргументов с помощью подстановки команды и find -exec

```
[ian@echidna lpi103]$ cat `ls text[12]`  
1 apple  
2 pear  
3 banana  
9     plum  
3     banana  
10    apple  
[ian@echidna lpi103]$ cat $(find . -name "text[12]")  
1 apple  
2 pear  
3 banana  
9     plum  
3     banana  
10    apple  
[ian@echidna lpi103]$ find . -name "text[12]" -exec cat '{}' ';'   
1 apple  
2 pear  
3 banana  
9     plum  
3     banana  
10    apple
```

Приведенные примеры работают, но с ограничениями. Положим, что у вас есть файл, содержащий разделитель (пробел в этом случае). Посмотрите Листинг 76 и попробуйте понять, что делает каждая команда, прежде чем читать дальше.

### Листинг 76. Использование вывода в качестве аргументов с помощью подстановки команды и find -exec

```
[ian@echidna lpi103]$ echo grapes>"text sample2"  
[ian@echidna lpi103]$ cat `ls text*le2`  
cat: text: No such file or directory  
cat: sample2: No such file or directory  
[ian@echidna lpi103]$ cat "`ls text*le2`"  
grapes  
[ian@echidna lpi103]$ cat "`ls text*2`"  
cat: text2  
text sample2: No such file or directory
```

Вот, что мы делали.

- Мы создали файл "text sample2", содержащий одну строку со словом "grapes"
- Мы попытались использовать подстановку команды, чтобы отобразить содержимое "text sample2". Нас постигла неудача, так как bash передал **два** параметра команде cat, а именно text и sample2.
- Будучи умнее bash, мы заключили значения команды подстановки в кавычки. Это сработало
- Наконец, мы заменили шаблон, и вывод представляет собой странную ошибку. Здесь получилось так, что bash передал команде **cat** **один** параметр, который эквивалентен строке, являющей результатом

```
echo -e "text2\ntext sample2"
```

Если это кажется странным, попробуйте сами!

Что нам требуется так это способ выделения имен файлов вне зависимости от количества в них слов. Мы не упомянули ранее, что когда вывод команды как, например `ls`, используется в конвейере или команде подстановке, то он считывается построчно. Один способ состоит в использовании встроенной команды `read` в цикле со встроенной командой `while`. Хотя это и находится за рамками этого руководства, мы покажем данное решение.

### Листинг 77. Использование while и read в цикле

```
[ian@echidna lpi103]$ ls text*2 | while read l; do cat "$l";done
9      plum
3      banana
10     apple
grapes
```

## xargs

Большую часть времени мы будем обрабатывать списки файлов, поэтому нам требуется средства их создания и обработки. К счастью, у команды `find` есть опция `-print0`, которая разделяет строки своего вывода символом конца строки, а не символом новой строки. Такие команды как `tar` и `xargs` содержат опцию `-0` (или `--null`), которая позволяет им понимать такой тип параметров. Мы уже встречались с `tar`. Команда `xargs` работает почти как опция `-exec` команды `find`, однако существует большая разница между ними, которую мы увидим. Давайте посмотрим пример.

### Листинг 78. Использование xargs с -0

```
[ian@echidna lpi103]$ find . -name "text*2" -print0 |xargs -0 cat
9      plum
3      banana
10     apple
1 apple
2 pear
3 banana
grapes
```

Заметим, что теперь мы перенаправили вывод из `find` в `xargs`. Вам не надо использовать точку с запятой в конце команды и, по умолчанию, `xargs` добавляет аргументы к командной строке. Однако система выдала 7 строк вместо ожидаемых четырех. Что пошло не так?

## снова о find

Мы можем использовать команду `wc` для проверки того, что всего в двух файлах, вывод которых мы ожидали, четыре строки . Корень проблемы лежит в том, что `find` ведет поиск в каталоге с резервными копиями, в котором она находит `backup/text1.bkp.2`, соответствующий нашему шаблону. Чтобы решить проблему, воспользуемся опцией `-maxdepth` команды `find`, чтобы ограничить глубину поиска до текущего каталога. Есть также опция `-mindepth`, которая позволит еще более уточнить поиск. В Листинге 79 приведено полное решение.

## Листинг 79. Ограничение find

```
[ian@echidna lpi103]$ ls text*2
text2  text sample2
[ian@echidna lpi103]$ wc text*2
      3      6     25 text2
      1      1      7 text sample2
      4      7     32 total
[ian@echidna lpi103]$ find . -name "text*2" -maxdepth 1 -print0 |xargs -0 cat
9      plum
3      banana
10     apple
grapes
```

## Подробно о xargs

Существует разница между [xargs](#) и [find -exec](#).

- Команда [xargs](#) по умолчанию передает так много аргументов команде, насколько это возможно. Вы можете ограничить число входных строк с помощью [-l](#) или [--max-lines](#) за которой следует число. Кроме того, можете использовать [-n](#) или [--max-args](#) для ограничения количества передаваемых аргументов или [-S](#) или [--max-chars](#) для ограничения максимального числа символов в строке аргументов. Если ваша команда способна обработать несколько аргументов, то более эффективной будет передача ей как можно большего числа параметров за раз.
- Вы можете использовать '{}' как делали для [find -exec](#) если определите опцию [-i](#) или [--replace](#). Вы можете изменить поведение '{}' по умолчанию для строки, которые сигнализируют место подстановки входного параметра, определив значение для [-i](#). То есть подразумевается [-l 1](#).

Наш последний пример с [xargs](#) показан в Листинге 80.

## Листинг 80. Примеры xargs

```
[ian@echidna lpi103]$ # передача всех аргументов за раз
[ian@echidna lpi103]$ find . -name "text*2" |xargs echo
./text2 ./backup/text1.bkp.2 ./text sample2
[ian@echidna lpi103]$ # покажем файлы, которые мы создали раньше с помощью команды touch
[ian@echidna lpi103]$ ls f[0-n]*|xargs echo
f1 f1a f2 f3 f4 f5 f6 f7 f8 f9
[ian@echidna lpi103]$ # удалим их всех одной строкой
[ian@echidna lpi103]$ ls f[0-n]*|xargs rm
[ian@echidna lpi103]$ # используем строку подстановки
[ian@echidna lpi103]$ find . -name "text*2" |xargs -i echo - '{}' -
- ./text2 -
- ./backup/text1.bkp.2 -
- ./text sample2 -
[ian@echidna lpi103]$ # Ограничимся одной строкой вывода на вызов
[ian@echidna lpi103]$ find . -name "text*2" |xargs -l1 echo
./text2
./backup/text1.bkp.2
./text sample2
[ian@echidna lpi103]$ # Ограничимся одним аргументом на вызов
[ian@echidna lpi103]$ find . -name "text*2" |xargs -n1 echo
./text2
./backup/text1.bkp.2
```

```
./text  
sample2
```

Заметим, что мы не использовали здесь `-print0`. Объясняет ли это последний пример в Листинге 80?

### Разделение вывода

В этом разделе рассмотрим еще одну команду. Иногда вам понадобится просмотреть вывод на экране и сохранить его в файл для последующего использования. Хотя вы можете это сделать, перенаправив вывод в файл в одном окне, а затем использовать `tail -fn1`, чтобы увидеть текст на другом экране, использования команды `tee` гораздо проще.

Вы можете использовать `tee` в конвейере. Ее аргументы это файл (или файлы), в которые необходимо скопировать стандартный вывод. Опция `-a` добавляет, а не переписывает файлы. Как мы видели ранее в этом разделе при обсуждении конвейеров, вам понадобиться перенаправить stderr в stdout, до того как соединить его с `tee` в случае, если вам требуется оба потока. В Листинге 81 показано использование `tee` для сохранения вывода двух файлов f1 и f2.

### Листинг 81. Разделение stdout с помощью tee

```
[ian@echidna lpi103]$ ls text[1-3]|tee f1 f2  
text1  
text2  
text3  
[ian@echidna lpi103]$ cat f1  
text1  
text2  
text3  
[ian@echidna lpi103]$ cat f2  
text1  
text2  
text3
```

[предыдущая](#) | [следующая](#)

## Создание, отслеживание и уничтожение процессов

Эта глава содержит материал темы 1.103.5 экзамена Junior Level Administration (LPIC-1) 101. Тема имеет вес 5.

В этом разделе вы изучите следующие темы:

- Выполнение задач в приоритетном и фоновом режимах
- Запуск процесса без терминального ввода/вывода
- Отслеживание и отображение процессов
- Передача сигналов процессам
- Поиск и уничтожение процессов

Если остановитесь и немного задумаетесь, то станет довольно очевидно, что на вашем компьютере работает много программ. На самом деле в графическом режиме у вас может быть открыто несколько окон терминалов, браузер, игры, таблицы и другие приложения. В примерах мы вводили команды, ждали их выполнение и только потом могли продолжать

работу. В разделе [Использование командной строки](#) мы столкнулись с командой `ps`, которая отображала статус процесса, и мы видели, что у процесса есть собственный номер Process ID (PID) и номер родительского процесса Parent Process id (PPID). В этом разделе, вы изучите, как выполнять больше задач в одном окне терминала.

### Приоритетные и фоновые задачи

Когда вы выполняете команду в терминальном окне, как мы делали до этого, то вы запускали ее в *приоритетном режиме*. Наши команды работали довольно быстро, но предположим, что вы в графической среде и хотите запустить на рабочем столе цифровые часы. Не будем учитывать тот факт, что в большинстве сред они уже есть; мы просто рассматриваем как пример.

Если вы работаете в системе X Window, то возможно у вас есть такие утилиты как `xclock` или `xeyes`. Мы будем использовать `xclock`. В man-странице сказано, что вы можете запустить цифровые часы с помощью команды

```
xclock -d -update 1
```

Часть `-update 1` является запросом на обновление часов раз в секунду, иначе стрелки часов обновлялись бы раз в минуту. Давайте запустим ее в терминальном окне. Мы увидим картину как на Рисунке Figure 2, а терминал будет выглядеть как в Листинге 82. Если у вас нет `xclock` или системы X Window, мы покажем, как создать в терминале простые часы, чтобы вы могли выполнить упражнения.

**Рисунок 2. Часы xclock**



**Листинг 82. Работающие часы xclock**

```
[ian@echidna ian]$ xclock -d -update 1
```

К сожалению, терминальное окно больше не отображает приглашение, но нам надо как-то вернуться. К счастью, в Bash есть клавиша *приостановки* Ctrl-z. Нажав эту комбинацию, вы снова видите приглашение как показано в Листинге 83.

**Листинг 83. Приостановление xclock с помощью Ctrl-z**

```
[ian@echidna ian]$ xclock -d -update 1
[1]+  Stopped                  xclock -d -update 1
[ian@echidna ian]$
```

Часы все еще на рабочем столе, но они не работают. Произошло приостановление. На самом деле, если вы перетащите на него другое окно, но оно даже не перерисуется. Вы также видите, что на терминале отобразилось сообщение "[1]+ Stopped". В сообщении 1 это *номер задачи*. Вы можете перезапустить часы, набрав `fg %1`. Вы также можете использовать имя команды или часть ее, набрав `fg %xclock` или `fg %?clo`. Наконец, вы можете просто использовать `fg` без параметров для перезапуска самой последней задачи, job 1 в нашем случае. Перезапуск с помощью `fg` также влечет возврат задачи в приоритетный режим,

поэтому вы не видите приглашения. Что вам необходимо сделать, так это поместить задачу в фон; команда `bg` принимает те же параметры, что и команда `fg` и делает тоже самое. В Листинге 84 показано, как вернуть в приоритетный режим `xclock` и приостановить ее, используя две формы **команды `fg`**. Вы можете снова ее приостановить и поместить в фон; часы будут работать, а вы сможете продолжать работу в терминале.

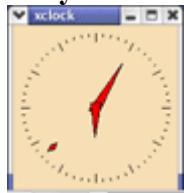
#### Листинг 84. Помещение `xclock` в фоновый режим работы

```
[ian@echidna ian]$ fg %1  
xclock -d -update 1  
  
[1]+ Stopped                  xclock -d -update 1  
[ian@echidna ian]$ fg %?clo  
xclock -d -update 1  
  
[1]+ Stopped                  xclock -d -update 1  
[ian@echidna ian]$ bg  
[1]+ xclock -d -update 1 &  
[ian@echidna ian]$
```

#### Использование "&"

Вы заметили, что, когда поместили `xclock` в фон, сообщение больше не гласило "Stopped" и что оно было завершено символом амперсанда (&). На самом деле, вам не надо приостанавливать процесс, чтобы помещать его в фон. Вы можете просто добавить в команду знак амперсанда, и shell интерпретатор запустит команду (или список команд) в фоном режиме. Давайте запустим аналоговые часы таким способом. Вы увидите часы как на Рисунке 3, а терминал будет выглядеть как в Листинге 85.

#### Рисунок 3. Аналоговые часы `xclock`



#### Листинг 85. Запуск `xclock` в фоне с помощью &

```
[ian@echidna ian]$ xclock -bg wheat -hd red -update 1&  
[2] 5659
```

Заметим, что на этот раз сообщение немного отличается. Оно содержит номер задачи и идентификатор процесса (PID). Мы рассмотрим PID и его статус немного позже. Сейчас рассмотрим команду **jobs** для поиска запущенных задач. Добавим опцию `-l`, чтобы посмотреть PID процессов, и увидим, что задача 2 имеет PID 5659 как показано в Листинге 86. Заметим также, что у задачи 2 стоит знак плюс (+) перед номером, сигнализирующий, что это *текущая задача*. Эта задача также перейдет в приоритетный режим, если не будет задана никакая работа команде `fg`.

#### Листинг 86. Отображение работы и информации о процессе

```
[ian@echidna ian]$ jobs -l
[1]- 4234 Running
[2]+ 5659 Running
          xclock -d -update 1 &
          xclock -bg wheat -hd red -update 1 &
```

Прежде, чем будем рассматривать другие вопросы, связанные с задачами, реализуем сами цифровые часы. Мы будем использовать команду `sleep` для создания задержки на две секунды, а затем используем команду `date` для выдачи текущей даты и времени. Мы вставим эти команды в цикл `while`, используя блок `do/done` для создания бесконечного цикла. Наконец, мы используем круглые скобки для создания списка команд и переведем весь список в фоновой режим с помощью амперсанда.

### Листинг 87. Цифровые часы

```
[ian@echidna ian]$ (while sleep 2; do date;done) &
[1] 16291
[ian@echidna ian]$ Thu Nov 10 22:58:02 EST 2005
Thu Nov 10 22:58:04 EST 2005
Thu Nov 10 22:58:06 EST 2005
Thu Nov 10 22:58:08 EST 2005
fThu Nov 10 22:58:10 EST 2005
Thu Nov 10 22:58:12 EST 2005
gThu Nov 10 22:58:14 EST 2005

( while sleep 2; do
  date;
done )
Thu Nov 10 22:58:16 EST 2005
Thu Nov 10 22:58:18 EST 2005
```

Как и ожидали, наш список выполняется как задача 1 с PID 16291. Каждые две секунды команда `date` выдает на терминал время и текущую дату. Ваш ввод выделен шрифтом. При медленном наборе символы будут рассеяны по экрану, прежде чем будет набрана вся команда. На самом деле заметим, что 'f' 'g', которые мы напечатали для выдачи списка в приоритетный режим, находятся на двух строках. После того, как мы ввели команду `fg`, bash отобразит команду, которая сейчас работает в интерпретаторе, а именно, список команд, который продолжает работать каждые две секунды.

Как только мы перевели задачу в приоритетный режим, то можем как завершить задачу (или убить), так и предпринять какое-либо другое действие. В этом случае мы используем `Ctrl-c` для завершения работы наших часов.

### Стандартный ввод/вывод и фоновые процессы

Вывод команды `date` в предыдущем примере рассеян вместе с символами команды `fg`, которую мы пытались набрать. Встает интересный вопрос. Что случится с процессом, если ему требуется ввод из `stdin`?

Терминальный процесс, в котором мы запустили фоновое приложение, называется *контролирующим терминалом*. В случае не использования перенаправления, потоки `stdout` и `stderr` фонового процесса направляются на контролирующий терминал. Также фоновая задача ожидает ввод от контролирующего терминала, но у контролирующего терминала нет способа передать ваши символы в `stdin` фонового процесса. В таком случае Bash приостанавливает процесс, так что он больше не исполняется. Вы можете вывести его в приоритетный режим и предоставить необходимый ввод. Листинг 88 иллюстрирует простые примеры, в которых вы

переключаете список команд в фоновый режим. Через некоторое время нажимаете **Enter** и процесс останавливается. Переводите задачу в приоритетный режим, предоставляете строку ввода, за которой следует сигнал окончания ввода Ctrl-d файла. Список команд завершается, и мы отображаем созданный файл.

### Листинг 88. Ожидание stdin

```
[ian@echidna ian]$ (date; cat - >bginput.txt; date)&
[1] 18648
[ian@echidna ian]$ Fri Nov 11 00:03:28 EST 2005

[1]+ Stopped                  ( date; cat - >bginput.txt; date )
[ian@echidna ian]$ fg
( date; cat - >bginput.txt; date )
input data
Fri Nov 11 00:03:53 EST 2005
[ian@echidna ian]$ cat bginput.txt
input data
```

### Задачи без терминалов

На практике вы возможно захотите осуществлять ввод\вывод для фоновых процессов из или в файлы Встает другой вопрос; что случится с процессом, если контролирующий терминал закроется или пользователь выйдет из системы? Ответ зависит от используемого интерпретатора. Если он посыпает сигнал SIGHUP (или зависание), то приложение закроется. Мы коротко рассмотрим сигналы, а сейчас рассмотрим другой способ решения этой проблемы.

#### nohup

Команда **nohup** используется для запуска команды, которая будет игнорировать сигналы зависания и добавлять stdout и stderr в файл. По умолчанию это файл nohup.out или \$HOME/nohup.out. Если писать в файл нельзя, то команда не запустится. Если вы хотите, то можете перенаправить куда угодно stdout или stderr как мы уже узнали из предыдущего раздела этого руководства.

Еще одной особенностью **nohup** является то, что она не будет выполнять конвейер или список команд. В теме [Перенаправление стандартного ввода\вывода](#) Мы рассмотрели, как использовать сценарии. Вы можете сохранить конвейер или список команд в файл и запустить его с помощью **sh** (интерпретатором по умолчанию) или командой **bash**, хотя вы не можете использовать команду . или **source** как мы делали в примере ранее. В следующем руководстве этой серии (по теме 104, рассказывающем об Устройствах, файловых системах Linux, структуре файловой системы) мы покажем, как сделать сценарий исполняемым, но сейчас мы будем использовать команду **sh** или **bash**. Листинг 89 показывает, как это можно сделать с нашими написанными часами. Не стоит и говорить, что запись времени в файл не особенно полезна, кроме того, файл будет расти в размере, поэтому мы установим интервал обновления каждые 30 секунд.

### Листинг 89. Использование nohup и сценария

```
[ian@echidna ian]$ echo "while sleep 30; do date;done">pmc.sh
[ian@echidna ian]$ nohup . pmc.sh&
[1] 21700
[ian@echidna ian]$ nohup: appending output to `nohup.out'
```

```
[1]+ Exit 126          nohup . pmc.sh
[ian@echidna ian]$ nohup sh pmc.sh&
[1] 21709
[ian@echidna ian]$ nohup: appending output to `nohup.out'

[ian@echidna ian]$ nohup bash pmc.sh&
[2] 21719
[ian@echidna ian]$ nohup: appending output to `nohup.out'
```

Если отобразим содержимое nohup.out, то увидим, что первая строка отображает причину получения кода выхода 126 в нашей первой попытке выше. Последующие строки являются выводом двух версий pmc.sh, которые сейчас работают в фоне. Это проиллюстрировано в Листинге 90.

### Листинг 90. Вывод не прерванных процессов

```
[ian@echidna ian]$ cat nohup.out
/bin/nice: .: Permission denied
Fri Nov 11 15:30:03 EST 2005
Fri Nov 11 15:30:15 EST 2005
Fri Nov 11 15:30:33 EST 2005
Fri Nov 11 15:30:45 EST 2005
Fri Nov 11 15:31:03 EST 2005
```

Сейчас обратим внимание на статус процесса. Если вы планируете передохнуть, то не спешите, так как вам предстоит создать еще две задачи, которые создадут еще большие файлы в вашей системе. Вы можете использовать команду [fg](#), чтобы перевести каждый процесс в приоритетный режим, а затем использовать Ctrl-c для его завершения, и если вы позволите им поработать чуть подольше, то увидим другие способы мониторинга и взаимодействия с ними.

### Статус процесса

В предыдущих частях этого раздела мы познакомились с командой [jobs](#) и увидели, как использовать ее для просмотра Process ID (или PID) наших задач.

### ps

Есть другая команда, команда [ps](#), которая отображает различную информацию о статусе процесса. Помните, что "ps" это акроним от "process status". Команда [ps](#) принимает ноль или более номеров PID в качестве аргументов и отображает статус соответствующих процессов. Если использовать команду [jobs](#) с опцией [-p](#), то вывод будет представлять собой PID лидера группы процессов каждой задачи. Мы будем использовать вывод этой команды как аргументы команды [ps](#), что и показано в Листинге 91.

### Листинг 91. Статус фоновых процессов

```
[ian@echidna ian]$ jobs
[1]- Running          nohup sh pmc.sh &
[2]+ Running          nohup bash pmc.sh &
[ian@echidna ian]$ jobs -p
21709
21719
[ian@echidna ian]$ ps `jobs -p`
```

```

PID TTY      STAT   TIME COMMAND
21709 pts/3    SN    0:00 sh pmc.sh
21719 pts/3    SN    0:00 bash pmc.sh

```

Если используем команду **ps** без всяких опций, то увидим список процессов, которые контролируются текущим терминалом как показано в Листинге 92.

### Листинг 92. Отображение статуса с помощью ps

```
[ian@echidna ian]$ ps
PID TTY      TIME CMD
20475 pts/3  00:00:00 bash
21709 pts/3  00:00:00 sh
21719 pts/3  00:00:00 bash
21922 pts/3  00:00:00 sleep
21930 pts/3  00:00:00 sleep
21937 pts/3  00:00:00 ps
```

Некоторые опции как **-f** (full), **-j** (jobs), и **-l** (long) позволяет отобразить информацию с нужной точностью. Если мы не определим никаких номеров PID, то существует другая полезная опция **--forest**, которая отображает команды в виде дерева, показывая порождение процессов. В частности, мы видим, что команды **sleep** предыдущего листинга являются детьми сценариев, которые мы запустили в фоновом режиме. Если мы запустим эту команду в другой момент времени, то можем увидеть команду **date** в списке процессов, однако на этом примере трудно увидеть другие расхождения. Проиллюстрируем на примере Листинга 93.

### Листинг 93. Расширенная информация о статусе процессов

```
[ian@echidna ian]$ ps -f
UID      PID  PPID  C STIME TTY          TIME CMD
ian      20475 20474  0 15:02 pts/3    00:00:00 -bash
ian      21709 20475  0 15:29 pts/3    00:00:00 sh pmc.sh
ian      21719 20475  0 15:29 pts/3    00:00:00 bash pmc.sh
ian      21945 21709  0 15:34 pts/3    00:00:00 sleep 30
ian      21953 21719  0 15:34 pts/3    00:00:00 sleep 30
ian      21954 20475  0 15:34 pts/3    00:00:00 ps -f
[ian@echidna ian]$ ps -j --forest
 PID  PGID  SID TTY          TIME CMD
20475 20475 20475 pts/3    00:00:00 bash
21709 21709 20475 pts/3    00:00:00 sh
21945 21709 20475 pts/3    00:00:00 \_ sleep
21719 21719 20475 pts/3    00:00:00 bash
21953 21719 20475 pts/3    00:00:00 \_ sleep
21961 21961 20475 pts/3    00:00:00 ps
```

### Список других процессов

Команды **ps**, которые мы использовали в примерах, выдавали список только тех процессов, которые запущены через терминал (обратим внимание на столбец SID во втором примере Листинга 93). Чтобы увидеть все процессы, контролируемые терминалами, используйте опцию **-a**. Опция **-x** отобразит процессы без контролирующего терминала, а опция **-e**

отобразит информацию для **каждого** процесса. В Листинге 94 приведен полный формат всех процессов, контролируемых терминалом.

#### Листинг 94. Отображение остальных процессов

```
[ian@echidna ian]$ ps -af
UID      PID  PPID   C STIME TTY          TIME CMD
ian      4234  32537  0 Nov10 pts/0    00:00:00 xclock -d -update 1
ian      5659  32537  0 Nov10 pts/0    00:00:00 xclock -bg wheat -hd red -update
ian     21709  20475  0 15:29 pts/3    00:00:00 sh pmc.sh
ian     21719  20475  0 15:29 pts/3    00:00:00 bash pmc.sh
ian     21969  21709  0 15:35 pts/3    00:00:00 sleep 30
ian     21977  21719  0 15:35 pts/3    00:00:00 sleep 30
ian     21978  20475  0 15:35 pts/3    00:00:00 ps -af
```

Заметим, что этот список включает два процесса `xclock`, которые мы запустили ранее в графическом режиме этой системы (о чем свидетельствует `pts/0`), в то время как оставшиеся процессы ассоциированы с `ssh` (Secure Shell) соединением (`pts/3` в этом случае).

Существует множество опций у команды `ps`, которые позволяют отобразить нужную информацию. Например, опции, позволяющие отобразить процессы конкретного пользователя. За подробностями обращайтесь к `man`-страницам команды `ps` или же можете получить краткое описание, используя команду `ps --help`.

#### top

Если вы используете команду `ps` несколько раз к ряду, чтобы увидеть различные изменения, то возможно вам стоит вместо нее использовать команду `top`. Она выводит постоянно обновляющийся список процессов и некоторую полезную информацию. Смотри `man`-страницы `top`, чтобы узнать о списке опций, а также о том, как сортировать процессы по объему используемой памяти или другим критериям. В Листинге 95 показано несколько первых строк вывода команды `top`.

#### Листинг 95. Вывод других процессов

```
3:37pm up 46 days, 5:11, 2 users, load average: 0.01, 0.17, 0.19
96 processes: 94 sleeping, 1 running, 0 zombie, 1 stopped
CPU states: 0.1% user, 1.0% system, 0.0% nice, 0.9% idle
Mem: 1030268K av, 933956K used, 96312K free, 0K shrd, 119428K buff
Swap: 1052216K av, 1176K used, 1051040K free, 355156K cached

 PID USER      PRI  NI   SIZE  RSS SHARE STAT %CPU %MEM      TIME COMMAND
22069 ian        17   0  1104 1104   848 R      0.9  0.1  0:00 top
  1 root       8   0   500  480   444 S      0.0  0.0  0:04 init
  2 root       9   0     0    0     0 SW     0.0  0.0  0:00 keventd
  3 root       9   0     0    0     0 SW     0.0  0.0  0:00 kapmd
  4 root      19  19     0    0     0 SWN    0.0  0.0  0:00 ksoftirqd_CPU0
  5 root       9   0     0    0     0 SW     0.0  0.0  0:00 kswapd
```

#### Сигналы

Рассмотрим теперь *сигналы* в Linux. Они являются асинхронным средством взаимодействия с процессами. Мы уже упомянули сигнал `SIGHUP`, а также использовали комбинации `Ctrl-c` и `Ctrl-z` для посылки сигнала процессам. Главный способ посылки сигнала состоит в

использовании команды **kill**.

### Посылка сигналов с помощью **kill**

Команда **kill** посыпает сигнал определенной задаче или процессу. Листинг 96 содержит пример использования сигналов SIGTSTP и SIGCONT для остановки и возобновления фоновой задачи. Использование сигнала SIGTSTP эквивалентно использованию команды **fg** для перевода задачи в приоритетный режим, а затем Ctrl-z для ее приостановки. Действие сигнала SIGCONT похоже на работу команды **bg**.

### Листинг 96. Остановка и запуск фоновых задач

```
[ian@echidna ian]$ kill -s SIGTSTP %1
[ian@echidna ian]$ jobs -l
[1]+ 21709 Stopped                  nohup sh pmc.sh
[2]- 21719 Running                 nohup bash pmc.sh &
[ian@echidna ian]$ kill -s SIGCONT %1
[ian@echidna ian]$ jobs -l
[1]+ 21709 Running                 nohup sh pmc.sh &
[2]- 21719 Running                 nohup bash pmc.sh &
```

В этом примере мы использовали номер задачи (%1), но вы также можете посыпать сигналы по идентификатору процесса (то есть 21709 является PID задачи %1). Если вы используете команду **tail** в то время как задача %1 остановлена, то только один процесс изменит файл nohup.out.

Существует всевозможное множество сигналов в вашей системе, которые вы можете отобразить с помощью команды **kill -l**. Некоторые используются для сообщения об ошибках, например для сообщения о неверных кодах операции, исключения при работе с плавающей точкой или попытке обратится к памяти другого процесса. Заметим, что у сигналов есть как номер, например, 20, так и имя, например, SIGTSTP. Вы можете использовать как номер, так и имя с помощью опции **-s**. Следует всегда проверять номера сигналов на системе, прежде чем делать какие-либо предположения о сигналах.

### Обработчики сигналов и завершение процесса

Мы уже видели, что Ctrl-c завершает процесс. На самом деле она посыпает сигнал SIGINT (или прерывание) процессу. Если вы используете **kill** без указания сигнала, то система пошлет сигнал SIGTERM. Для большинства применений эти два сигнала эквивалентны.

Мы сказали, что команда **nohup** иммунизирует процесс от восприятия сигнала SIGHUP. В общем случае процесс может реализовать *обработчик сигнала для перехвата* сигналов. Поэтому процесс может реализовать обработчик сигнала для перехвата как SIGINT так и SIGTERM. Так как обработчик сигнала знает, какой сигнал был послан, он может, например, проигнорировать сигнал SIGINT и завершить работу только при получении сигнала SIGTERM. В Листинге 97 показано, как послать сигнал SIGTERM задаче %1. Заметим, что статус процесса изменился на "Завершен" сразу после того, как мы послали сигнал. Статус изменится на "Прерван", если мы пошлем сигнал SIGINT. Через несколько мгновений произойдет очистка процессов, и теперь задача больше не находится в списке задач.

### Листинг 97. Завершение процесса с помощью SIGTERM

```
[ian@echidna ian]$ kill -s SIGTERM %1
[ian@echidna ian]$ jobs -l
[1] 21709 Terminated                nohup sh pmc.sh
```

```
[2]- 21719 Running          nohup bash pmc.sh &
[ian@echidna ian]$ jobs -l
[2]+ 21719 Running          nohup bash pmc.sh &
```

Обработчики сигналов предоставляют процессу гибкость в том, что он может выполнять свою обычную работу и прерываться по сигналу только для особых целей. Кроме того, что процесс может перехватить запросы на окончание работы и возможно предпринять определенные действия, как например закрытие файлов или проверить работу текущей транзакции, сигналы часто используются, чтобы сообщить демону о его перезапуске и повторном прочтении файла конфигурации. Вы можете сделать это с процессом inetd, чтобы ваши изменения параметров сети вступили в силу или послать сигнал демону печати (lpd), когда добавляете новый принтер.

### Безаппеляционное завершение процессов

Некоторые сигналы не могут быть перехвачены, как например некоторые аппаратные исключения. SIGKILL, который скорее всего вы будете использовать, нельзя отловить обработчиком сигналов, и он используется для завершения процесса. В общем случае его следует использовать только, когда другие средства завершения работы процесса не помогают.

### Logout и nohup

Помните, мы говорили, что команда `nohup` позволит всем нашим процессам продолжать работу после нашего выхода из системы. Давайте сделаем это, а затем снова войдем в систему. После нашего возвращения проверим статус процесса, исполняющего наши написанные часы с помощью `jobs` и `ps` как мы уже делали до этого. Вывод представлен в Листинге 98.

### Листинг 98. Повторный заход в систему

```
[ian@echidna ian]$ jobs
[ian@echidna ian]$ ps -a
  PID TTY      TIME CMD
 4234 pts/0    00:00:00 xclock
 5659 pts/0    00:00:00 xclock
27217 pts/4    00:00:00 ps
```

Мы видим, что работаем в этот раз на pts/4, но наших задач нет и следа, как будто мы только запустили команду `ps` и два процесса xclock в графическом режиме с терминала (pts/0). Не совсем то, что мы ожидали увидеть. Однако не все потеряно. В Листинге 99 мы покажем один способ как найти потерянные задачи с помощью опции `-S` означающей идентификатор сессии, а также идентификатор сессии 20475, который мы видели в [Листинге 93](#). Подумайте о других способах обнаружения задач для случая, если вы не знаете идентификатора сессии.

### Листинг 99. Повторный заход в систему

```
[ian@echidna ian]$ ps -js 20475
  PID  PGID   SID TTY      TIME CMD
21719 21719 20475 ?        00:00:00 bash
27335 21719 20475 ?        00:00:00 sleep
```

Зная о том, как убивать процессы, вы можете убить их, используя PID и команду **[kill](#)**.

| [предыдущая](#) | [следующая](#)

## Приоритеты исполнения процесса

Эта тема описывает материал темы 1.103.6 экзамена Junior Level Administration (LPIC-1) 101. Тема имеет вес 3.

В этом разделе вы изучите следующие темы:

- Приоритеты исполнения процесса
- Установка приоритетов
- Изменение приоритетов

## Приоритеты

Как мы уже видели в предыдущем разделе, Linux, как и большинство современных операционных систем выполняет множество процессов. Это достигается путем разделения CPU и других ресурсов всеми процессами. Если некоторый процесс может использовать 100% ресурсов CPU, то другие процессы могут перестать отвечать на запросы и вообще что-то делать. Когда мы рассматривали [Статус процесса](#) в предыдущем разделе, то видели, что вывод по умолчанию команды **top** выдает список процессов, расположенных в порядке убывания потребления ресурсов CPU. Если мы запустим наши часы и команду **top**, то вряд ли увидим этот процесс в списке, потому как большую часть времени он не использует ресурсы CPU.

В вашей системе могут быть команды, которые могут использовать несколько CPU. Это такие программы как видео-редакторы, программы преобразования изображений или же кодирования звука, как например mp3 в ogg.

Мы создадим небольшой сценарий, который использует CPU и делает немного больше. Он принимает два параметра, счетчик и метку. Он выводит метку, текущую дату и время, затем уменьшает счетчик до тех пор, пока не достигнет 0, затем снова печатает метку и дату. Этот сценарий не проверяет ошибки, но зато он подходит для иллюстрации.

### Листинг 100. Скрипт, интенсивно использующий CPU

```
[ian@echidna ian]$ echo 'x="$1">count1.sh
[ian@echidna ian]$ echo 'echo "$2" $(date)'>>count1.sh
[ian@echidna ian]$ echo 'while [ $x -gt 0 ]; do let x=$x-1;done'>>count1.sh
[ian@echidna ian]$ echo 'echo "$2" $(date)'>>count1.sh
[ian@echidna ian]$ cat count1.sh
x="$1"
echo "$2" $(date)
while [ $x -gt 0 ]; do let x=$x-1;done
echo "$2" $(date)
```

Если вы запустили этот сценарий на своей системе, то можете увидеть результат как в Листинге 101. Этот сценарий интенсивно использует CPU, как мы скоро увидим. Если вы используете не свою рабочую станцию, убедить что вам можно использовать ресурсы CPU .

### Листинг 101. Запуск count1.sh

```
[ian@echidna ian]$ sh count1.sh 10000 A
A Mon Nov 14 07:14:04 EST 2005
```

```
A Mon Nov 14 07:14:05 EST 2005
[ian@echidna ian]$ sh count1.sh 99000 A
A Mon Nov 14 07:14:26 EST 2005
A Mon Nov 14 07:14:32 EST 2005
```

Пока все хорошо. Давайте теперь используем полученные ранее знания и создадим список команд, давайте запустим сценарий в фоновом режиме работы и используем команду **top**, чтобы увидеть количество ресурсов CPU, потребляемое сценарием. Список команд показан в Листинге 102, а вывод команды **top** в Листинге 103.

### Листинг 102. Запуск count1.sh и top

```
[ian@echidna ian]$ (sh count1.sh 99000 A&);top
```

### Листинг 103. Интенсивное использование CPU

```
7:20am up 48 days, 20:54, 2 users, load average: 0.05, 0.05, 0.00
91 processes: 88 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 0.1% user, 0.0% system, 0.0% nice, 0.9% idle
Mem: 1030268K av, 1002864K used, 27404K free, 0K shrd, 240336K buff
Swap: 1052216K av, 118500K used, 933716K free 605152K cached

PID USER      PRI  NI   SIZE  RSS SHARE STAT %CPU %MEM     TIME COMMAND
8684 ian        20   0  1044 1044    932 R    98.4  0.1    0:01 sh
```

Неплохо. С помощью простого сценария мы заняли 98.4% ресурсов CPU.

### Отображение и установка приоритетов

Если мы выполняем большую задачу, то можем обнаружить, что она влияет на нашу возможность (или же возможности других пользователей) выполнения других задач в системе. Системы Linux и UNIX используют систему приоритетов из 40 значений, начиная от -20 (наивысший приоритет) и заканчивая 19 (низший приоритет).

#### nice

Процессы обычных пользователей обычно имеют нулевой приоритет. Команда **nice** отобразит наш приоритет по умолчанию. Команда **ps** также может отображать приоритет (**nice**, или **NI**, уровень), например с помощью опции **-l**. Проиллюстрируем на примере Листинга 104, в котором выделен наш приоритет в виде значения 0.

### Листинг 104. Отображение информации о приоритетах

```
[ian@echidna ian]$ nice
0
[ian@echidna ian]$ ps -l
  F S  UID   PID  PPID C PRI  NI ADDR      SZ WCHAN  TTY          TIME CMD
000 S  500  7283  7282  0  70   0    - 1103 wait4  pts/2    00:00:00 bash
000 R  500  9578  7283  0  72   0    -  784 -          pts/2    00:00:00 ps
```

Команда **nice** также может использоваться для запуска процесса с другим приоритетом. Вы

можете использовать опцию `-n` или `(--adjustment)` вместе с положительным числом, чтобы увеличить приоритет или отрицательное число, чтобы уменьшить его. Помните, что процесс с наименьшим значением приоритета работает чаще всех, поэтому считайте, что увеличение значения приоритета означает для процесса быть более *дружелюбным* по отношению к другим процессам. Заметим, что вам обычно требуются права суперпользователя (root), чтобы применять отрицательные значения. Другими словами, обычные пользователи могут только сделать свои процессы более дружелюбными. В Листинге 105, мы запустим две копии сценария `count1.sh` в фоновом режиме с разными приоритетами исполнения. Заметим, что между окончанием их работы появилась задержка в 5 секунд. Попытайтесь поэкспериментировать с разными значениями `nice`, или же запустить с другим значением приоритета, чтобы увидеть возможную разницу.

### Листинг 105. Использование `nice` для установки приоритетов

```
[ian@echidna ian]$ (sh count1.sh 99000 A&);\
> (nice -n 19 sh count1.sh 99000 B&);\
> sleep 2;ps -l;sleep 20
B Mon Nov 14 08:17:36 EST 2005
A Mon Nov 14 08:17:36 EST 2005
  F S  UID  PID  PPID  C PRI  NI ADDR     SZ WCHAN   TTY      TIME CMD
000 S  500  7283  7282  0 70    0   -  1104 wait4  pts/2    00:00:00 bash
000 R  500 10765     1 84  80    0   -  1033 -        pts/2    00:00:01 sh
000 R  500 10767     1 14  79   19   -  1033 -        pts/2    00:00:00 sh
000 R  500 10771  7283  0 72    0   -   784 -        pts/2    00:00:00 ps
A Mon Nov 14 08:17:43 EST 2005
B Mon Nov 14 08:17:48 EST 2005
```

Заметим также, что, как и в команде `nohup` вы не можете использовать список команд или конвейер как аргумент `nice`.

### Изменение приоритетов

#### `renice`

Если вы запустили процесс и поняли, что он должен работать с другим приоритетом, то существует способ изменить приоритет работающего процесса с помощью команды `renice`. Вы указываете абсолютный приоритет (а не величину изменения) процесса или процессов, приоритет которых хотите изменить. Смотри Листинг 106.

### Листинг 106. Использование `renice` для изменения приоритетов

```
[ian@echidna ian]$ sh count1.sh 299000 A&
[1] 11322
[ian@echidna ian]$ A Mon Nov 14 08:30:29 EST 2005

[ian@echidna ian]$ renice +1 11322;ps -l
11322: old priority 0, new priority 1
  F S  UID  PID  PPID  C PRI  NI ADDR     SZ WCHAN   TTY      TIME CMD
000 S  500  7283  7282  0 75    0   -  1104 wait4  pts/2    00:00:00 bash
000 R  500 11322  7283 96 77    1   -  1032 -        pts/2    00:00:11 sh
000 R  500 11331  7283  0 76    0   -   786 -        pts/2    00:00:00 ps
[ian@echidna ian]$ renice +3 11322;ps -l
11322: old priority 1, new priority 3
  F S  UID  PID  PPID  C PRI  NI ADDR     SZ WCHAN   TTY      TIME CMD
000 S  500  7283  7282  0 75    0   -  1104 wait4  pts/2    00:00:00 bash
000 R  500 11322  7283 93 76    3   -  1032 -        pts/2    00:00:16 sh
```

```
000 R 500 11339 7283 0 76 0 - 785 - pts/2 00:00:00 ps
```

Больше информации о командах `nice` и `renice` вы можете получить из man-страниц.

| [предыдущая](#) | [следующая](#)

## Поиск с помощью регулярных выражений

Этот раздел описывает материал темы 1.103.7 экзамена Junior Level Administration (LPIC-1) 101. Тема имеет вес 3.

В этом разделе рассмотрены следующие темы:

- Регулярные выражения
- Поиск в файлах и файловой системе с помощью регулярных выражений
- Использование регулярных выражений совместно с `sed`

## Регулярные выражения

Регулярные выражения впервые появились в теории компьютерных языков. Большинство студентов по computer science учат, что язык, описываемый регулярными выражениями, в точности такой, какой принимает конечный автомат. Регулярные выражения в этом разделе могут нести более сложный смысл, поэтому они **не** в точности такие же, какие вы изучали на занятиях по информатике, хотя у них одинаковое родство.

Регулярные выражения (также называемые как " regex" или "regexp") представляют способ описания текстовой строки или *шаблона* таким образом, что программа может осуществлять *соответствие* шаблона в произвольных текстовых строках, обеспечивая мощные инструменты поиска информации. Утилита `grep` (от *generalized regular expression processor*) является стандартной частью инструментария программиста или администратора Linux или UNIX, позволяя использовать регулярные выражения для поиска файлов или вывода команды. В разделе о [текстовых потоках и фильтрах](#) мы познакомились с `sed` или *stream editor*, который является еще одним стандартным инструментом, использующим регулярные выражения для поиска и замены текста в файлах или текстовых потоках. Этот раздел поможет лучше понять использование регулярных выражений в `grep` и `sed`. Другой программой, использующей регулярные выражения, является `awk`, которая входит в материал экзамена 201 на сертификацию LPIC-2.

Как и по остальным темам этого руководства, по регулярным выражениям и теории языков написано много книг. Смотри раздел [Ресурсы](#), чтобы узнать о дополнительных источниках информации.

Как только вы узнаете о регулярных выражениях, то увидите сходство между синтаксисом регулярных выражений и шаблонами (или подстановкой), описанными в разделе [Шаблоны и подстановки](#). Сходство это только поверхностно.

## Основные строительные блоки

В программе GNU grep, которую можно найти на большинстве Linux систем, используется два синтаксиса регулярных выражений: *основной* и *расширенный*. У программы GNU grep нет никаких функциональных отличий. Здесь описан основной синтаксис, а также отличие его от расширенного синтаксиса.

Регулярные выражения состоят из *символов и операторов*, дополненных *метасимволами*. Большинство символов соответствует своим значениям, а большинство метасимволов необходимо предварять обратным слешем (`\`). Основными операторами являются

## Конкатенация

Конкатенация соединяет два регулярных выражения. Например, регулярное выражение **a** найдет соответствие в строке **abcdcba** дважды (первый и последний символ **a**) точно также и с регулярным выражением **b**. Однако **ab** соответствует только **abcdcba**, в то время как **ba** соответствует только **abcdcba**.

## Повторение

Оператор Клини \* или повторения соответствует нулю или более появлений предшествующего регулярного выражения. Поэтому выражение **a\*b** соответствует любой строке из **a**, завершенной символом **b** включая самой строке из символа **b**.

Оператор Клини \* не надо записывать как escape-последовательность, но в выражении, если вы хотите найти сам символ (\*) необходимо писать escape-последовательность.

## Выбор

Оператор выбора () соответствует либо предшествующему, либо последующему выражению. Его надо писать, как escape-последовательность, если используется основной синтаксис. Например, выражение **a\*\|b\*c** найдет строку из любого числа **a** или любого числа букв **b** (но не оба), завершенную символом **c**. Снова простой символ **c** соответствует такому выражению.

Вам часто потребуется заключать регулярные выражения в кавычки, чтобы избежать подстановки интерпретатором.

В качестве примеров мы будем использовать текстовые файлы, созданные ранее в каталоге lpi103. Рассмотрите простые примеры в Листинге 107. Заметим, что **grep** принимает регулярное выражение как обязательный параметр и ноль и более файлов, в которых надо осуществить поиск. Если никаких файлов не указано, то grep будет искать в **stdin**, что делает ее похожей на фильтр, который можно использовать в конвейере. Если соответствия не найдено, то вывода **grep** не будет, хотя можно проверить ее код выхода.

## Листинг 107. Простые регулярные выражения

```
[ian@echidna lpi103]$ grep p text1
1 apple
2 pear
[ian@echidna lpi103]$ grep pea text1
2 pear
[ian@echidna lpi103]$ grep "p*" text1
1 apple
2 pear
3 banana
[ian@echidna lpi103]$ grep "pp*" text1
1 apple
2 pear
[ian@echidna lpi103]$ grep "x" text1
[ian@echidna lpi103]$ grep "x*" text1
1 apple
2 pear
3 banana
[ian@echidna lpi103]$ cat text1 | grep "l\|n"
1 apple
3 banana
[ian@echidna lpi103]$ echo -e "find an\n* here" | grep "\*"
* here
```

Глядя на приведенные примеры, вы иногда можете удивиться полученным результатам,

особенно при использовании повторения. Вы, возможно, ожидали, что **p\*** или, по крайней мере **pp\*** соответствует паре символов **p**, но **p\*** и **x\*** тоже соответствует каждой строке файла, так как оператор **\*** соответствует **нулю** или большему числу раз предшествующего регулярного выражения.

## Первые ярлыки

Вы знаете основные строительные блоки регулярных выражений, давайте рассмотрим некоторые удобные сокращения.

+

Оператор **+** похож на оператор **\***, за исключением того, что он соответствует **одному** или более вхождению предыдущего регулярного выражения. В основном синтаксисе его необходимо писать как escape-последовательность.

?

Символ **?** означает ноль или более вхождений предыдущего выражения. Это не тот же знак **?**, используемый при подстановках.

.

Метасимвол **.** (точка) означает любой символ. Одним из наиболее часто используемых шаблонов является **.\***, который соответствует строке произвольной длины из любых символов (или не содержащей символов совсем). Сравните точку со знаком **?**, используемым в подстановке и **.\*** с **\***, используемым в подстановке.

## Листинг 108. Регулярные выражения

```
[ian@echidna lpi103]$ grep "pp\+" text1 # at least two p's
1 apple
[ian@echidna lpi103]$ grep "pl\?e" text1
1 apple
2 pear
[ian@echidna lpi103]$ grep "pl\?e" text1 # p with optional l between
1 apple
2 pear
[ian@echidna lpi103]$ grep "p.*r" text1 # p, some string then r
2 pear
[ian@echidna lpi103]$ grep "a.." text1 # a followed by two other letters
1 apple
3 banana
```

## Соответствие начала или конца строки

Знак **^** (карапки) означает начало строки, в то время как **\$** (знак доллара) означает конец строки. Поэтому **^.b** соответствует любым двум символам в начале строки, за которыми следует **b**, в то время как **ar\$** соответствует любой строке, заканчивающейся на **ar**.

Регулярное выражение **^\\$** соответствует пустой строке.

## Более сложные выражения

До сих пор мы рассматривали применение повторения к одному символу. Если вы хотите найти одно или более вхождений строки из нескольких символов как **an**, которая встречается дважды в **banana**, используйте круглые скобки, которые надо записывать как escape-последовательности в основном синтаксисе. Также вы можете захотеть осуществить поиск нескольких символов, не используя такой обобщенный оператор как **.** или длинную последовательность альтернатив. Это можно сделать, заключив альтернативы в квадратные скобки (**[]**), которые не надо писать как escape-последовательность при использовании

основного синтаксиса. Выражения в квадратных скобках составляют *класс символов*. Кроме нескольких исключений, о которых мы поговорим позже, использование квадратных скобок позволяет обойтись без использования escape-последовательностей при использовании специальных символов, таких как as . и \*.

### Листинг 109. Круглые скобки и классы символов

```
[ian@echidna lpi103]$ grep "\(\an\)\+" text1 # find at least 1 an
3 banana
[ian@echidna lpi103]$ grep "an\(\an\)\+" text1 # find at least 2 an's
3 banana
[ian@echidna lpi103]$ grep "[3p]" text1 # find p or 3
1 apple
2 pear
3 banana
[ian@echidna lpi103]$ echo -e "find an\n* here\nsomewhere." | grep "[.*]"
* here
somewhere.
```

Существует несколько дополнительных возможностей при работе с классами символов.

#### Диапазон выражения

Диапазон выражения это два символа, разделенных знаком - (дефис), как например 0-9 для цифр или 0-9a-fA-F для шестнадцатеричных чисел. Заметим, что диапазон зависит от локали.

#### Именованные классы

Несколько именованных классов обеспечивают удобное обозначение для часто используемых классов. Именованные классы начинаются с [: и заканчиваются :].

Некоторые примеры:

**[:alnum:]**

Цифровые и буквенные символы

**[:blank:]**

Пробелы и символы табуляции

**[:digit:]**

Цифры от 0 до 9 (эквивалентно от 0-9)

**`[:upper:] и [:lower:]**

Соответственно буквы верхнего и нижнего регистра.

#### ^(отрицание)

Будучи использован на первом месте в квадратных скобках, знак ^ (каретка) дополняет значение остальных символов, так что соответствие происходит, только если (кроме ведущего ^) подстрока не принадлежит классу.

Зная особые значения символов выше, делаем вывод, что если вы хотите обнаружить символ - (дефис) в классе символа, то должны помещать его первым или последним. Если вы хотите обнаружить символ ^ (каретка), то не ставьте его первым. Знак ] (правая квадратная скобка) закрывает класс, кроме случая, когда он стоит первым.

Классы символов -- это та область, в которой регулярные выражения и подстановка **ведут себя одинаково**, хотя отрицание отличается (^ против !). В Листинге 108 приведены примеры классов символов.

### Листинг 110. Классы символов

```
[ian@echidna lpi103]$ # Ищет символы от 3 до 7
[ian@echidna lpi103]$ echo -e "123\n456\n789\n0" | grep "[3-7]"
123
456
789
[ian@echidna lpi103]$ # Ищем цифру, за которой до конца строки нет букв п и г
[ian@echidna lpi103]$ grep "[[:digit:]][^nr]*$" text1
1 apple
```

## Использование регулярных выражений совместно с sed

В кратком введении в [Sed](#) упоминалось о том, что sed использует регулярные выражения. Regexp могут использоваться как в выражениях адресации, так и в выражениях подстановки. Так, выражение `/abc/s/xyz/XYZ/g` означает: применить подстановку команды, которая заменит на XYZ все вхождения xuz **только** в строках, содержащих abc. В Листинге 111 приведено два примера с файлом text1, а в другом мы заменяем последнее слово перед точкой(.) на строку LAST WORD. Заметим, что строка First не изменилась, так как не была предварена пробелом.

## Листинг 111. Регулярные выражения в sed

```
[ian@echidna lpi103]$ sed -e '/\(\.*a\)\|\(\.*p\)/s/a/A/g' text1
1 Apple
2 pear
3 bAnAnA
[ian@echidna lpi103]$ sed -e '/[^lmnXYZ]*$/s/ear/each/g' text1
1 apple
2 peach
3 banana
[ian@echidna lpi103]$ echo "First. A phrase. This is a sentence." | \
> sed -e 's/ [^ ]*/./ LAST WORD./g'
First. A LAST WORD. This is a LAST WORD.
```

## Расширенные регулярные выражения

Расширенные регулярные выражения позволяют не писать escape-последовательности нескольких символов, которые приходилось предварять знаком \ в основном синтаксисе, включая круглые скобки, '?', '+', '|', и '{'. Это значит, что они должны быть записаны как escape-последовательность, только если вы хотите, чтобы они интерпретировались как символы. Вы можете использовать опцию `-E` (или `--extended-regexp`) команды grep, чтобы сигнализировать об использовании расширенного синтаксиса регулярных выражений. Можно использовать альтернативу в виде [egrep](#). Некоторые старые версии [sed](#) не поддерживают расширенные регулярные выражения. Если ваша версия [sed](#) поддерживает расширенные regexps, используйте опцию `-r`, чтобы сообщить [sed](#), что вы будете использовать расширенный синтаксис. В Листинге 112 показан пример, рассмотренный ранее, с использованием расширенной версии [egrep](#).

## Листинг 112. Расширенные регулярные выражения

```
[ian@echidna lpi103]$ grep "an\(\an\)\+" text1 # find at least 2 an's
3 banana
[ian@echidna lpi103]$ egrep "an(an)+" text1 # find at least 2 an's
3 banana
```

## Поиск информации в файлах

Этот раздел завершает некоторые примеры мощных команд `grep` и `find`, которые позволяют искать информацию в файловой системе. Снова, примеры довольно простые; мы используем файлы, созданные в каталоге `lpi103` и его детях.

Для начала `grep` может искать сразу в нескольких файлах. Если вы добавите опцию `-n`, то она скажет номера найденных строк. Если вы просто хотите знать количество найденных строк, используйте опцию `-c`, а если вам нужен список файлов, в которых есть совпадения, используйте опцию `-l`. В Листинге 113 приведены некоторые примеры.

### Листинг 113. Поиск в нескольких файлах

```
[ian@echidna lpi103]$ grep plum *
text2:9 plum
text6:9 plum
text6:9 plum
yaa:9   plum
[ian@echidna lpi103]$ grep -n banana text[1-4]
text1:3:3 banana
text2:2:3      banana
[ian@echidna lpi103]$ grep -c banana text[1-4]
text1:1
text2:1
text3:0
text4:0
[ian@echidna lpi103]$ grep -l pear *
ex-here.sh
nohup.out
text1
text5
text6
xaa
```

Наш последний пример использует команду `find`, чтобы найти все обычные файлы в текущем каталоге и его детях, а затем использовании `xargs` для передачи списка файлов команде `grep`, которая определит число появлений строки `banana` в каждом файле. Наконец, вывод фильтруется через другой экземпляр `grep`, на этот раз с опцией `-v` для поиска всех файлов, которые **не** содержат ни одного появления искомой строки.

### Листинг 114. Поиск файлов, в которых есть хотя бы одно вхождение `banana`

```
[ian@echidna lpi103]$ find . -type f -print0| xargs -0 grep -c banana| grep -v ":0$"
./text1:1
./text2:1
./xab:1
./yaa:1
./text5:1
./text6:4
./backup/text1.bkp.2:1
./backup/text1.bkp.1:1
```

В этом разделе мы затронули лишь малую часть того, что вы можете делать с помощью командной строки Linux и регулярных выражений. Более подробную информацию вы можете

прочесть в man-страницах.

| [предыдущая](#) | [следующая](#)

## Редактирование файлов в vi

Этот раздел описывает материал темы 1.103.8 экзамена Junior Level Administration (LPIC-1)101. Тема имеет вес 1.

В этом разделе рассмотрены следующие темы:

- Редактирование текста в vi

### Использование vi

Редактор vi есть почти в каждой системе Linux и UNIX. На самом деле, если в системе есть только один текстовый редактор, то это наверняка vi, поэтому следует знать как им пользоваться. В этом разделе представлены основные команды vi, а для полного руководства по vi, обратитесь к нашему "введению в vi -- метод шпаргалки" (смотри [Ресурсы](#)), или же обратитесь к man-страницам или многочисленным книгам.

#### Запуск vi

Большинство дистрибутивов Linux сейчас поставляется с vim (от ViIMproved) редактором, а не классическим vi. Vim обратно совместим с vi, для которого также доступна графическая оболочка (gvim), а также обычный текстовый режим. Команда vi обычно является псевдонимом или символьной ссылкой на программу. Просмотрите тему [Откуда интерпретатор берет команды?](#), чтобы узнать какая точно команда используется.

Вы можете вспомнить [изменение приоритетов](#), в котором мы пытались изменить приоритет работающего сценария count1.sh. Возможно, вы пытались сделать это сами, но команда выполнялась так быстро, что вы не успевали изменить приоритет с помощью `renice`. Давайте запустим редактор vi и добавим строку в начало файла, чтобы заснуть на 20 секунд, и у нас появилось время, чтобы изменить приоритеты.

Чтобы запустить редактор vi, используйте команду `vi`, а также имя файла в качестве параметра. Редактор имеет много опций. За более полной информацией обратитесь к man-страницам. Наберите команду

**vi count1.sh**

Вы увидите вывод как в Листинге 115. Если вы используете vim, некоторые слова могут быть подсвечены другим цветом. Vim поддерживает подсветку синтаксиса (она не являлась частью редактора vi), и по умолчанию она может быть включена.

#### Листинг 115. Редактирование count1.sh в vi

```
x="$1"
echo "$2" $(date)
while [ $x -gt 0 ]; do let x=$x-1;done
echo "$2" $(date)
~
~
~
~
~
~
"count1.sh" 4L, 82C
```

## Режимы vi

Редактор vi может работать в двух режимах:

### Режим команд

В режиме команд, вы перемещаетесь по файлу и выполняете такие действия как поиск текста, удаление текста, изменение текста и так далее. Обычно запуск редактора происходит в режиме команд.

### Режим вставки

В режиме вставки вы набираете текст согласно позиции курсора. Чтобы вернуться в режим команд, нажмите **Esc** (Escape) клавишу.

Эти два режима определяют поведение редактора. Vi датирован временем, когда не все терминальные клавиатуры содержали клавиши перемещения курсора, поэтому вся работа может быть выполнена в vi с помощью обычных клавиш печатной машинки, а также паре специальных клавиш как **Esc** и **Insert**. Однако вы можете настроить vi на использование дополнительных клавиш, если они доступны; большинство клавиш клавиатуры выполняют какую-либо работу в vi. Из-за своего прошлого и медленной работы ранних терминальных соединений, vi заслужил хорошую репутацию за счет использования коротких и непонятных команд.

## Выход из vi

Одну из первых вещей, которой я хотел бы выучить в новом редакторе, это как осуществлять выход из программы до того, как начать работать. Следующие способы выхода vi включают сохранение или отмену изменений или перезапуск редактирования файла. Если команды не работают, то возможно, вы находитесь в режиме вставки, поэтому нажмите **Esc**, чтобы покинуть режим вставки и перейти в режим команд.

**:q!**

Выход с отменой всех изменений в файле. Это часто используемая команда, чтобы вернуть все в первоначальный вид.

**:w!**

Записать файл (в независимости от того было ли модифицировано содержимое или нет). Попытка перезаписать существующие файлы или файлы только для чтения или другие не записываемые файлы. Вы можете определить имя файла в качестве параметра, и этот файл будет записан, а не тот с которым вы начали работу. В общем безопаснее пропускать !, кроме случаев, когда вы знаете, что делаете.

**ZZ**

Записать файл, если он был изменен. Затем произвести выход. Эта команда часто применяется для нормального выхода из vi.

**:e!**

Редактировать текущую копию файла на диске. Команда перезагрузит файл, отменив созданные вами изменения. Вы также можете использовать команду, если копия на диске была изменена по какой-либо причине и вам требуется последняя ее версия.

**:**

Запустить команду интерпретатора. Наберите команду и нажмите **Enter**. Когда команда завершится, вы увидите ее вывод и приглашение вернуться в редактор vi.

Замечания:

1. Когда вы наберете двоеточие (:), то курсор переместится вниз экрана, где вы можете набирать команду и параметры.
2. Если вы пропустите восклицательный знак в описанных выше командах, то можете

получить сообщение об ошибке, как например, о том, что изменения не были сохранены или файл не может быть записан (например, вы редактируете файл только для чтения).

3. У команды : есть длинные формы (:quit, :write, :edit), но они используются редко.

## Перемещение

Следующие команды используются для перемещения по файлу:

<b>h</b>	Перейти на один символ влево на текущей строке
<b>j</b>	Перейти на следующую строку
<b>k</b>	Перейти на предыдущую строку
<b>l</b>	Сдвинуться на один знак вправо в текущей строке
<b>w</b>	Перейти к следующему слову на текущей строке
<b>e</b>	Перейти на предыдущее слово в текущей строке
<b>b</b>	Перейти в начало предыдущего слова на текущей строке
<b>Ctrl-f</b>	Пролистнуть страницу вперед
<b>Ctrl-b</b>	Пролистнуть страницу назад

Если вы наберете число перед этими командами, то команда будет исполнена определенное число раз. Это число называется *счетчиком повторений* или просто *счетчиком*. Например, 5h осуществит переход влево на пять символов. Вы можете использовать счетчики повторений со многими командами vi.

## Переход по строкам

Следующие команды используются для перехода к определенным строкам вашего файла:

<b>G</b>	Перейти к определенной строке вашего файла. Например, 3G переходит к строке 3. Без параметров, G переходит к последней строке файла.
<b>H</b>	Переходит к строке, отстоящей вниз относительно верхнего края экрана. Например, 3H осуществляет переход к третьей строке сверху относительно текущего экрана.
<b>L</b>	Аналог H, но переход осуществляется относительно нижней части экрана. Так осуществляет на вторую строку относительно нижней части экрана.

## Поиск

Вы можете осуществлять поиск в файле с помощью регулярных выражений:

<b>/</b>	Используйте / и регулярное выражение для поиска вперед по файлу.
<b>?</b>	Используйте ? и регулярное выражение для поиска по файлу назад.
<b>n</b>	Используйте n, чтобы повторить последний поиск в любом из направлений.

Вы можете предварять все вышеперечисленные команды числом, означающим счетчик повторений. Так 3/x найдет третье вхождение x относительно текущей позиции, так как и /x за которой следует команда 2n.

## Модификация текста

Используйте следующие команды, если вам надо вставить, удалить или изменить текст:

**i**

Перейти в режим вставки в текущей позиции. Наберите свой текст и нажмите **Esc**, чтобы вернуться в режим команд. Используйте I, чтобы начать вставку в начале текущей строки.

**a**

Войти в режим вставки после символа в текущей позиции. Наберите свой текст и нажмите **Esc**, чтобы вернуться в режим команд. Используйте A, чтобы осуществить вставку в конец текущей строки.

**c**

Используйте c, чтобы изменить текущий символ и перейти в режим вставки, чтобы набрать замещаемые символы.

**o**

Вставить новую строку сразу за текущей строкой. Используйте O, чтобы вставить новую строку сразу над текущей строкой.

**cw**

Удалить остаток текущего слова, войти в режим вставки и заменить его. Используйте счетчик повторений, чтобы заменить несколько слов. Используйте c\$, чтобы заменить слова до конца строки.

**dw**

Тоже, что и cw (и c\$) выше, только вход в режим вставки не осуществляется.

**dd**

Удалить текущую строку. Используйте счетчик, чтобы удалить несколько строк.

**x**

Удалить символ в позиции курсора. Используйте счетчик, чтобы удалить несколько символов.

**p**

Вставить последний удаленный текст после текущего символа. Используйте P, чтобы вставить его до текущего символа.

**xp**

Комбинация x и p. производит замену символа в позиции курсора и символа справа от него.

## Заключение

Мы собрались добавить строку в файл count1.sh. Чтобы сохранить оригинал и сохранить модификацию в count2.sh, мы можем использовать команды vi, после того как открыли файл в vi. Заметим, что <Esc> означает нажать Esc клавишу.

## Листинг 116. Команды редактора для добавления строки в count1.sh

```
1G
0
sleep 20<Esc>
:w! count2.sh
:q
```

Просто, когда знаешь как.

В следующем руководстве этой серии рассматривается Тема 104 об Устройствах, файловых систем Linux и Структуре файловой системы (FHS).

| [предыдущая](#) | [следующая](#)

## Ресурсы

### Научиться

- Просмотрите весь список [руководств, рекомендованных для сдачи экзамена LPIC](#) на developerWorks, чтобы узнать об основах Linux и подготовиться к сертификации на системного администратора.
- Возможно, вам пригодится [версия этого руководства на английском языке](#).
- В разделе [Программа LPIC](#) вы можете найти списки заданий, вопросы и подробные требования для каждого из трех уровней сертификации на системного администратора Linux в Linux Professional Institute.
- В " [Основы для начинающих разработчиков Linux](#)" (developerWorks, Март 2005), вы изучите, как открывать окно терминала или командной строки и многое другое.
- В серии из трех частей "[Sed в примерах](#)" (developerWorks, Октябрь и Ноябрь 2000), Даниэл Роббинс покажет вам, как использовать мощный редактор (но о котором часто забывают) потоков UNIX, sed. Sed идеальный, чрезвычайно мощный инструмент для обработки множества файлов или создания сценариев для модификации существующих файлов.
- [Потоковый редактор sed](#) -- это полезный сайт, поддерживаемый Эриком Пементом, содержит множество ссылок на sed, FAQ по sed, а также набор полезных команд sed.
- В нашем руководстве по использованию vi, "[Введение в vi -- метод шпаргалки](#)" (developerWorks, Декабрь 2000), Даниэл Роббинс покажет вам, как использовать редактор vi для редактирования текста, научит пользоваться режимом вставки, копировать и вставлять текст, а также использовать важные расширения vim, как визуальный режим и многооконное редактирование.
- Проект документации Linux [Linux Documentation Project](#) содержит большое число полезной документации, в частности различные HOWTO.
- На странице [Linux Man Page Howto](#) вы узнаете, как работают страницы помощи man.
- Посетите домашнюю страницу проекта иерархии файловой системы [Filesystem Hierarchy Standard](#) (FHS).
- Посетите домашнюю страницу [LSB](#), чтобы узнать о Linux Standard Base (LSB), проекте Free Standards Group (FSG), направленном на разработку стандартной двоичной операционной среды.
- [Введение в теорию автоматов, языков и вычислений \(2 издание\)](#) (Addison-Wesley, 2001) является хорошим источником информации по регулярным выражениям и конечным автоматам.
- [Изучаем регулярные выражения, 2 издание](#) (O'Reilly Media, Inc., 2002) книга раскрывает использование регулярных выражений в grep, sed, и других средах программирования.
- [LPIC Linux Certification in a Nutshell](#) (O'Reilly, 2001) и [LPIC I Exam Cram 2: Linux](#)

*Professional Institute Certification Exams 101 and 102 (Exam Cram 2)* (Que, 2004) -- это справочники для тех, кто предпочитает книги.

- Найдите больше [руководств для разработчиков Linux](#) в разделе [Linux developerWorks](#).

## Получить продукты и технологии

- Создайте свой следующий проект на Linux с помощью [пробных программ IBM](#), доступных для прямого скачивания с developerWorks.

## Обсудить

- [Примите участие в обсуждении материала на форуме](#).
- Станьте частью сообщества developerWorks и участвуйте в [блогах developerWorks](#).

# Подготовка к экзамену LPI 101: Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard

Младший уровень администрирования (LPIC-1). Тема 104.

[Ян Шилдс](#), Старший программист, EMC

**Описание:** В данном руководстве Иэн Шилдз продолжает готовить вас к сдаче экзамена 101 на администратора младшего уровня (LPIC-1) Linux Professional Institute. В этом четвертом из пяти руководств Иэн знакомит вас с устройствами, файловыми системами Linux и стандартом Filesystem Hierarchy Standard. По окончании изучения этого руководства вы будете уметь создавать и форматировать разделы в различных файловых системах Linux, управлять ими и обслуживать их

[Больше статей из этой серии](#)

**Дата:** 16.04.2006 (Опубликовано: 24.01.2007)

**Уровень сложности:** средний

## Раздел 1. Предварительная информация

Узнайте, чему эти руководства могут вас обучить, и как вы можете извлечь из них максимум пользы.

Об этой серии

Профессиональный институт Linux ([Linux Professional Institute](#), LPI) сертифицирует системных администраторов Linux по двум уровням: *младший уровень* (также называемый первым сертификационным уровнем) и *средний уровень* (также называемый вторым сертификационным уровнем). Чтобы получить первый сертификационный уровень, необходимо сдать экзамены 101 и 102; чтобы получить второй сертификационный уровень необходимо сдать экзамены 201 и 202.

DeveloperWorks предлагает пособия для помощи в подготовке к каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем, и по каждой теме существует соответствующее руководство от DeveloperWorks для самостоятельного изучения. К экзамену LPI 101 относятся следующие пять тем и соответствующие руководства от DeveloperWorks:

Таблица 1. Экзамен LPI 101: Руководства и темы

Экзамен LPI 101, тема	Руководство DeveloperWorks	Краткое содержание руководства
Тема 101	<a href="#">Подготовка к экзамену LPI 101 (тема 101): Аппаратные средства и архитектура</a>	Научиться настраивать аппаратное обеспечение в Linux. По окончании изучения этого руководства вы будете знать, как Linux настраивает аппаратное обеспечение современных ПК, и что делать, если возникают проблемы.
Тема 102	<a href="#">Подготовка к экзамену LPI 101: Установка Linux и управление системой</a>	Ознакомление с установкой Linux и управлением системой. По окончании изучения этого руководства вы будете знать, как Linux использует разделы диска, как происходит загрузка Linux и как

		устанавливать пакеты программного обеспечения и управлять ими.
Тема 103	<a href="#"><u>Подготовка к экзамену LPI 101: Система команд GNU и UNIX</u></a>	Введение в наиболее распространенные команды GNU и UNIX. По окончании изучения этого руководства вы будете знать, как использовать команды программной оболочки bash, включая команды обработки текста и фильтрации, как искать файлы и каталоги и как управлять обработкой.
Тема 104	Подготовка к экзамену LPI 104: Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard.	(Данное руководство). Как создавать файловые системы на разделах диска, как предоставлять пользователям доступ к ним, управлять доступом к файлам и ограничениями для пользователей и как восстанавливать файловые системы в случае необходимости. Также изучаются жесткие и символьические ссылки, нахождение файлов в файловой системе и их оптимальное расположение. Подробнее см. <a href="#"><u>цели</u></a> ниже.
Тема 110	<a href="#"><u>Подготовка к экзамену LPI 110: Система X WINDOW</u></a>	Изучение системы X WINDOW в Linux. По окончании изучения этого руководства вы будете знать, как устанавливать и обслуживать систему X WINDOW. Это руководство охватывает оба основных пакета X на Linux: XFree86 и X.Org.

Чтобы сдать экзамены 101 и 102 (и получить первый сертификационный уровень), вы должны уметь:

- работать с командной строкой Linux;
- выполнять элементарные задачи обслуживания: помогать пользователям, добавлять пользователей в систему, выполнять резервное копирование и восстанавливать, выключать и перезагружать систему;
- устанавливать и конфигурировать рабочую станцию (включая X Window), и подключать ее к локальной сети, подключать автономный компьютер к Интернет.

Для продолжения подготовки к первому сертификационному уровню см. [руководства developerWorks LPI к экзамену 101](#). Узнайте подробнее о [полном наборе руководств developerWorks LPI](#).

Linux Professional Institute не поддерживает материалы и методики для подготовки к экзаменам, предлагаемые третьими сторонами. За более подробной информацией обращайтесь по электронной почте [info@lpi.org](mailto:info@lpi.org).

## О данном руководстве

Перед вами руководство «Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard», четвертое из пяти руководств, предназначенных для подготовки к экзамену LPI 101. В этом руководстве вы научитесь создавать разделы диска и управлять ими. Также вы узнаете о стандарте *Filesystem Hierarchy Standard (FHS)*, который содержит рекомендации о том, где могут находиться различные типы данных и где они должны храниться в Linux-системе.

Руководство составлено в соответствии с целями LPI для этой темы. Ориентировочно по целям с большей значимостью на экзамене может быть задано больше вопросов.

*Таблица 2. Устройства, файловые системы Linux, стандарт Filesystem Hierarchy Standard:*

*цели экзамена, охватываемые данным руководством.*

<b>Цель экзамена</b>	<b>Уровень значимости цели</b>	<b>Краткое содержание цели</b>
1.104.1 <u>Создание разделов диска и файловых систем</u>	Уровень 3	Изучить конфигурирование разделов диска и создание файловых систем на носителях, в частности, на жестких дисках, научиться использовать различные <code>mkfs</code> -команды для создания файловых систем, включая <code>ext2</code> , <code>ext3</code> , <code>reiserfs</code> , <code>yfat</code> и <code>xfs</code>
1.104.2 <u>Поддержание целостности файловых систем</u>	Уровень 3	Освоение проверки целостности файловых систем, контроля свободного пространства и <code>inodes</code> , устранение элементарных проблем в файловых системах. Изучение обслуживания стандартных файловых систем и журналируемых файловых систем.
1.104.3 <u>Создание и удаление файловых систем</u>	Уровень 3	Изучение монтирования и размонтирования файловых систем вручную. Также изучение настройки монтирования файловых систем при начальном запуске и настройки файловых систем на съемных носителях, например, для накопителей на магнитной ленте, дискет, CD-дисков так, чтобы их мог монтировать и размонтировать обычный пользователь.
1.104.4 <u>Управление дисковыми квотами</u>	Уровень 5	Изучение управления квотами для пользователей, включая установление квот в файловой системе, редактирование, контроль квот и генерацию отчетов о пользовательских квотах.
1.104.5 <u>Использование полномочий для управления доступом к файлам</u>	Уровень 5	Изучение управления доступом к файлам с использованием полномочий, включая права доступа как к обычным и специальным файлам, так и к каталогам. Также изучение таких средств управления доступом, как <code>suid</code> , <code>sgid</code> и <code>sticky bit</code> ; использование группового поля для предоставления доступа рабочим группам; флаг неизменяемости и используемый по умолчанию режим создания файлов.
1.104.6 <u>Управление владением файлами</u>	Уровень 1	Изучение управления владения файлами для пользователей и групп, включая изменение пользователя и группы-владельца файла, а также группы-владельца файла по умолчанию для новых файлов.
1.104.7 <u>Создание и изменение жестких и символьических ссылок</u>	Уровень 1	Изучение создания и управления жесткими и символьическими ссылками для файлов, включая создание и идентификацию ссылок. Изучение копирования файлов с помощью ссылок и использование связанных файлов для поддержки задач системного администрирования.

1.104.8

Уровень 5

Нахождение  
системных файлов  
и корректное  
размещение  
файлов

В начало

Изучение стандарта Filesystem Hierarchy Standard, включая типовое размещение файлов и классификации каталогов. Нахождение файлов и команд в системе Linux.

## **Необходимые условия**

Чтобы извлечь максимум пользы из этого руководства, необходимо иметь базовые знания о Linux и рабочую Linux-систему, где вы сможете опробовать команды, описанные в этом руководстве.

Это руководство является продолжением первых трех пособий из этой серии, поэтому сначала вы можете повторить [руководства к темам 101, 102 и 103](#).

Различные версии программ могут иметь различный формат вывода, поэтому результаты могут отличаться от листингов и рисунков, представленных в данном руководстве.

## **Раздел 2. Создание разделов и файловых систем**

Этот раздел охватывает материалы темы 1.104 для экзамена 101 на младший уровень администрирования. Тема имеет 3 уровень значимости.

В этом разделе вы научитесь:

- конфигурировать разделы диска;
- создавать файловые системы на жестких дисках и других носителях;
- использовать `mkfs`-команды для настройки ext2, ext3, reiserfs, vfat и xfs- разделов

Сначала краткий обзор. В руководстве к теме 101 "[Подготовка к экзамену LPI 101 \(тема 101\): Устройства и архитектура](#)" вы изучили IDE и SCSI диски, такие как /dev/hda и /dev/sdb и их разделы, в частности, /dev/hda1, /dev/sda5 и /dev/sda1.

В руководстве к теме 102 "[Подготовка к экзамену LPI 101 \(тема 102\): Установка Linux и управление системой](#)" вы более подробно изучили дисковые разделы, включая *первичные, расширенные и логические*. Также вы узнали о том, что файловые системы Linux содержат *файлы*, которые размещаются на диске или другом *блочном устройстве хранения в каталогах*. Как и во многих других системах, каталоги в Linux могут содержать другие каталоги, называемые *подкаталогами*. В указанном руководстве также подробно рассказывалось о том, чем нужно руководствоваться при создании дисковых разделов.

В этом разделе мы повторяем сведения о блочных устройствах и дисковых разделах, а затем рассказываем о команде `fdisk`, используемой для создания, редактирования или удаления разделов на блочных устройствах. Также в нем рассматриваются различные формы команды `fdisk` (`mkfs` означает *make filesystem*); эти команды используются для форматирования разделов с определенной файловой системой.

**Замечание:** В дополнение к средствам и файловым системам, требующимся для экзаменов LPI, вы можете столкнуться с другими инструментами и файловыми системами. Краткий обзор некоторых других возможностей см. в разделе [Другие инструменты и файловые системы](#).

### **Блочные устройства и разделы**

Кратко рассмотрим блочные устройства и разделы. За более подробной информацией вернитесь к руководствам по темам [101](#) и [102](#).

## Блочные устройства

Блочное устройство представляет собой уровень абстракции, описывающий любое устройство хранения информации, которое может быть разбито на блоки определенного размера; доступ к каждому блоку осуществляется независимо от доступа к другим блокам. Такой доступ часто называют *произвольным доступом*.

Абстрагированное представление устройств в виде блоков фиксированного размера с произвольным доступом позволяет программам использовать их независимо от того, является ли устройство жестким диском, дискетой, CD- диском, сетевым диском или каким-либо другим устройством, например файловой системой в оперативной памяти.

Примерами блочных устройств могут быть первый жесткий диск (IDE) (`/dev/hda`) или второй SCSI-диск (`/dev/sdb`). Для просмотра каталога `/dev` используйте команду `ls -l`. Первый символ **b** в строке указывает на **блочное** устройство: флоппи- или CD-дисковод, IDE- или SCSI-диск; а **c** – на **символьное** устройство, например, накопитель на магнитной ленте или терминал. См. примеры в листинге 1.

### Листинг 1. Блочные и символьные устройства Linux

```
[ian@lyrebird ian]$ ls -l /dev/fd0 /dev/hda /dev/sdb /dev/st0 /dev/tty0
brw-rw---- 1 ian      floppy     2,   0 Jun 24 2004 /dev/fd0
brw-rw---- 1 root     disk       3,   0 Jun 24 2004 /dev/hda
brw-rw---- 1 root     disk       8,  16 Jun 24 2004 /dev/sdb
crw-rw---- 1 root     disk      9,   0 Jun 24 2004 /dev/st0
crw--w---- 1 root     root      4,   0 Jun 24 2004 /dev/tty0
```

## Разделы

Для некоторых блочных устройств, таких как дискеты, CD и DVD- диски, принято использовать одну файловую систему на всем носителе. Однако на жестких дисках больших объемов и даже на небольших USB- накопителях доступное пространство принято делить или разбивать на несколько *разделов*.

Разделы могут отличаться по объему, на каждом из них может быть своя файловая система, так что один диск может использоваться для различных целей, включая использование его несколькими операционными системами. Например, я использую тестовые системы под несколькими различными дистрибутивами Linux, а также иногда под Windows®, и все они используют один или два общих жестких диска.

Из руководств 101 и 102 вы помните, что жесткий диск имеет *геометрию*, определяемую в терминах цилиндров, головок и секторов. Даже несмотря на то, что современные диски используют *логическую адресацию блоков* (LBA), которая в значительной степени маскирует геометрию диска, основной единицей размещения для разделов диска остается цилиндр.

### Вывод информации о разделах

Информация о разделах диска хранится в *таблице разделов*. Таблица разделов содержит информацию о начале и окончании каждого раздела, информацию о его *типе* и о том, является ли он загрузочным или нет. Чтобы создать или удалить раздел, нужно отредактировать таблицу разделов, используя специальную программу. Для экзамена LPI вам необходимо знать программу `fdisk`, описанную здесь, хотя существуют и другие инструменты.

Для просмотра разделов используется команда `fdisk` с опцией `-l`. Если вы хотите просмотреть разделы для конкретного диска, добавьте имя устройства, например `/dev/hda`. Заметьте, что инструменты для разбиения на разделы требуют административных прав

доступа. Листинг 2 показывает разделы на одном из моих жестких дисков.

## Листинг 2. Просмотр разделов диска с помощью команды fdisk

```
[root@lyrebird root]# fdisk -l /dev/hda
```

```
Disk /dev/hda: 160.0 GB, 160041885696 bytes  
255 heads, 63 sectors/track, 19457 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	2078	16691503+	7	HPFS/NTFS
/dev/hda2		2079	3295	9775552+	c	Win95 FAT32 (LBA)
/dev/hda3		3296	3422	1020127+	83	Linux
/dev/hda4		3423	19457	128801137+	f	Win95 Ext'd (LBA)
/dev/hda5		3423	3684	2104483+	82	Linux swap
/dev/hda6		3685	6234	20482843+	83	Linux
/dev/hda7		6235	7605	11012526	83	Linux
/dev/hda8		7606	9645	16386268+	83	Linux
/dev/hda9		9646	12111	19808113+	83	Linux
/dev/hda10		12112	15680	28667961	83	Linux
/dev/hda11		15681	19457	30338721	83	Linux

### Замечания:

1. В заголовке отражена информация об объеме диска и его геометрии. Большинство больших дисков, использующих LBA, имеют 255 головок на цилиндр и 63 сектора в дорожке, что составляет 16065 секторов или 8225280 байт на цилиндр.
2. В данном примере первый раздел (/dev/hda1) помечен как *загрузочный* (или *активный*). Как вы видели в руководстве к теме 102, это обеспечивает загрузку раздела с помощью стандартной загрузочной записи DOS. Этот признак не имеет смысла в LILO или GRUB- загрузчиках.
3. Столбцы *Start* и *End* показывают начальный и конечный цилиндры для каждого раздела. Они не должны перекрываться, а должны следовать строго друг за другом без промежутков.
4. Столбец *Blocks* показывает число блоков размером 1 килобайт (1024 байт) в разделе. Максимальное количество блоков в разделе, следовательно, равняется половине произведения числа цилиндров (*End* + 1 - *Start*) на число секторов в цилиндре. Знак + в конце означает, что используются не все секторы раздела.
5. Поле *Id* указывает на предполагаемое использование раздела. Тип 82 – файл подкачки, 83 – раздел для хранения информации. Существует около 100 различных типов томов. Данный диск используется несколькими операционными системами, в том числе Windows/XP, поэтому на нем есть разделы с файловой системой NTFS (и FAT32).

## Создание разделов с помощью команды fdisk

Только что вы узнали, как просмотреть данные о разделах диска с помощью [fdisk](#). Эта команда также позволяет редактировать таблицу разделов с целью создания и удаления разделов.

### Предупреждения

Прежде чем изменять разделы, необходимо запомнить несколько важных моментов. Если не следовать этим рекомендациям, вы рискуете **потерять существующую информацию**.

1. **Не изменяйте разделы, которые используются в настоящий момент.** Составьте

план действий и четко его придерживайтесь.

2. **Знайте возможности вашего инструментального средства.** `Fdisk` не выполняет изменений без вашего подтверждения. Другие инструменты, как, например `parted`, могут применять изменения сразу.
3. **Прежде чем начинать, создайте резервную копию важной информации**, так как любая операция может привести к потере данных.
4. Инструменты для создания разделов диска оперируют таблицей разделов. Если ваш инструмент не выполняет также операций, связанных с перемещением, изменением размера, форматированием или другими способами изменения дискового пространства, ваши данные не будут повреждены. Если вы случайно допустите ошибку, как можно скорее прервите работу и обратитесь за помощью. Возможно, вы еще сможете восстановить разделы и данные.

### Запуск fdisk

Для запуска `fdisk` в интерактивном режиме просто задайте в качестве параметра имя диска, например, `/dev/had` или `/dev/sdb`. В следующем примере показана загрузка с рабочего CD-диска Knoppix. Если вы обладаете правами администратора, то получите результат, аналогичный листингу 3.

### Листинг 3. Интерактивный запуск fdisk

```
root@ttyp1[knoppix]# fdisk /dev/hda
```

```
The number of cylinders for this disk is set to 14593.  
There is nothing wrong with that, but this is larger than 1024,  
and could in certain setups cause problems with:  
1) software that runs at boot time (e.g., old versions of LILO)  
2) booting and partitioning software from other OSs  
(e.g., DOS FDISK, OS/2 FDISK)
```

Command (`m` for help):

Современные диски содержат более 1024 цилиндров, поэтому обычно вы будете получать предупреждение, как в листинге 3. Нажмите `m`, чтобы получить список возможных однобуквенных команд, показанный в листинге 4.

### Листинг 4. Помощь в fdisk

Command	action
a	toggle a bootable flag
b	edit bsd disklabel
c	toggle the dos compatibility flag
d	delete a partition
l	list known partition types
m	print this menu
n	add a new partition
o	create a new empty DOS partition table
p	print the partition table
q	quit without saving changes
s	create a new empty Sun disklabel
t	change a partition's system id
u	change display/entry units
v	verify the partition table
w	write table to disk and exit

```
x extra functionality (experts only)
```

Command (m for help):

Для просмотра разделов диска нажмите p; результат - в листинге 5.

### Листинг 5. Просмотр существующей таблицы разделов

```
Disk /dev/hda: 120.0 GB, 120034123776 bytes  
255 heads, 63 sectors/track, 14593 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	2611	20972826	7	HPFS/NTFS

Command (m for help):

Объем данного диска 120 ГБ, имеется раздел под Windows XP, занимающий около 20 ГБ. Это первичный раздел, помеченный как загрузочный, что типично для Windows-систем.

### Формирование структуры диска для рабочей станции

Теперь используем часть свободного пространства для организации рабочей станции со следующими дополнительными разделами. На практике вы вряд ли будете использовать такое количество разных типов файловых систем, но здесь мы сделаем это для примера.

1. Еще один первичный раздел для наших загрузочных файлов. Он будет монтироваться как /boot и содержит файлы ядра и исходных RAM-дисков. Если вы используете загрузчик GRUB, то его файлы тоже будут располагаться здесь. Из руководства к теме 102 следует, что для этого необходимо около 100 Мбайт. Из листинга 5 видно, что объем цилиндра примерно 8 Мбайт, поэтому загрузочный раздел /boot займет 13 цилиндров. Это будет /dev/hda2.
2. Создадим расширенный раздел для размещения логических разделов, занимающий остальное свободное пространство. Это будет /dev/hda3.
3. Создадим раздел подкачки размером 500 Мбайт как /dev/hda5. Он займет 64 цилиндра.
4. Создадим логический раздел объемом около 20 ГБ для нашей Linux- системы. Это будет /dev/hda6.
5. Создадим отдельный раздел для данных пользователя размером 10 ГБ. В дальнейшем он будет монтироваться как /home, а пока это будет просто /dev/hda7.
6. И наконец, создадим маленький (2 ГБ) раздел для обмена данными между системами Linux и Windows. В дальнейшем на нем будет использоваться файловая система FAT32 (или vfat). Это будет /dev/hda8.

### Создание разделов

Начнем с использования команды n для создания нового раздела; см. листинг 6.

### Листинг 6. Создание первого раздела

```
Command (m for help): n  
Command action  
  e   extended  
  p   primary partition (1-4)  
p
```

```
Partition number (1-4): 2
First cylinder (2612-14593, default 2612):
Using default value 2612
Last cylinder or +size or +sizeM or +sizeK (2612-14593, default 14593): 2624
```

```
Command (m for help): p
```

```
Disk /dev/hda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	2611	20972826	7	HPFS/NTFS
/dev/hda2		2612	2624	104422+	83	Linux

```
Command (m for help):
```

Мы берем значение по умолчанию для первого цилиндра и задаем значение 2624 для последнего цилиндра, в результате получаем раздел из 13 цилиндров. Из листинга 6 видно, что наш раздел в действительности занимает примерно 100 Мбайт. Поскольку это первичный раздел, он должен иметь номер от 1 до 4. Рекомендуется назначать номера разделов последовательно; если этого не делать, некоторые инструменты выдают предупредительные сообщения.

Заметьте также, что наш новый раздел будет иметь тип 83, то есть раздел для хранения данных в Linux. Это можно рассматривать как указатель операционной системы, которую планируется использовать на этом разделе. Дальнейшее использование должно быть согласовано с этим, но в данный момент мы даже не будем форматировать раздел, не говоря уж о том, чтобы размещать на нем какую-либо информацию.

Теперь создадим расширенный раздел, который будет содержать логические разделы диска. Присвоим этому разделу номер 3 (/dev/hda3). Процесс и результат показан в листинге 7.

Заметьте, что тип раздела назначается автоматически.

### Листинг 7. Создание расширенного раздела

```
Command (m for help): n
Command action
  e  extended
  p  primary partition (1-4)
e
Partition number (1-4): 3
First cylinder (2625-14593, default 2625):
Using default value 2625
Last cylinder or +size or +sizeM or +sizeK (2625-14593, default 14593):
Using default value 14593
```

```
Command (m for help): p
```

```
Disk /dev/hda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	2611	20972826	7	HPFS/NTFS
/dev/hda2		2612	2624	104422+	83	Linux
/dev/hda3		2625	14593	96140992+	5	Extended

```
Command (m for help):
```

Теперь перейдем к разделу файла подкачки как логического раздела внутри нашего расширенного раздела. Мы задаем для конечного цилиндра значение +64 (цилиндра) вместо того, чтобы считать самим. Отметьте, что при этом мы используем команду **t**, чтобы задать для вновь создаваемого раздела тип 82 (раздел подкачки Linux). Иначе это будет раздел с типом 83 (данные Linux).

#### Листинг 8. Создание раздела подкачки

```
Command (m for help): n
Command action
  l  logical (5 or over)
  p  primary partition (1-4)
l
First cylinder (2625-14593, default 2625):
Using default value 2625
Last cylinder or +size or +sizeM or +sizeK (2625-14593, default 14593): +64

Command (m for help): t
Partition number (1-5): 5
Hex code (type L to list codes): 82
Changed system type of partition 5 to 82 (Linux swap)

Command (m for help): p

Disk /dev/hda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

      Device Boot      Start        End    Blocks   Id  System
/dev/hda1  *           1       2611    20972826    7  HPFS/NTFS
/dev/hda2            2612       2624        104422+   83  Linux
/dev/hda3            2625      14593    96140992+   5  Extended
/dev/hda5            2625       2689        522081   82  Linux swap

Command (m for help):
```

Теперь определим основной раздел для Linux и раздел /home. Для этого просто зададим объемы +20480 Мбайт и +10240 Мбайт, т.е. 20 ГБ и 10 ГБ соответственно. Предоставим **fdisk** самостоятельно подсчитать число цилиндров. Результаты представлены в листинге 9.

#### Листинг 9. Создание основного раздела Linux

```
Command (m for help): n
Command action
  l  logical (5 or over)
  p  primary partition (1-4)
l
First cylinder (2690-14593, default 2690):
Using default value 2690
Last cylinder or +size or +sizeM or +sizeK (2690-14593, default 14593): +20480M

Command (m for help): n
Command action
  l  logical (5 or over)
  p  primary partition (1-4)
l
First cylinder (5181-14593, default 5181):
```

```

Using default value 5181
Last cylinder or +size or +sizeM or +sizeK (5181-14593, default 14593): +10240M

Command (m for help): p

Disk /dev/hda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

      Device Boot   Start     End   Blocks   Id  System
/dev/hda1  *        1    2611  20972826    7  HPFS/NTFS
/dev/hda2          2612    2624      104422+   83  Linux
/dev/hda3          2625   14593  96140992+    5  Extended
/dev/hda5          2625    2689      522081    82  Linux swap
/dev/hda6          2690    5180  20008926    83  Linux
/dev/hda7          5181    6426      10008463+   83  Linux

Command (m for help):

```

Последний раздел – раздел с файловой системой FAT32. Выполним уже знакомые действия для создания раздела /dev/hda9, определив объем как +2048 Мбайт, а затем изменим тип раздела на **b** (для FAT32 в версии Windows 95). Затем сохраним изменения.

### Сохранение таблицы разделов

До настоящего времени мы редактировали таблицу разделов в оперативной памяти. Можно использовать команду **q** для выхода без сохранения изменений. Если что-то выполнено не так, как нужно, можно использовать **d** для удаления одного или более разделов и переопределить их заново. Если все сделано верно, используем **V** для проверки, а затем **W**, чтобы сохранить новую таблицу разделов и выйти. Смотрите листинг 10. Если вновь запустить **fdisk -l**, увидим, что изменения уже применены в Linux. В отличие от некоторых других операционных систем, для того, чтобы увидеть эти изменения, не всегда необходима перезагрузка. Перезагрузка может потребоваться, например, если раздел /dev/hda3 переназначается в /dev/hda2 из-за, того, что раздел /dev/hda2 был удален. Если перезагрузка необходима, **fdisk** сообщит вам об этом.

### Листинг 10. Сохранение таблицы разделов.

```

Command (m for help): v
127186915 unallocated sectors

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
root@ttyp0[knoppix]# fdisk -l /dev/hda

Disk /dev/hda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

      Device Boot   Start     End   Blocks   Id  System
/dev/hda1  *        1    2611  20972826    7  HPFS/NTFS

```

/dev/hda2	2612	2624	104422+	83	Linux
/dev/hda3	2625	14593	96140992+	5	Extended
/dev/hda5	2625	2689	522081	82	Linux swap
/dev/hda6	2690	5180	20008926	83	Linux
/dev/hda7	5181	6426	10008463+	83	Linux
/dev/hda8	6427	6676	2008093+	b	W95 FAT32

## Подробнее про fdisk

Можно отметить, что мы не изменяли загрузочный раздел. Наш диск по-прежнему имеет главную загрузочную запись Windows и соответственно будет загружаться с первого раздела, который помечен как загрузочный (раздел NTFS в нашем примере).

Ни LILO, ни GRUB не используют флаг загрузочного раздела. Если какой-либо из этих загрузчиков будет установлен в главной загрузочной записи, он может загрузить раздел Windows XP. Также можно установить LILO или GRUB в раздел /boot (/dev/hda2), пометить этот раздел как загрузочный и удалить загрузочный флажок с раздела /dev/hda1. Оставить первоначальную загрузочную запись полезно, если впоследствии на машине вновь будет использоваться только Windows.

Мы рассмотрели один из способов формирования рабочей станции в Linux. Другие возможности описаны далее в руководстве, в разделе [Нахождение и размещение системных файлов](#).

## Типы файловых систем

Linux поддерживает несколько различных типов файловых систем. Каждая имеет свои достоинства, недостатки и отличительные черты. Важное свойство файловой системы – журналирование – позволяет быстро восстановить систему после сбоя. Как правило, журналируемые системы предпочтительнее нежурналируемых, если у вас есть выбор. Ниже приведен краткий обзор типов файловых систем, которые необходимо знать для экзамена LPI. Более подробную информацию см. в разделе [Ресурсы](#)

### Файловая система ext2

Файловая система ext2 (также известная как *вторая расширенная файловая система*) разработана для устранения недостатков в системе Minix, использовавшейся в ранних версиях Linux. Она широко использовалась в Linux в течение длительного времени. Ext2 не журналируется и в значительной степени вытеснена ext3.

### Файловая система ext3

Файловая система ext3 дополняет возможности стандартной ext2 журналированием и поэтому представляет собой эволюционное развитие очень стабильной файловой системы. Она обеспечивает разумную производительность в большинстве ситуаций и продолжает совершенствоваться. Поскольку она представляет собой расширенный вариант системы ext2, есть возможность преобразовывать систему ext2 в ext3 и, в случае необходимости, обратно.

### Файловая система ReiserFS

ReiserFS – это файловая система, основанная на В-дереве, с очень хорошими рабочими характеристиками, особенно для большого числа маленьких файлов. ReiserFS хорошо масштабируется и является журналируемой.

### Файловая система XFS

XFS – журналируемая файловая система. Она имеет ряд эффективных функций и оптимизирована для масштабирования. XFS активно кэширует перемещаемую информацию в оперативной памяти, поэтому при использовании этой системы рекомендуется иметь

источник бесперебойного питания.

### Файловая система раздела подкачки

Пространство для подкачки должно быть отформатировано, но обычно оно не рассматривается как отдельная файловая система.

### Файловая система vfat

Эта файловая система (также известная как *FAT32*) не является журналируемой и имеет множество недостатков по сравнению с файловыми системами, используемыми Linux. Она применяется для обмена данными между системами Windows и Linux, поскольку читается обеими. **Не используйте** эту файловую систему в Linux, за исключением случаев совместного использования данных системами Windows и Linux. Если распаковать архив Linux на диск с системой vfat, вы потеряете права доступа, например на выполнение программ, а также символические ссылки, которые могли храниться в архиве.

Как ext3, так и ReiserFS являются зрелыми файловыми системами и используются по умолчанию в ряде дистрибутивов. Обе они рекомендованы к широкому использованию.

### Создание файловых систем

Для создания файловых систем в Linux используется команда **mkfs**, а для создания раздела подкачки – команда **mkswap**. Команда **mkfs** фактически является интерфейсом доступа к целому ряду команд, специфичных для конкретных файловых систем, например, **mkfs.ext3** для ext3, **mkfs.reiserfs** для ReiserFS.

Поддержка каких файловых систем имеется в вашей системе? Чтобы это выяснить, используйте команду **ls /sbin/mk\***. Пример представлен в листинге 11.

### Листинг 11. Команды для создания файловых систем

```
root@ttyp0[knoppix]# ls /sbin/mk*
/sbin/mkdosfs      /sbin/mkfs.ext2    /sbin/mkfs.msdos      /sbin/mkraid
/sbin/mke2fs       /sbin/mkfs.ext3    /sbin/mkfs.reiserfs   /sbin/mkreiserfs
/sbin/mkfs        /sbin/mkfs.jfs     /sbin/mkfs.vfat      /sbin/mkswap
/sbin/mkfs.cramfs /sbin/mkfs.minix  /sbin/mkfs.xfs
```

Отметьте различные формы некоторых команд. Например, команды mke2fs, mkfs.ext2 и mkfs.ext3 равнозначны, как и mkreiserfs и mkfs.reiserfs.

Существует несколько общих опций для всех **mkfs**-команд. Опции, которые специфичны для создаваемой файловой системы, передаются командам создания в зависимости от типа, определенного параметром **-type**. В наших примерах используется **mkfs -type**, но можно использовать и другие формы с тем же результатом. Например, можно использовать **mkfs -type reiserfs**, **mkreiserfs** или **mkfs.reiserfs**. Для вызова справочных страниц по конкретной файловой системе укажите в качестве имени соответствующую команду **mkfs**, например, **man mkfs.reiserfs**. Многие значения, приведенные в нижеследующих примерах вывода, управляемы опциями для **mkfs**.

### Создание файловой системы ext3

### Листинг 12. Создание файловой системы ext3

```
root@ttyp0[knoppix]# mkfs -t ext3 /dev/hda8
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
```

```
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
251392 inodes, 502023 blocks
25101 blocks (5.00%) reserved for the super user
First data block=0
16 block groups
32768 blocks per group, 32768 fragments per group
15712 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Полезная опция, используемая при создании ext2 и ext3 – опция [-L](#) с именем, которая назначает метку тому. Метку можно использовать при монтировании файловой системы вместо имени устройства; это обеспечивает определенный уровень изоляции в отношении изменений, который необходимо отразить в различных управляющих файлах. Для просмотра и установки метки на существующую систему ext2 или ext3 используется команда [e2label](#). Длина метки ограничена 16 символами.

Следует заметить, что ext3 ведет журнал. Если вы хотите добавить журналирование к существующей системе ext2, используйте команду [tune2fs](#) с опцией [-j](#).

## Создание файловой системы ReiserFS

### Листинг 13. Создание файловой системы ReiserFS

```
.root@ttyp0[knoppix]# mkfs -t reiserfs /dev/hda6
mkfs.reiserfs 3.6.17 (2003 www.namesys.com)
```

A pair of credits:

Many persons came to [www.namesys.com/support.html](http://www.namesys.com/support.html), and got a question answered for \$25, or just gave us a small donation there.

Jeremy Fitzhardinge wrote the teahash.c code for V3. Colin Plumb also contributed to that.

```
Guessing about desired format. Kernel 2.4.26 is running.
Format 3.6 with standard journal
Count of blocks on the device: 5002224
Number of blocks consumed by mkreiserfs formatting process: 8364
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: 72e317d6-8d3a-45e1-bcda-ad7eff2b3b40
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
          ALL DATA WILL BE LOST ON '/dev/hda6'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
```

Tell your friends to use a kernel based on 2.4.18 or later, and especially not a kernel based on 2.4.9, when you use reiserFS. Have fun.

ReiserFS is successfully created on /dev/hda6.

Для задания метки тома используйте **-l** (или опцию **--label** с именем). Для добавления или просмотра метки к существующей системе ReiserFS используется команда **reiserfstune**. Максимальное число символов в метке – 16.

## Создание файловой системы XFS

### Листинг 14. Создание файловой системы XFS

```
root@ttyp0[knoppix]# mkfs -t xfs /dev/hda7
meta-data=/dev/hda7              isize=256    agcount=16, agsize=156382 blks
                                =          sectsz=512
data     =                      bsize=4096   blocks=2502112, imaxpct=25
                                =          sunit=0    swidth=0 blks, unwritten=1
naming   =version 2            bsize=4096
log      =internal log         bsize=4096   blocks=2560, version=1
                                =          sectsz=512  sunit=0 blks
realtime =none                 extsz=65536   blocks=0, rtextents=0
```

Для задания метки тома в системе XFS используется опция **-L** с именем. Для добавления метки к существующей файловой системе XFS используется команда **xfs\_admin** с опцией **-L**. Для просмотра метки используется команда **xfs\_admin** с опцией **-l**. В отличие от ext2, ext3 и ReiserFS максимальное число символов в метке составляет 12.

## Создание файловой системы vfat

### Листинг 15. Создание файловой системы vfat

```
root@ttyp0[knoppix]# mkfs -t vfat /dev/hda8
mkfs.vfat 2.10 (22 Sep 2003)
```

Метка тома в системе FAT32 назначается с помощью опции **-n**. Команда **e2label** отображает или устанавливает метку тома в системе vfat, а также в разделах ext. Длина метки ограничена 16 символами.

## Создание пространства подкачки

### Листинг 16. Создание пространства подкачки

```
root@ttyp0[knoppix]# mkswap /dev/hda5
Setting up swapspace version 1, size = 534605 kB
```

Разделы подкачки, в отличие от обычновенных файловых систем, не монтируются, а активизируются командой **Swapon**. Стартовые сценарии Linux автоматически активизируют разделы подкачки.

## Другие инструменты и файловые системы

Следующие инструменты и файловые системы не входят в цели данного экзамена LPI. Здесь мы приводим очень краткий обзор инструментов и файловых систем, которые вам могут встретиться.

### Инструменты для создания разделов

Многие дистрибутивы Linux содержат команды **cfdisk** и **sfdisk**. Команда **cfdisk** предоставляет более удобный графический интерфейс, чем fdisk, используя библиотеку функций ncurses, как показано на рисунке 1. Команда **sfdisk** предназначена для использования программистами и допускает использование сценариев. Применяйте ее, только если умеете ею пользоваться.

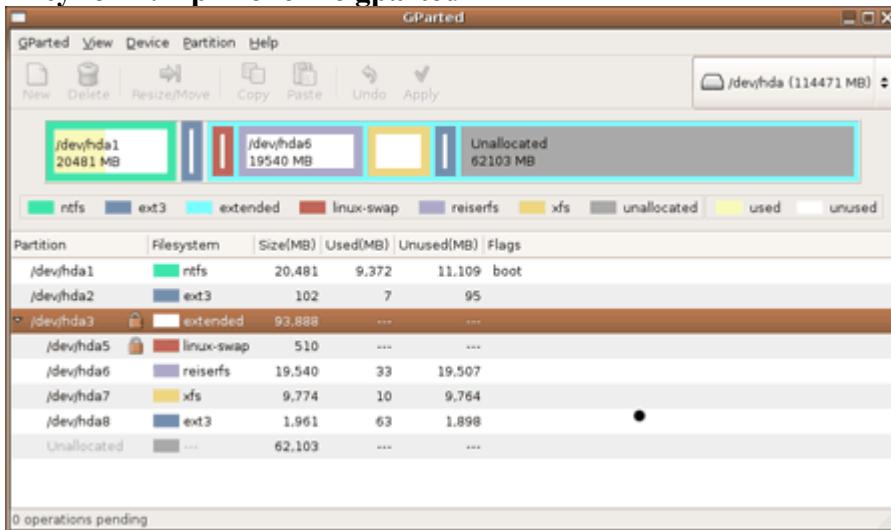
**Рисунок 1. Использование cfdisk**



Другим распространенным инструментом для работы с таблицей разделов является команда **parted**, с помощью которой можно изменять и формировать множество типов разделов, а также создавать и удалять их. Для изменения объема NTFS- раздела вместо команды **parted** используется **ntfsresize**. Команда **qtparted** использует графический интерфейс на базе Qt. Она выполняет как функции **parted**, так и **ntfsresize**.

Команда **gparted** – еще один инструмент с графическим интерфейсом, разработанный для среды GNOME. Она использует библиотеки GTK+GUI, как показано на рисунке 2. (Информацию о **qtparted** и **gparted** см. в разделе [Ресурсы](#)).

**Рисунок 2. Применение gparted**



Имеется также ряд коммерческих инструментов для создания дисковых разделов. Возможно, наиболее известный из них – PartitionMagic, теперь распространяемый Symantec.

Многие дистрибутивы позволяют делить диск на разделы, а иногда также сжимать существующие разделы Windows NTFS или FAT32 в процессе установки. Более точную информацию см. в руководстве по установке конкретного дистрибутива.

### **Диспетчер логических томов**

Диспетчер логических томов (LVM) для Linux позволяет объединять несколько физических устройств хранения в единую группу томов. Например, можно добавить раздел к существующей группе томов, вместо того чтобы искать необходимое для вашей файловой системы непрерывное дисковое пространство.

### **RAID**

RAID (резервированный массив независимых дисков) – это технология, обеспечивающая надежное хранение информации с использованием недорогих дисков, которые гораздо доступнее используемых в системах высшей ценовой категории. Существует несколько типов RAID-массивов. Технология RAID может быть реализована как на аппаратном уровне, так и на программном. Linux поддерживает оба варианта.

### **Другие файловые системы**

Вам также могут встретиться файловые системы, не рассмотренные здесь.

*Journaled File System (JFS)* от IBM, в настоящее время используемая в корпоративных серверах компании IBM, разработана для серверных сред с высокой пропускной способностью. Она реализована для Linux и входит в состав некоторых дистрибутивов. Для создания файловой системы JFS используется команда `mkfs.jfs`.

Существуют и другие файловые системы, например `cramfs`, часто используемая встроенными устройствами.

В следующем разделе рассказывается, как поддерживать целостность файловой системы и что делать при возникновении неполадок.

| [предыдущая](#) |

## **Раздел 3. Целостность файловых систем**

Этот раздел охватывает материалы темы 1.104.2 для экзамена 101 на младший уровень администрирования. Тема имеет третий уровень значимости.

Из этой темы вы узнаете, как:

- контролировать свободное пространство и inodes
- проверять целостность файловых систем
- решать несложные проблемы, возникающие в файловых системах

Рассмотрены как стандартные, так и журналируемые файловые системы. Акцент сделан на системах ext2 и ext3, но также затронуты средства других файловых систем. Большая часть представленного материала относится как к ядру 2.4, так и 2.6. Примеры, приведенные в этом разделе, в основном используют систему Ubuntu 5.10 “Breezy Badger” (версия, основанная на Debian), с ядром 2.6.12, которая была установлена на файловые системы, созданные в предыдущем разделе. Результаты, полученные при использовании других систем, могут отличаться от представленных.

### **Контроль свободного пространства**

И блоки данных, и блоки inode занимают место в файловой системе, поэтому необходимо контролировать используемое пространство, чтобы быть уверенным в наличии свободного места на диске для расширения файловой системы.

## df

Команда **df** выводит информацию о монтированных файловых системах. (Подробнее о монтировании файловых систем – в следующем разделе [Монтирование и размонтирование файловых систем](#)). Если добавить опцию **-T**, к выводу будет добавлен тип файловой системы. Результат выполнения команды **df** в системе Ubuntu, установленной на файловые системы, созданные в предыдущем разделе, показан в листинге 17.

### Листинг 17. Вывод информации об использовании файловых систем

```
ian@pinguino:~$ df -T
Filesystem      Type      1K-blocks      Used   Available  Use% Mounted on
/dev/hda6    reiserfs     20008280    1573976    18434304    8% /
tmpfs        tmpfs       1034188         0    1034188     0% /dev/shm
tmpfs        tmpfs       1034188    12588    1021600     2% /
lib/modules/2.6.12-10-386/volatile
/dev/hda2    ext3        101105     19173     76711    20% /boot
/dev/hda8    vfat        2004156         8    2004148     1% /dos
/dev/hda7    xfs        9998208     3544    9994664     1% /home
/dev/hda1    ntfs       20967416   9594424   11372992    46% /media/hda1
```

Заметьте, что вывод содержит общее число блоков, а также число используемых и свободных блоков. Также указывается файловая система, например, `/dev/hda7`, и ее точка монтирования: `/home` для `/dev/hda7`. Две записи `tmpfs` относятся к файловым системам в виртуальной памяти. Они существуют только в оперативной памяти или пространстве подкачки и создаются в момент монтирования без использования команды **mkfs**. Подробнее о команде `tmpfs` – в части «Общие курсы: расширенное руководство по реализации файловых систем, Часть 3» (см. ссылку в разделе [Ресурсы](#)).

Если необходимо вывести данные об использовании inode, применяется команда **df** с опцией **-i**. Можно исключить вывод данных по определенной файловой системе, используя опцию **-X**, или ограничить информацию определенными типами файловых систем, используя опцию **-t**. При необходимости их можно использовать несколько раз. Примеры представлены в листинге 18.

### Листинг 18. Просмотр inode

```
ian@pinguino:~$ df -i -x tmpfs
Filesystem      Inodes  IUsed  IFree  IUse% Mounted on
/dev/hda6          0      0      0      -   /
/dev/hda2        26208     34    26174     1% /boot
/dev/hda8          0      0      0      -   /dos
/dev/hda7      10008448    176  10008272     1% /home
/dev/hda1        37532   36313    1219    97% /media/hda1
ian@pinguino:~$ df -iT -t vfat -t ext3
Filesystem      Type  Inodes  IUsed  IFree  IUse% Mounted on
/dev/hda2    ext3    26208     34    26174     1% /boot
/dev/hda8    vfat      0      0      0      -   /dos
```

Возможно, вас не удивит то, что для системы FAT32 не отображаются inodes, но неожиданностью может стать то, что и для ReiserFS их тоже нет в выводе. ReiserFS содержит метаданные о файлах и каталогах в объектах stat items. Вследствие того, что в ReiserFS используется сбалансированная древовидная структура, в ней нет заранее определенного

числа inodes, в отличие, например, от файловых систем ext2, ext3 или xfs.

Кроме того, существуют некоторые другие опции команды **df**, используемые для ограничения вывода локальными файловыми системами или для контроля формата вывода. Например, используйте опцию **-H** для вывода результатов в удобном для пользователя формате (например, 1К для 1024), или опцию **-h** (или **--si**) для отражения размеров в десятичном представлении (1К=1000).

Если вы не знаете точно, какая файловая система используется для определенной части вашего дерева каталогов, можно применить команду **df** с указанием пути или даже имени файла в качестве параметра, как показано в листинге 19.

### Листинг 19. Удобочитаемый формат вывода результатов df

```
ian@pinguino:~$ df --si ~ian/index.html
Filesystem      Size  Used  Avail Use% Mounted on
/dev/hda7        11G   3.7M   11G   1% /home
```

## du

Команда **df** выводит информацию только о файловой системе в целом. Иногда необходимо узнать, сколько места занимает каталог `home`, или какой размер раздела потребуется, чтобы разместить каталог `/usr` в отдельной файловой системе. Для решения этих задач используется команда **du**.

Команда **du** выводит информацию о файле (файлах), имена которых заданы в качестве параметров. Если задано имя каталога, то **du** определяет размер всех файлов и подкаталогов этого каталога на всех уровнях вложения. Результат работы команды может быть очень объемным. К счастью, существует опция **-s** для вывода сводной информации по каталогу. Если использовать **du** для получения информации о нескольких каталогах, можно добавить опцию **-c** для вывода суммарных данных. Также можно задавать формат вывода. Для этого применяются опции, аналогичные используемым в команде **df** (**-h**, **-H**, **--si** и т.п.).

Листинг 20 показывает два варианта вывода для моего каталога `home` во вновь установленной системе Ubuntu.

### Листинг 20. Использование du

```
ian@pinguino:~$ du -hc *
0      Desktop
16K    index.html
16K    total
ian@pinguino:~$ du -hs .
3.0M  .
```

Причина различия между результатом команды **du -c \***, насчитавшей 16 КБ, и команды **du -s**, получившей объём 3 МБ, в том, что последняя включает файлы и каталоги, начинающиеся с точки, такие как `.bashrc`, которые не просматриваются первой.

Еще следует отметить, что для использования **du** вы должны иметь права чтения каталогов, к которым вы ее применяете.

Теперь применим **du** для просмотра общего объема, занимаемого каталогом `/usr` и всеми его подкаталогами первого уровня. Результат представлен в листинге 21. Чтобы с гарантией

иметь соответствующие права доступа, используйте полномочия root.

### Листинг 21. Использование du для каталога /usr

```
root@pinguino:~# du -shc /usr/*
66M    /usr/bin
0      /usr/doc
1.3M   /usr/games
742K   /usr/include
0      /usr/info
497M   /usr/lib
0      /usr/local
7.3M   /usr/sbin
578M   /usr/share
0      /usr/src
14M    /usr/X11R6
1.2G   total
```

### Проверка файловых систем

Иногда в системе может произойти сбой или отключиться питание. В этих случаях Linux не может аккуратно размонтировать файловые системы, и они могут оказаться в несогласованном состоянии. Работать с поврежденной файловой системой не следует, поскольку это скорее всего приведет к усугублению имеющихся ошибок.

Основной инструмент для проверки файловых систем - команда **fsck**, которая, аналогично **mkfs**, является интерфейсом доступа к командам проверки различных типов файловых систем. Несколько примеров таких команд приведено в листинге 22.

### Листинг 22. Примеры программ fsck.

```
ian@pinguino:~$ ls /sbin/*fsck*
/sbin/dosfsck      /sbin/fsck.ext3      /sbin/fsck.reiser4    /sbin/jfs_fscklog
/sbin/e2fsck        /sbin/fsck.jfs       /sbin/fsck.reiserfs   /sbin/reiserfsck
/sbin/fsck          /sbin/fsck.minix    /sbin/fsck.vfat
/sbin/fsck.cramfs   /sbin/fsck.msdos   /sbin/fsck.xfs
/sbin/fsck.ext2     /sbin/fsck.nfs     /sbin/jfs_fsck
```

Процесс загрузки системы с помощью команды **fsck** проверяет корневую файловую систему и другие файловые системы, указанные в управляющем файле */etc/fstab*. Если файловая система не была размонтирована корректно, проводится проверка целостности системы. Это определяется значением поля *pass* (или *passno*) (шестое поле записи */etc/fstab*). Файловые системы со значением pass, установленным в ноль, не тестируются во время загрузки. Корневая файловая система имеет значение pass, равное 1, и тестируется первой. Другие файловые системы обычно имеют значение pass от двух и выше, которое указывает, в каком порядке их надо проверять. Несколько операций **fsck** могут выполняться параллельно, поэтому различные файловые системы могут иметь одинаковые значения *pass*, как в нашем примере системы */boot* и */home*.

### Листинг 23. Тестирование системы при загрузке на основании данных fstab.

```
# <file system> <mount point>  <type>  <options>      <dump>  <pass>
proc            /proc           proc    defaults        0        0
/dev/hda6        /               reiserfs defaults        0        1
/dev/hda2        /boot          ext3    defaults        0        2
```

/dev/hda8	/dos	vfat	defaults	0	0
/dev/hda7	/home	xfs	defaults	0	2

Следует отметить, что некоторые журналируемые файловые системы, такие как ReiserFS и xfs, могут иметь значение pass, установленное в 0, поскольку проверку и восстановление файловой системы производит программа журналирования, а не [fsck](#).

### Восстановление файловых систем

Если автоматическая проверка при загрузке не может восстановить согласованность файловой системы, обычно происходит переход в однопользовательскую командную оболочку и выводится сообщение с указаниями по ручному запуску [fsck](#). В системе ext2, которая не журналируется, вам может быть представлена серия вопросов для подтверждения операций по восстановлению файловой системы. Как правило, рекомендуется следовать предложениям [fsck](#) по восстановлению системы, выбирая у (для подтверждения операции). Когда система перезагрузится, проверьте, не пропала ли какая-либо информация или файлы.

Если вы заподозрили порчу данных или хотите запустить проверку вручную, большинство программ требуют сначала размонтировать файловую систему. Поскольку размонтировать корневую файловую систему работающей системы невозможно, максимум, что можно сделать - перейти в однопользовательский режим (используя [telinit 1](#)), а затем перемонтировать корневую файловую систему в режиме «только чтение»; после этого можно провести проверку согласованности. (Монтирование файловых систем описано в следующем разделе – [Монтирование и размонтирование файловых систем](#).) Наилучший способ проверки файловых систем – загрузиться в резервную систему с CD-диска или USB- накопителя и провести проверку ваших файловых систем в размонтированном виде.

### Преимущества журналирования

Для проверки системы ext2 с помощью команды [fsck](#) может потребоваться значительное время, поскольку при этом необходимо полное чтение внутренней структуры данных (*метаданных*). Поскольку файловые системы становятся все больше и больше, это занимает все больше и больше времени; несмотря на то, что быстродействие дисков растет, полная проверка может занять до нескольких часов.

Эта проблема побудила к созданию *журналируемых* файловых систем. Такие файловые системы хранят недавние изменения в метаданных. После сбоя, чтобы определить, в каких частях файловой системы в результате сбоя могли возникнуть ошибки, драйвер файловой системы просматривает журнал. Это позволяет сократить время проверки целостности файловой системы до нескольких секунд, независимо от ее размера. Более того, драйвер файловой системы обычно проверяет файловую систему на этапе монтирования, поэтому дополнительная проверка с помощью [fsck](#), как правило, не требуется. Фактически в файловой системе xfs команде [fsck](#) делать нечего!

Перед иницированием проверки файловой системы вручную следует уточнить параметры конкретной команды [fsck](#) по документации. В примерах, представленных в листинге 24, команда [fsck](#) запускается с рабочего компакт-диска Ubuntu.

### Листинг 24. Ручной запуск fsck

```
root@ubuntu:~# fsck -p /dev/hda6
fsck 1.38 (30-Jun-2005)
Reiserfs super block in block 16 on 0x306 of format 3.6 with standard journal
Blocks (total/free): 5002224/4608416 by 4096 bytes
Filesystem is clean
Replaying journal..
```

```
Reiserfs journal '/dev/hda6' in blocks [18..8211]: 0 transactions replayed
Checking internal tree..finished
root@ubuntu:~# fsck -p /dev/hda2
fsck 1.38 (30-Jun-2005)
B00T: clean, 34/26208 files, 22488/104420 blocks
root@ubuntu:~# fsck -p /dev/hda7
fsck 1.38 (30-Jun-2005)
root@ubuntu:~# fsck -a /dev/hda8
fsck 1.38 (30-Jun-2005)
dosfsck 2.11, 12 Mar 2005, FAT32, LFN
/dev/hda8: 1 files, 2/501039 clusters
```

## «Продвинутые» инструменты

Также существуют более функциональные средства для проверки и восстановления файловых систем. Правила использования можно найти в документации man, а практические рекомендации – в Linux Documentation Project (см. [Ресурсы](#)). Почти все эти команды требуют, чтобы файловая система была размонтирована, хотя некоторые функции могут использоваться в файловых системах, смонтированных в режиме «только чтение». Некоторые из этих команд описаны далее.

Прежде чем предпринимать какие-либо исправления, обязательно создавайте резервную копию файловой системы.

### Инструменты для файловых систем ext2 и ext3

#### tune2fs

Настраивает параметры файловых систем ext2 и ext3. Используется для добавления журнала к системе ext2, делая, таким образом, из нее ext3, а также выводит или устанавливает максимальное число монтирований, после которого необходима проверка. Вы также можете задать метку и назначить или запретить выполнение дополнительных опций.

#### dump2fs

Выывает информацию о дескрипторах суперблоков и групп блоков в файловых системах ext2 и ext3.

#### debugfs

Команда для интерактивной отладки файловой системы. Используйте ее для проверки или изменения состояния файловых систем ext2 или ext3.

### Инструменты для файловых систем ReiserFS

#### reiserfstune

Выывает и настраивает параметры файловой системы ReiserFS.

#### debugreiserfs

Выполняет функции, аналогичные dump2fs и debugfs, для файловой системы ReiserFS.

### Инструменты для файловой системы XFS

#### xfs\_info

Выывает информацию о системе XFS.

#### xfs\_growfs

Расширяет файловую систему XFS (если имеется дополнительный раздел).

#### xfs\_admin

Изменяет параметры файловой системы XFS.

#### xfs\_repair

Восстанавливает файловую систему XFS, когда проверок при монтировании установке недостаточно для восстановления системы.

### xfs\_db

Проверяет или отлаживает файловую систему XFS.

[предыдущая](#)

## Раздел 4. Монтирование и размонтирование файловых систем

Этот раздел охватывает материалы темы 1.104.3 для экзамена 101 на Младший уровень администрирования. Тема имеет третий уровень значимости.

Из этой темы вы узнаете, как:

- монтировать файловые системы
- размонтировать файловые системы
- конфигурировать файловые системы, монтируемые при загрузке
- конфигурировать монтируемые пользователем съемные файловые системы, такие как системы на магнитных лентах, дискетах и CD.

### Монтирование файловых систем

Файловая система Linux представляет собой единое большое дерево с корнем /. Тем не менее мы говорим о файловых системах различных устройств и разделов. Сейчас мы разрешим это кажущееся несоответствие. Корневая файловая система монтируется в процессе инициализации. Все остальные созданные нами файловые системы не могут быть использованы системой Linux, пока они не будут *смонтированы в точку монтирования*.

Точка монтирования – это просто каталог в текущей совокупности смонтированных файловых систем, где файловая система данного устройства прикрепляется к общему дереву. Монтирование – это процесс, который делает файловую систему устройства частью единой файловой системы, доступной для Linux. Например, можно монтировать файловые системы на разделах жесткого диска, таких как /boot, /tmp или /home, а также на дискетах - /mnt/floppy и на CD-ROM - /media/cdrom1.

Кроме файловых систем на разделах, дискетах и CD, существуют и другие типы файловых систем. Мы вкратце упоминали файловую систему tmpfs, являющуюся файловой системой в виртуальной памяти. Также можно монтировать одну файловые системы одного компьютера на другом компьютере, используя сетевые файловые системы, такие как NFS или AFS. Можно создать файл в файловой системе, отформатировать его как файловую систему (возможно, другого типа) и смонтировать эту новую файловую систему.

Хотя процесс монтирования фактически монтирует *файловую систему* какого-либо устройства (или другого ресурса), принято говорить, что вы "монтируете устройство", понимая под этим "монтирование файловой системы устройства".

Базовая форма команды **mount** имеет два параметра: устройство (или ресурс), содержащие монтируемую файловую систему, и точка монтирования. Например, смонтируем наш раздел с системой FAT32 /dev/hda8 в точке монтирования /dos, как показано в листинге 25.

### Листинг 25. Монтирование /dos

```
root@pinguino:~# mount /dev/hda8 /dos
```

Точка монтирования должна существовать прежде, чем в нее что-либо будет смонтировано. В

результате монтирования файлы и подкаталоги монтируемой файловой системы становятся файлами и подкаталогами точки монтирования. Если каталог точки монтирования уже содержал файлы и подкаталоги, они становятся невидимыми до тех пор, пока файловая система не будет демонтирована. Хороший способ избежать этого – использовать в качестве точек монтирования только пустые каталоги.

После монтирования файловой системы файлы и каталоги, созданные или скопированные в точку монтирования или в ее подкаталог, будут располагаться в смонтированной файловой системе. Так, в нашем примере, файл /dos/sampdir/file.txt будет создан в системе FAT32, смонтированной в точке /dos.

Обычно команда **mount** автоматически определяет тип файловой системы. Но иногда может потребоваться явное задание типа файловой системы, для чего используется опция **-t**, как показано в листинге 26.

### Листинг 26. Монтирование с явным заданием типа файловой системы

```
root@pinguino:~# mount -t vfat /dev/hda8 /dos
```

Чтобы увидеть, какие файловые системы смонтированы, используйте **mount** без параметров. В Листинге 27 приведен пример для нашей системы.

### Листинг 27. Просмотр смонтированных файловых систем

```
/dev/hda6 on / type reiserfs (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
usbfs on /proc/bus/usb type usbfs (rw)
tmpfs on /lib/modules/2.6.12-10-386/volatile type tmpfs (rw,mode=0755)
/dev/hda2 on /boot type ext3 (rw)
/dev/hda8 on /dos type vfat (rw)
/dev/hda7 on /home type xfs (rw)
/dev/hda1 on /media/hdal type ntfs (rw)
tmpfs on /dev type tmpfs (rw,size=10M,mode=0755)
```

Аналогичную информацию можно просмотреть с помощью команд /proc/mounts или /etc/mtab; обе они выводят информацию о смонтированных файловых системах.

### Опции монтирования

Команда **mount** имеет несколько опций, которые меняют ее поведение по сравнению с поведением по умолчанию. Например, можно смонтировать файловую систему «только для чтения», указав атрибут **-o ro**. Если файловая система уже смонтирована – добавьте **remount**, как показано в листинге 28.

### Листинг 28. Установка атрибута "только чтение"

```
root@pinguino:~# mount -o remount,ro /dos
```

## **Замечания:**

- указывайте опции через запятую;
- при перемонтировании уже смонтированной файловой системы достаточно определить либо точку монтирования, либо название устройства. Указывать и то и другое не обязательно;
- нельзя перемонтировать файловую систему, созданную только для чтения, в режим чтения/записи. Неизменяемые носители, например, на CD-ROM, автоматически монтируются только для чтения.
- для перемонтирования устройства, допускающего запись, в режим чтения/записи введите **-o remount, rw**

Команды перемонтирования не будут выполнены, если какой-либо процесс имеет открытые файлы или каталоги в перемонтируемой файловой системе. Для нахождения открытых файлов используется команда **lsof**. За более подробной информацией о дополнительных опциях команды **lsof** обращайтесь к документации man.

## **fstab**

Из руководства к теме 102 "[Подготовка к экзамену LPI 101 \(тема 102\) d: Установка Linux и управление пакетами](#)", вы узнали, как с помощью параметра **root=** в GRUB и LILO сообщить загрузчику о том, какая файловая система монтируется в качестве корневой. Смонтировав эту файловую систему, процесс установки запускает **mount** с опцией **-a** для автоматического монтирования набора файловых систем. Этот набор задается в файле /etc/fstab. В листинге 29 показан файл /etc/fstab для системы Ubuntu, установленной на файловые системы, созданные ранее в данном руководстве.

### **Листинг 29. Пример использования fstab**

```
root@pinguino:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
#<file system> <mount point> <type> <options> <dump> <pass>
proc          /proc      proc    defaults        0      0
/dev/hda6      /         reiserfs defaults        0      1
/dev/hda2      /boot     ext3    defaults        0      2
/dev/hda8      /dos      vfat    defaults        0      0
/dev/hda7      /home     xfs     defaults        0      2
/dev/hda1      /media/hda1 ntfs    defaults        0      0
/dev/hda5      none      swap    sw            0      0
/dev/hdc       /media/cdrom0 udf,iso9660 user,noauto  0      0
/dev/fd0       /media/floppy0 auto   rw,user,noauto  0      0
```

Строки, начинающиеся символом #, являются комментариями. Остальные строки содержат шесть полей. Поскольку эти поля позиционные, все они должны быть заполнены.

#### **file system**

Для вышеупомянутых примеров имя должно быть задано как /dev/hda1.

#### **mount point**

Это точка монтирования, рассмотренная в разделе [Монтирование файловых систем](#).

Для пространства подкачки это поле имеет значение **none**. Для файловых систем ext2, ext3 и xfs можно также указывать метку тома, например: **LABEL=XFSHOME**. Это делает систему более устойчивой при установке и удалении устройств.

#### **type**

Определяет тип файловой системы. CD/DVD-диски часто имеют разные файловые

системы - ISO9660 или UDF - поэтому вы можете перечислить различные возможности в виде списка, разделенного запятыми. Если вы хотите, чтобы `mount` автоматически определила тип, используйте `auto`, как сделано в последней строке для дискеты.

#### **option**

Определяет параметры монтирования. Для монтирования со значениями по умолчанию используйте `defaults`. Несколько полезных опций:

- `rw` и `ro` указывают монтирование файловой системы в режиме чтения/записи или только для чтения.
- `noauto` указывает, что файловая система не должна автоматически монтироваться при загрузке или при выдаче команды `mount -a`. В нашем примере эта опция применена для съемных устройств.
- `user`
- определяет, что пользователь, не имеющий прав `root`, может монтировать или демонтировать данную файловую систему. Это особенно полезно для съемных носителей. Эта опция должна быть задана в `/etc/fstab`, а не в команде `mount`.
- `exec` или `noexec` определяют, позволяют ли исполнение файлов из данной файловой системы. Для файловых систем, монтируемых пользователем, по умолчанию устанавливается значение `noexec`, если только **после** поля `user` не указано `exec`.
- `noatime` отключает запись атрибута времени доступа к файлу. Это может повысить производительность.

#### **dump**

Определяет, будет ли команда `dump` включать данную файловую систему ext2 или ext3 в резервные копии. Значение 0 означает, что `dump` игнорирует данную файловую систему.

#### **pass**

Ненулевые значения `pass` определяют порядок проверки файловых систем во время загрузки, как описано в теме [Проверка файловых систем](#).

Для монтирования файловых систем, перечисленных в `/etc/fstab`, достаточно задать либо имя устройства, либо точку монтирования. Оба параметра одновременно задавать не нужно.

За более подробным описанием функций `fstab` и `mount`, включая не рассмотренные здесь опции, обращайтесь к документации `man`.

### [Размонтирование файловых систем](#)

Все смонтированные файловые системы обычно автоматически размонтируются системой при перезагрузке или выключении. При размонтировании файловой системы все кэшированные данные файловой системы сохраняются на диск.

Также можно размонтировать файловую систему вручную. В действительности это необходимо делать всякий раз, когда вы удаляете записываемый съемный носитель - дискету, USB-диск или флэш-накопитель. Прежде чем размонтировать файловую систему, следует убедиться в отсутствии работающих процессов, которые имеют открытые файлы в этой файловой системе. Затем используйте команду `umount`, указав в качестве аргумента либо имя устройства, либо точку монтирования. Несколько примеров успешного и безуспешного размонтирования приведено в листинге 30.

### **Листинг 30. Размонтирование файловых систем**

```
root@pinguino:~# lsof /dos
root@pinguino:~# umount /dos
root@pinguino:~# mount /dos
```

```
root@pinguino:~# umount /dev/hda8
root@pinguino:~# umount /boot
umount: /boot: device is busy
umount: /boot: device is busy
root@pinguino:~# lsof /boot
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
klogd  6498 klog    1r   REG      3,2 897419 6052 /boot/System.map-2.6.12-10-386
```

После размонтирования файловой системы файлы в каталоге, использовавшемся в качестве точки монтирования, снова становятся видимыми.

### Пространство подкачки

Вы могли заметить, в описании команды **fstab**, что пространство подкачки не имеет точки монтирования. В процессе загрузки система обычно активизирует пространство подкачки, указанное в **/etc/fstab**, если не указана опция **noauto**. Для управления пространством подкачки в работающей системе, например, для добавления нового раздела подкачки, используются команды **swapon** и **swapoff**. Подробнее см. документацию **man**.

Для просмотра активизированных в данный момент устройств подкачки используйте **cat /proc/swaps**.

| [предыдущая](#) |

## Раздел 5. Дисковые квоты

Данный раздел охватывает материалы темы 1.104.4 для экзамена 101 на младший уровень администрирования. Раздел имеет третий уровень значимости.

Из этого раздела вы узнаете, как:

- устанавливать дисковые квоты
- устанавливать пределы квот
- проверять квоты
- получать отчеты о квотах

Установка квот позволяет контролировать использование дисков пользователями и группами пользователей. Квоты не дают отдельным пользователям и группам использовать большую часть файловой системы, чем им разрешено, или полностью заполнять эту часть. Квоты устанавливаются и изменяются пользователем root. Чаще всего они используются в многопользовательских системах, реже – в однопользовательских рабочих станциях.

### Установка режима квотирования

Для установки режима квотирования необходима поддержка ядра. Как правило, современные ядра версий 2.4 или 2.6 имеют всю необходимую поддержку. В более ранних версиях поддержка могла быть неполной, что требовало от вас сборки собственного ядра. В современных реализациях поддержка квот чаще всего реализуется в виде модулей ядра. Существует три версии поддержки квот: **vfsold** (версия 1), **vfsv0** (версия 2) и **xfs** (для файловой системы XFS). Данный раздел охватывает вторую версию квотирования для файловых систем, отличных от XFS, и квоты **xfs** для файловой системы XFS.

Первый шаг для введения квотирования – указание опций **usrquota** или **grpquota** в определении файловой системы в **/etc/fstab**, соответственно тому, что вы хотите ввести: квоты для пользователей, групп или то и другое. Рассмотрим создание обоих типов квот в файловой системе XFS, используемой для каталогов **home** в нашем примере, а также для файловой системы **/boot**, чтобы видеть, как это делается в различных файловых системах. Сделайте, как

показано в листинге 31.

### Листинг 31. Установка режима квотирования в /etc/fstab

/dev/hda2	/boot	ext3	defaults,usrquota,grpquota	0	2
/dev/hda7	/home	xfs	defaults,usrquota,grpquota	0	2

В файловой системе XFS информация о квотах входит в состав метаданных. В других файловых системах информация о квотах для пользователей хранится в файле `aquota.user` в корне файловой системы, а для групп пользователей – в файле `aquota.group`. Для квот первой версии использовались файлы `quota.user` и `quota.group`.

Внеся изменения в `/etc/fstab` и добавив квоты, необходимо перемонтировать файловые системы и, для файловых систем, отличных от XFS, создать файлы квот и разрешить проверку квотирования. Команда `quotacheck` проверяет квотирование на всех файловых системах и создает необходимые файлы `aquota.user` и `aquota.group`, если их не существует. Также она может восстановить поврежденные файлы квот. Более подробно см. руководство `man`. Команда `quotaon` включает проверку квот. Пример показан в листинге 32. Следующие опции используются в обеих командах:

**-a**

для всех файловых систем в `/etc/fstab`, для которых разрешено автоматическое монтирование

**-u**

для пользовательских квот (установлено по умолчанию)

**-g**

для групповых квот

**-v**

для подробного вывода

### Листинг 32. Создание файлов квот и включение квотирования

```
root@pinguino:~# quotacheck -augv
quotacheck: Scanning /dev/hda2 [/boot] quotacheck: Cannot stat old user quota
file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
quotacheck: Cannot stat old user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
done
quotacheck: Checked 4 directories and 23 files
quotacheck: Old file not found.
quotacheck: Old file not found.
quotacheck: Skipping /dev/hda7 [/home]
root@pinguino:~# quotaon -ugva
/dev/hda2 [/boot]: group quotas turned on
/dev/hda2 [/boot]: user quotas turned on
```

### Проверка квот при загрузке

Команды `quotacheck` и `quotaon` обычно входят в состав инициализационных сценариев, поэтому квотирование включается каждый раз при перезагрузке системы. Дополнительная информация содержится в руководстве Quota Mini HOWTO (см. [Ресурсы](#)).

Команда **quotaoff** отключает использование квот, если необходимо.

### Установка пределов квот

Как видно, квотирование управляет либо через бинарные файлы в корне файловой системы, либо через метаданные файловой системы. Для установки квоты для отдельного пользователя используется команда **edquota**. Эта команда извлекает информацию о квотах для данного пользователя из различных файловых систем, для которых включено квотирование, создает временный файл и открывает редактор, позволяющий изменять квоты. Информацию о том, какой именно редактор используется, см. в документации тарн команды **edquota**. Для изменения квот необходимо обладать полномочиями root. Полученная информация будет выглядеть примерно так, как в листинге 33.

### Листинг 33. Запуск edquota

```
Disk quotas for user ian (uid 1000):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/hda2          0          0          0          0          0          0
/dev/hda7      2948          0          0        172          0          0
```

Как видно из примера, **edquota** показывает текущее использование блоков 1K и inode для каждой файловой системы, где включено квотирование. Также существуют мягкие и жесткие пределы на использование блоков и inode. В данном примере их значения установлены в 0, что означает, что пределы квот не установлены.

Мягкие пределы – это пределы, при достижении которых пользователь получает предупреждения о превышении квоты. Жесткие пределы – это границы, которые пользователь не может превысить. Можно считать, что ограничения на блоки – это ограничения на объем сохраняемой информации, а ограничения на inode – это ограничения количества файлов и каталогов.

### Изменение пределов квот

Чтобы изменить пределы квот, отредактируйте значения во временном файле и сохраните его. Чтобы не применять изменения, закройте файл без сохранения. Предположим, вы хотите установить для меня ограничения в файловой системе /home: по объему – 10 Мбайт, по количеству – 1000 файлов. Добавляя 10% запаса на жесткие пределы, устанавливаем значения, как показано в листинге 34.

### Листинг 34. Установка пределов

```
Disk quotas for user ian (uid 1000):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/hda2          0          0          0          0          0          0
/dev/hda7      2948      10240      11264        172      1000      1100
```

Сохраните файл, чтобы применить изменения. В данном примере для пользователя ian в файловой системе /boot квоты не были изменены, поскольку пользователь ian не имеет прав для записи в эту файловую систему. Также обратите внимание, что любые изменения, внесенные в данные об используемых блоках или inodes, будут проигнорированы.

### Копирование квот

Теперь предположим, что вы создаете идентификаторы для группы пользователей –

слушателей учебного курса. Допустим, у вас есть пользователи gretchen, tom и greg, и вы хотите назначить им такие же квоты, как у ian. Для этого применяется опция **-p** команды **edquota**, которая использует значения квот пользователя ian в качестве прототипа для квот других пользователей, как показано в листинге 35.

### Листинг 35. Установка квот по прототипу

```
root@pinguino:~# edquota -p ian gretchen tom greg
```

### Квоты для групп пользователей

Команду **edquota** также можно использовать для ограничения выделения дискового пространства на основании принадлежности файлов группам. Пусть, например, три упомянутых выше слушателя объединены в основную группу xml-101. Чтобы задать пределы на суммарный объем, используемый всеми членами группы, на уровне 25 Мбайт и 2500 файлов, используйте команду **edquota -q xml-101** и установите значения, как показано в листинге 36.

### Листинг 36. Установка квот для группы пользователей

```
Disk quotas for group xml-101 (gid 1001):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/hda2          0          0          0          0          0          0
/dev/hda7        28      25600      28160      10      2500      2750
```

### Льготный период

Пользователи могут превышать мягкие пределы квот в течение "*льготного периода*", который по умолчанию составляет 7 дней. После истечения этого периода мягкие пределы становятся жесткими. Льготные периоды устанавливаются с помощью опции **-y** команды **edquota**. Перед вами вновь окажется редактор с данными, аналогичными представленным в листинге 37. Как и раньше, сохраните изменения, чтобы обновить значения. Убедитесь, что пользователям предоставлено достаточно времени для получения предупреждений по электронной почте и удаления некоторых файлов.

### Листинг 37. Установка льготных периодов

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem      Block grace period      Inode grace period
/dev/hda2            7days                  7days
/dev/hda7            7days                  7days
```

### Проверка квот

Команда **quota** без указания опций выводит квоты для вызвавшего ее пользователя во всех файловых системах, где такие квоты установлены. Опция **-v** выводит информацию для всех файловых систем, в которых включено квотирование. Пользователь root может также добавить имя пользователя к команде, чтобы просмотреть ограничения для конкретного пользователя. Эти команды представлены в листинге 38.

### Листинг 38. Просмотр квот

```
root@pinguino:~# quota
Disk quotas for user root (uid 0): none
root@pinguino:~# quota -v
Disk quotas for user root (uid 0):
  Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
    /dev/hda2     19173      0       0           26      0       0       0
    /dev/hda7      16       0       0           5      0       0       0
root@pinguino:~# quota -v ian
Disk quotas for user ian (uid 1000):
  Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
    /dev/hda2      0       0       0           0      0       0       0
    /dev/hda7    2948    10240   11264          172    1000    1100
```

Наряду с текущими уровнями использования выводятся жесткие и мягкие пределы. В листинге 39 показано, что будет, если превысить границы мягкого предела, и что произойдет, если попытаться превысить жесткий предел. В данном примере мы создаем файл размером примерно 4 Мбайт, а затем копируем его. Вместе с первоначальным уровнем использования около 3 Мбайт этого достаточно для превышения мягкого предела. Обратите внимание, что рядом с мягким пределом выводится звездочка, показывающая, что пользователь превысил квоту. Также заметьте, что в столбце *grace period* теперь показано, сколько времени есть у пользователя, чтобы исправить положение.

### Листинг 39. Превышение квот

```
ian@pinguino:~$ dd if=/dev/zero of=big1 bs=512 count=8000
8000+0 records in
8000+0 records out
4096000 bytes transferred in 0.019915 seconds (205674545 bytes/sec)
ian@pinguino:~$ cp big1 big2
ian@pinguino:~$ quota
Disk quotas for user ian (uid 1000):
  Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
    /dev/hda7    10948*    10240   11264    7days      174    1000    1100
ian@pinguino:~$ cp big1 big3
cp: writing `big3': Disk quota exceeded
```

### Создание отчета о квотах

Проверять квоты для каждого пользователя последовательно не очень удобно, поэтому для создания отчета о квотах используется команда [repquota](#). В листинге 40 показано, как просмотреть квоты для всех пользователей и групп каталога /home.

### Листинг 40. Превышение квот

```
root@pinguino:~# repquota -ug /home
*** Report for user quotas on device /dev/hda7
Block grace time: 7days; Inode grace time: 7days
                                         Block limits                               File limits
User          used    soft    hard grace      used    soft    hard grace
-----
root          --     16      0      0            5      0      0
ian          +-   11204  10240  11264  6days     175  1000  1100
tom          --      8    10240   11264            3    1000  1100
```

```

gretchen --      8  10240   11264          3  1000   1100
greg     --    12  10240   11264          4  1000   1100

*** Report for group quotas on device /dev/hda7
Block grace time: 7days; Inode grace time: 7days
                                         Block limits           File limits
Group        used    soft    hard grace    used    soft    hard grace
-----
root       --      16      0      0          5      0      0
ian        --  11204      0      0         175      0      0
xml-101   --     28  25600  28160         10  2500  2750

```

Обратите внимание на знак плюс в листинге для пользователя ian. Он показывает, что ian превысил квоту.

Как и в других командах, относящихся к квотам, опция **-a** создает отчет по всем файловым системам, где включено квотирование. Опция **-v** формирует более подробный вывод. Опция **-n** выводит список номеров пользователей без определения их имен. Это может повысить производительность для больших отчетов, но результат хуже читается человеком.

## В начало

### Предупреждение пользователей

Команда **warnquota** используется для отправки предупреждений по электронной почте пользователям, превысившим квоты. Если квоту превысила группа, сообщения по электронной почте отправляются пользователям, указанным в `/etc/quota.gradmins`. Обычно **warnquota** запускается периодически как задание **cron**. Более подробно о стоп и **warnquota** см. документацию `man`.

## Предыдущая

## Раздел 6. Полномочия доступа к файлам и управление доступом

Этот раздел охватывает материалы темы 1.104.5 для экзамена 101 на младший уровень администрирования (LPIC-1). Тема имеет пятый уровень значимости.

Из этой темы Вы узнаете о:

- Пользователях и группах
- Полномочиях доступа к файлам и каталогам
- Изменении полномочий
- Режимах доступа
- Файлах только для чтения
- Режимах создания файлов по умолчанию

### Пользователи и группы

К этому времени вам уже должно быть известно, что Linux является многопользовательской системой, и каждый пользователь принадлежит к одной основной группе и, возможно, дополнительным группам. Кроме того, войдя в систему в качестве одного пользователя, с помощью команд **SU** или **sudo -s** можно стать другим пользователем. Понятие владения файлами в Linux тесно связано с идентификаторами пользователя и группами, поэтому давайте повторим основные сведения о пользователях и группах.

## Кто я такой?

Если вы не стали другим пользователем, ваш идентификатор пользователя остался таким же, каким вы его ввели при входе в систему. Если вы становитесь другим пользователем, в приглашении командной строки может содержаться ваш идентификатор пользователя, как в большинстве примеров в этом руководстве. Если в приглашении командной строки не содержится идентификатора текущего пользователя, вы можете узнать его с помощью команды `whoami`. В листинге 41 показано несколько примеров, в которых настройки приглашения командной строки (из переменной среды PS1) отличаются от остальных примеров в этом руководстве.

#### Листинг 41. Определение идентификатора текущего пользователя

```
/home/ian$ whoami  
tom  
/home/ian$ exit  
exit  
$ whoami  
ian
```

#### В какие группы я вхожу?

Подобным же образом с помощью команды `groups` можно узнать, в какие группы вы входите. С помощью команды `id` можно получить информацию и о пользователях, и о группах. Добавив в качестве параметра к команде `groups` или `id` идентификатор пользователя, можно просмотреть информацию об этом пользователе, а не о текущем. Несколько примеров приведено в листинге 42.

#### Листинг 42. Определение членства в группах

```
$ su tom  
Password:  
/home/ian$ groups  
xml-101  
/home/ian$ id  
uid=1001(tom) gid=1001(xml-101) groups=1001(xml-101)  
/home/ian$ exit  
$ groups  
ian adm dialout cdrom floppy audio dip video plugdev lpadmin scanner admin xml-101  
$ id  
uid=1000(ian) gid=1000(ian) groups=4(adm),20(dialout),24(cdrom),25(floppy),  
29(audio),30(dip),44(video),46(plugdev),104(lpadmin),105(scanner),106(admin),  
1000(ian),1001(xml-101)  
$ groups tom  
tom : xml-101
```

#### Владение файлом и полномочия доступа к нему

Точно так же как у любого пользователя есть свой идентификатор, а сам он является членом основной группы, у каждого файла в системе Linux есть связанные с ним один владелец и одна группа.

#### Обычные файлы

Для того чтобы вывести информацию о владельцах и группах файлов, выполните команду `ls -l`

### Листинг 43. Определение владельца файла

```
gretchen@pinguino:~$ ls -l /bin/bash .bashrc
-rw-r--r-- 1 gretchen xml-101 2227 Dec 20 10:06 .bashrc
-rwxr-xr-x 1 root      root    645140 Oct  5 08:16 /bin/bash
```

В этом примере файл .bashrc пользователя gretchen принадлежит ей и входит в группу xml-101, которая является её основной группой. Точно так же, владельцем /bin/bash является пользователь root, а его основной группой – группа root. Имена пользователей и названия групп берутся из различных пространств имен, поэтому название группы может быть таким же, как имя пользователя. На самом деле по умолчанию во многих дистрибутивах для каждого нового пользователя создаётся группа с таким же названием.

Для каждого объекта файловой системы в модели полномочий Linux есть три типа полномочий: полномочия чтения (r), записи (w) и выполнения (x). В полномочия записи входят также возможности удаления и изменения объекта. Кроме того, эти полномочия указываются отдельно для владельца файла, членов группы файла и для всех остальных.

Вернемся к первой колонке листинга 43. Обратите внимание, что она содержит строку из десяти символов. Первый символ описывает тип объекта (в этом примере - обозначает обычный файл), а оставшиеся девять символов представляют три группы по три символа в каждой. Первая группа обозначает полномочия чтения, записи и выполнения для владельца файла. Знак "-" обозначает, что соответствующего полномочия дано не было. Поэтому пользователь gretchen может читать файл .bashrc, проводить в него запись, но не может выполнять его, в то время как пользователь root может читать файл /bin/bash, проводить в него запись и выполнять его. Вторая группа обозначает полномочия чтения, записи и выполнения для группы файла. Члены группы xml-101 могут считывать файл .bashrc пользователя gretchen, но не могут производить в него запись, так же, как и все остальные. Подобным же образом, члены группы root и все остальные пользователи могут считывать и выполнять файл /bin/bash.

### Каталоги

Для каталогов используются те же флаги полномочий, что и для обычных файлов, однако интерпретируются они иначе. Наличие у пользователя полномочий чтения каталога позволяет ему просматривать содержимое каталога. Пользователь, имеющий полномочия записи, может создавать и удалять файлы в этом каталоге. Полномочия выполнения позволяют пользователю входить в этот каталог и просматривать все подкаталоги. Без полномочий выполнения объекты файловой системы, находящиеся в этом каталоге, недоступны. Без полномочий чтения объекты файловой системы, находящиеся в каталоге, нельзя просматривать, однако доступ к ним можно получить, если вы знаете полный путь к этому объекту на диске. В листинге 44 приведен несколько искусственный пример, иллюстрирующий этот момент.

### Листинг 44. Полномочия и каталоги

```
ian@pinguino:~$ ls -l /home
total 8
drwxr-x--- 2 greg      xml-101   60 2005-12-20 11:37 greg
drwx----- 13 gretchen  xml-101  4096 2005-12-21 12:22 gretchen
drwxr-xr-x  15 ian      ian      4096 2005-12-21 10:25 ian
d-wx--x--x  2 tom      xml-101   75 2005-12-21 11:05 tom
ian@pinguino:~$ ls -a ~greg
.  ..  .bash_history  .bash_profile  .bashrc
```

```

ian@pinguino:~$ ls -a ~gretchen
ls: /home/gretchen: Permission denied
ian@pinguino:~$ ls -a ~tom
ls: /home/tom: Permission denied
ian@pinguino:~$ head -n 3 ~tom/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

```

Первый символ длинного листинга описывает тип объекта (**d** для каталога). У каталога `home` пользователя `greg` установлены полномочия на чтение и выполнение для членов группы `xml-101`, поэтому пользователь `ian` может получить список файлов, находящихся в этом каталоге. У каталога `home` пользователя `Gretchen` нет ни полномочий на чтение, ни полномочий на запись, поэтому пользователь `ian` не может получить доступ к нему. У каталога `home` пользователя `tom` установлены только полномочия на выполнение, но нет полномочий на чтение, поэтому пользователь `ian` не может просмотреть содержимое этого каталога, но может получить доступ к располагающимся в нем объектам, если он точно знает, что они существуют.

### **Другие объекты файловой системы**

В длинном листинге могут содержаться объекты файловой системы, отличные от файлов и каталогов, что можно увидеть по первому символу листинга. Мы будем рассматривать эти объекты ниже в этом разделе, отметим на данный момент лишь возможные типы объектов.

*Таблица 3. Типы объектов файловой системы*

<b>Код</b>	<b>Тип объекта</b>
-	Обычный файл
d	Каталог
l	Символическая ссылка
c	Специальное символьическое устройство
b	Специальное блочное устройство
p	Буфер FIFO
s	Сокет

### **Изменение полномочий**

#### **Добавление полномочий**

Предположим, вы создали сценарий командной оболочки "Hello world". При создании сценария он обычно не является исполняемым. Чтобы добавить полномочия на выполнение, используйте команду **chmod** с параметром **+x**, как показано в листинге 45.

#### **Листинг 45. Создание исполняемого сценария командной оболочки**

```

ian@pinguino:~$ echo 'echo "Hello world!"'>hello.sh
ian@pinguino:~$ ls -l hello.sh
-rw-r--r-- 1 ian ian 20 2005-12-22 12:57 hello.sh
ian@pinguino:~$ ./hello.sh
-bash: ./hello.sh: Permission denied
ian@pinguino:~$ chmod +x hello.sh
ian@pinguino:~$ ./hello.sh
Hello world!
ian@pinguino:~$ ls -l hello.sh

```

```
-rwxr-xr-x 1 ian ian 20 2005-12-22 12:57 hello.sh
```

Подобным же образом можно использовать **r** для установки полномочий на чтение и **w** для установки полномочий на запись. На самом деле можно использовать вместе любую комбинацию **r**, **w** и **x**. Например, команда **chmod +rwx** установит для файла все полномочия на чтение, запись и выполнение. Использование **chmod** в таком виде добавляет не установленные на данный момент полномочия.

### Выборочное изменение

Вы могли заметить, что в приведенном выше примере права на выполнение устанавливаются для владельца, группы и других. Чтобы действовать более избирательно, необходимо использовать префикс для выражения режима: **u** для установки полномочий для пользователей, **g** для установки полномочий для групп и **o** для установки полномочий для всех остальных. Указание **a** определяет полномочия для всех пользователей, что равнозначно отсутствию префикса. В листинге 46 показано, как добавить пользователю и группе полномочия на запись и выполнение другой копии сценария командной оболочки.

### Листинг 46. Выборочное добавление полномочий

```
ian@pinguino:~$ echo 'echo "Hello world!"'>hello2.sh
ian@pinguino:~$ chmod ug+xw hello2.sh
ian@pinguino:~$ ls -l hello2.sh
-rwxrwxr-- 1 ian ian 20 2005-12-22 13:17 hello2.sh
```

### Снятие полномочий

Иногда вам нужно не добавить полномочия, а снять их. Просто измените **+** на **-**, чтобы удалить все указанные и установленные полномочия. В листинге 47 показано, как снять все полномочия для остальных пользователей с двух сценариев командной оболочки.

### Листинг 47. Снятие полномочий

```
ian@pinguino:~$ ls -l hello*
-rwxrwxr-- 1 ian ian 20 2005-12-22 13:17 hello2.sh
-rwxr-xr-x 1 ian ian 20 2005-12-22 12:57 hello.sh
ian@pinguino:~$ chmod o-xrw hello*
ian@pinguino:~$ ls -l hello*
-rwxrwx--- 1 ian ian 20 2005-12-22 13:17 hello2.sh
-rwxr-x--- 1 ian ian 20 2005-12-22 12:57 hello.sh
```

Следует отметить, что можно за один раз изменить полномочия более чем у одного файла. Как и с некоторыми другими командами, с которыми вы встречались в руководстве для экзамена 103, вы даже можете использовать параметр **-R** (или **--recursive**) для рекурсивного обхода каталогов и папок.

### Установка полномочий

Теперь, когда вы можете добавлять и удалять полномочия, вы можете задать вопрос – как установить только определенный набор полномочий. Это делается с помощью знака **=** вместо **+** и **-**. Для того чтобы установить полномочия для приведенных выше сценариев так, чтобы другие пользователи не имели прав доступа, можно вместо команд на удаление полномочий

использовать команду `chmod o= hello*`.

Если вы желаете установить различные полномочия для пользователя, группы и остальных пользователей, вы можете разделять различные выражения запятыми; например, `ug=rwx, o=rx`, также вы можете использовать цифровой способ указания полномочий, описываемый ниже.

### Установка полномочий в восьмеричном формате

До настоящего момента для указания полномочий вы использовали символы (ugo и gwx). В каждой группе существует три возможных типа полномочий. Также можно указывать полномочия, используя вместо символов числа в восьмеричном формате. Для установки полномочий, таким образом, может потребоваться до четырёх восьмеричных цифр.

Рассматривать первую цифру мы будем при обсуждении атрибутов. Вторая цифра определяет полномочия для пользователя, третья – полномочия для группы и четвертая – полномочия для остальных пользователей. Каждая из этих трех цифр получается путем сложения желаемых полномочий: на чтение (4), на запись (2) и на исполнение (1). В примере для `hello.sh`, приведенном в листинге 45, сценарий был создан с полномочиями `-rw-r--r--`, что соответствует восьмеричному 644. Установка прав на выполнение для всех изменит режим на 755.

Использование полномочий в цифровом виде очень удобно в случаях, когда вы хотите установить все полномочия сразу, не указывая одинаковые полномочия для каждой группы. Используйте таблицу 4 в качестве удобного справочника для установки полномочий в восьмеричном формате.

Таблица 4. Установка полномочий в числовом формате

Символический	Восьмеричный
rwx	7
rw-	6
r-x	5
r--	4
-wx	3
-w-	2
--x	1
---	0

### Режимы доступа

Когда вы входите в систему, запускается новый процесс командной оболочки с вашим идентификатором пользователя и идентификатором группы. Эти идентификаторы определяют полномочия на доступ ко всем файлам в системе. Обычно это означает, что вы не можете открывать файлы, принадлежащие другим пользователям, и системные файлы. На самом деле мы как пользователи полностью полагаемся на другие программы, выполняющие действия от нашего имени. Поскольку программы, которые вы запускаете, наследуют ваш идентификатор пользователя, они не могут получить доступа к объектам файловой системы, доступа к которым не имеете вы.

В качестве важного примера можно привести файл `/etc/passwd`, который не может быть изменен обычными пользователями напрямую, так как полномочия на запись есть только у пользователя `root`. Однако обычным пользователям необходима возможность изменения файла `/etc/passwd` каким-либо образом всякий раз, когда им нужно изменить свой пароль. Итак, если пользователь не может изменить этот файл, как это можно сделать?

### suid and sgid

В модели полномочий Linux есть два специальных режима доступа, называемых `suid`

(установить идентификатор пользователя) и sgid (установить идентификатор группы). Если у исполняемой программы установлен режим доступа **suid**, она будет запущена так, как если бы это сделал владелец файла, а не пользователь, который фактически её запустил. Подобно этому, при установленном режиме доступа **sgid** программа будет работать так, как если бы её запустил пользователь, входящий в группу, которой принадлежит файл, а не в группу, в которой фактически состоит пользователь. Эти режимы можно установить как по отдельности, так и вместе.

В листинге 48 показан исполняемый файл **passwd**, владельцем которого является пользователь **root**:

#### Листинг 48. Режим доступа **suid** файла /usr/bin/passwd

```
ian@pinguino:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 25648 2005-10-11 12:14 /usr/bin/passwd
```

Заметьте, что вместо **X** в тройке символов полномочий пользователя стоит **S**. Это обозначает, что для этой конкретной программы установлен флаг выполнения и режим доступа **suid**. При запуске программы **passwd** она будет выполняться так, как будто бы её запустил пользователь **root**, со всеми правами доступа привилегированного пользователя, а не того пользователя, который её действительно запустил. Поскольку программа **passwd** работает с уровнем доступа пользователя **root**, она может изменить файл **/etc/passwd**.

Флаги **suid** и **sgid** занимают в длинном листинге каталога то же место, что и флаг **X**. Если файл исполняемый, установленные флаги **suid** и **sgid** будут отображаться как маленькая **S**, в противном случае они будут выводиться как большая **S**.

Несмотря на то, что флаги **suid** и **sgid** очень удобны, и даже необходимы во многих ситуациях, неправильное использование этих режимов доступа может привести к появлению брешей в системе обеспечения безопасности. У вас должно быть как можно меньше программ, работающих в режиме доступа **suid**. Команда **passwd** является одной из немногих, которая должна работать в режиме **suid**.

#### Установка **suid** и **sgid**

Флаги **suid** и **sgid** устанавливаются с использованием символа **S**; например, **u+S** устанавливает режим доступа **suid**, а **g-S** снимает режим доступа **sgid**. В восьмеричном формате режиму **suid** соответствует значение 4 в первой цифре (старший разряд), а режиму **sgid** соответствует значение 2.

#### Каталоги и **sgid**

Если для каталога установлен режим **sgid**, все файлы и каталоги, созданные в нем, будут наследовать идентификатор группы этого каталога. В частности, это полезно для деревьев каталогов, используемых группой людей, работающих над одним проектом. В листинге 49 показано, как пользователь **greg** может настроить каталог, с которым могут работать все пользователи группы **xml-101**, а также пример того, как пользователь **gretchen** может создать файл в каталоге.

#### Листинг 49. Режим доступа **sgid** и каталоги

```
greg@pinguino:~$ mkdir xml101
greg@pinguino:~$ chmod g+ws xml101
greg@pinguino:~$ ls -ld xml101
drwxrwsr-x 2 greg xml-101 6 Dec 25 22:01 xml101
```

```
greg@pinguino:~$ su - gretchen
Password:
gretchen@pinguino:~$ touch ~greg/xml101/gretchen.txt
gretchen@pinguino:~$ ls -l ~greg/xml101/gretchen.txt
-rw-r--r-- 1 gretchen xml-101 0 Dec 25 22:02 /home/greg/xml101/gretchen.txt
```

Теперь любой член группы xml-101 может создавать файлы в папке xml101 пользователя greg. Как показано в листинге 50, другие члены группы не могут изменять файл gretchen.txt, но у них есть полномочия на запись в каталог и потому они могут удалить файл.

### Листинг 50. Режим доступа sgid и владение файлом

```
gretchen@pinguino:~$ su - tom
Password:
~$ cat something >> ~greg/xml101/gretchen.txt
-su: /home/greg/xml101/gretchen.txt: Permission denied
~$ rm ~greg/xml101/gretchen.txt
rm: remove write-protected regular empty file `/home/greg/xml101/gretchen.txt'? y
~$ ls -l ~greg/xml101
total 0
```

### Бит закрепления в памяти

Только что вы увидели, как любой пользователь, имеющий полномочия на запись в каталог, может удалить находящиеся в ней файлы. Такая ситуация может быть приемлемой для проекта рабочей группы, но нежелательна для файлового пространства, находящегося в общем доступе, например, каталога /tmp. К счастью, решение существует.

Оставшийся флаг режима доступа называется битом *закрепления* в памяти (*sticky bit*). Он представляется символом **t** и числом 1 в восьмеричной цифре старшего разряда. Он отображается в длинном листинге каталога на месте флага выполнения для остальных пользователей (последний символ), значение регистра аналогично значению регистра для *suid* и *sgid*. Если для каталога установлен этот флаг, он допускает удаление файла или ссылок только владельцем или суперпользователем (root). В листинге 51 показано, как пользователь greg может установить бит закрепления в памяти на свой каталог xml101, а также показано, что этот флаг установлен для каталога /tmp.

### Листинг 51. Каталоги с закреплением в памяти

```
greg@pinguino:~$ chmod +t xml101
greg@pinguino:~$ ls -l xml101
total 0
greg@pinguino:~$ ls -ld xml101
drwxrwsr-t 2 greg xml-101 6 Dec 26 09:41 xml101
greg@pinguino:~$ ls -ld xml101 /tmp
drwxrwxrwt 13 root root 520 Dec 26 10:03 /tmp
drwxrwsr-t 2 greg xml-101 6 Dec 26 09:41 xml101
```

Исторически, в системах UNIX® бит закрепления в памяти использовался на файлах и обозначал, что исполняемые файлы необходимо хранить в области свопинга, чтобы исключить их повторную загрузку. Современные ядра системы Linux игнорируют установку бита закрепления в памяти для файлов.

## Краткое резюме по режимам доступа

В таблице 5 представлено краткое описание символьического и восьмеричного представления трех обсуждаемых здесь режимов доступа.

Таблица 5. Режимы доступа

Режим	Символическое	Восьмеричное
suid	s u	4000
sgid	s g	2000
sticky	t	1000

Сочетая эти данные с приведенной ранее информацией о полномочиях, вы можете увидеть, что полномочия и режимы доступа, соответствующие каталоги xml101 пользователя greg, записываемые как drwxrwsr-t, в полном восьмеричном представлении выглядят как 1775.

## Файлы только для чтения

Режимы доступа и полномочия предоставляют широкие средства управления тем, кто и что может делать с файлами и каталогами. Как бы то ни было, они не предотвращают неумышленного удаления файлов пользователем root. В различных файловых системах существуют дополнительные *атрибуты*, которые предоставляют дополнительные возможности. Одним из таких атрибутов является атрибут *только для чтения*. Если этот атрибут установлен, даже пользователь root не сможет удалить файл, пока атрибут не будет снят.

Чтобы просмотреть, установлен ли на файле или каталоги флаг "только для чтения" (или какой-либо иной атрибут), необходимо использовать команду `lsattr`. Чтобы сделать файл доступным только для чтения, необходимо подать команду `chattr` с параметром `-i`.

В листинге 52 показано, что пользователь root может создать файл только для чтения, но не может удалить его до тех пор, пока не снят флаг "только для чтения".

## Листинг 52. Файлы только для чтения

```
root@pinguino:~# touch keep.me
root@pinguino:~# chattr +i keep.me
root@pinguino:~# lsattr keep.me
-----i-----
keep.me
root@pinguino:~# rm -f keep.me
rm: cannot remove `keep.me': Operation not permitted
root@pinguino:~# chattr -i keep.me
root@pinguino:~# rm -f keep.me
```

Для того чтобы изменить флаг "только для чтения", необходимы полномочия root, или, по меньшей мере, привилегия CAP\_LINUX\_IMMUTABLE. Перевод файлов в режим "только для чтения" часто выполняется в ходе мероприятий по обеспечению безопасности и обнаружению вторжений. Дополнительную информацию можно найти в документации man по ключевому слову capabilities (`man capabilities`).

## umask

При создании нового файла ему присваиваются определенные полномочия. Часто устанавливается режим 0666, что открывает возможность чтения и записи в этот файл для всех пользователей. В любом случае на процесс определения полномочий при создании файла влияет значение umask, которое определяет, какие полномочия пользователь **не** желает автоматически присваивать вновь создаваемым файлам и каталогам. Система использует

значение *umask* для ограничения изначально установленных полномочий. Просмотреть значение параметра *umask* можно с помощью команды *umask*, пример использования которой показан в листинге 53.

### Листинг 53. Вывод *umask* в восьмеричном формате

```
ian@pinguino:~$ umask  
0022
```

Необходимо помнить, что *umask* указывает только полномочия, которые **не** должны быть предоставлены. По умолчанию в системах Linux параметру *umask* обычно присваивается значение 0022, что **снимает** с групп и других пользователей полномочия на запись во вновь создаваемые файлы. Для того, чтобы вывести значение параметра *umask* в символьическом виде, отображая, какие полномочия разрешены, используйте параметр *-S*.

Использовать команду **umask** можно не только для просмотра, но и для установки значения параметра *umask*. Итак, если вы желаете хранить ваши файлы в конфиденциальном порядке и полностью отключить доступ группы и всех остальных пользователей, вам необходимо использовать значение *umask*, равное 0077. Также можно установить *umask* в символьическом виде, **umask u=rwx, g=, o=**, как это показано в листинге 54.

### Листинг 54. Установка *umask*

```
ian@pinguino:~$ umask  
0022  
ian@pinguino:~$ umask -S  
u=rwx,g=rx,o=rx  
ian@pinguino:~$ umask u=rwx,g=,o=  
ian@pinguino:~$ umask  
0077  
ian@pinguino:~$ touch newfile  
ian@pinguino:~$ ls -l newfile  
-rw----- 1 ian ian 0 2005-12-26 12:49 newfile
```

В следующем разделе будет показано, как можно изменить владельца или группу существующего объекта файловой системы.

| [предыдущая](#) |

## Раздел 7. Установка владельца и группы файла

Этот раздел охватывает материалы темы 1.104.6 для экзамена 101 на младший уровень администрирования (LPIC-1). Тема имеет первый уровень значимости.

Из этой темы Вы узнаете о:

- Изменении группы файла
- Группе, назначаемой новым файлам по умолчанию
- Изменении владельца файла

Из предыдущего раздела вы узнали, что у каждого объекта файловой системы имеется владелец и группа. В этом разделе вы узнаете, как изменить владельца и группу существующего файла, а также установить группу, назначаемую новым файлам по

умолчанию.

## Группа файла

Для того чтобы изменить группу файла, необходимо использовать команду [chgrp](#), в качестве параметров для которой необходимо указать название группы и название одного или нескольких файлов. Если пожелаете, вы можете использовать номер группы. Если группу файла изменяет обычный пользователь, он должен быть членом назначаемой группы. Пользователь root может назначить файлу любую группу. В листинге 55 показан соответствующий пример.

### Листинг 55. Изменение группы-владельца

```
ian@pinguino:~$ touch file1 file2
ian@pinguino:~$ ls -l file*
-rw-r--r-- 1 ian ian 0 2005-12-26 14:09 file1
-rw-r--r-- 1 ian ian 0 2005-12-26 14:09 file2
ian@pinguino:~$ chgrp xml-101 file1
ian@pinguino:~$ chgrp 1001 file2
ian@pinguino:~$ ls -l file*
-rw-r--r-- 1 ian xml-101 0 2005-12-26 14:09 file1
-rw-r--r-- 1 ian xml-101 0 2005-12-26 14:09 file2
```

Как и у многих команд, описываемых в этом руководстве, у команды [chgrp](#) есть параметр [-R](#), который позволяет рекурсивно применять изменения ко всем выбранным файлам и подкаталогам.

## Группа, назначаемая по умолчанию

[В предыдущем разделе](#) вы узнали, как установка режима sgid для каталога может привести к тому, что файлы, создаваемые в этом каталоге, будут принадлежать группе этого каталога, а не той группе, в которую входит пользователь, создающий файл.

Также для временного изменения вашей основной группы на другую, членом которой вы являетесь, вы можете воспользоваться командой [newgrp](#). Будет создана новая командная оболочка, при выходе из которой будет восстановлена ваша прежняя основная группа, что продемонстрировано в листинге 56.

### Листинг 56. Использование команды newgrp для временного изменения группы по умолчанию

```
ian@pinguino:~$ newgrp xml-101
ian@pinguino:~$ groups
xml-101 adm dialout cdrom floppy audio dip video plugdev lpadmin scanner admin ian
ian@pinguino:~$ touch file3
ian@pinguino:~$ ls -l file3
-rw-r--r-- 1 ian xml-101 0 2005-12-26 14:34 file3
ian@pinguino:~$ exit
ian@pinguino:~$ groups
ian adm dialout cdrom floppy audio dip video plugdev lpadmin scanner admin xml-101
```

## Владелец файла

Пользователь root может изменить владельца файла с помощью команды [chown](#). В простейшей форме синтаксис этой команды схож с синтаксисом команды [chgrp](#), за

исключением того, что вместо названия или идентификатора группы используется имя или идентификатор пользователя. Одновременно можно изменить группу файла, добавив справа от имени или идентификатора пользователя двоеточие и название или идентификатор группы. В случае если установлено только двоеточие, будет использоваться группа пользователя по умолчанию. И, конечно же, параметр **-R** приведет к рекурсивному внесению изменений. В листинге 57 показан соответствующий пример.

### Листинг 57. Использование команды **chown** для изменения владельца файла

```
root@pinguino:~# ls -l ~ian/file4
-rw-r--r-- 1 ian ian 0 2005-12-26 14:44 /home/ian/file4
root@pinguino:~# chown greg ~ian/file4
root@pinguino:~# ls -l ~ian/file4
-rw-r--r-- 1 greg ian 0 2005-12-26 14:44 /home/ian/file4
root@pinguino:~# chown tom: ~ian/file4
root@pinguino:~# ls -l ~ian/file4
-rw-r--r-- 1 tom xml-101 0 2005-12-26 14:44 /home/ian/file4
```

В существовавшей ранее форме указания пользователя и группы вместо двоеточия использовалась точка. Использовать такой способ указания не рекомендуется, так как в случае, если в имени пользователя содержится точка, возникнет ошибка.

| [предыдущая](#) |

## Раздел 8. Жесткие и символические ссылки

Этот раздел охватывает материалы темы 1.104.7 для экзамена 101 на младший уровень администрирования (LPIC-1). Тема имеет первый уровень значимости.

Из этой темы Вы узнаете о:

- Жестких ссылках
- Символических ссылках

### Жесткие ссылки

В руководстве для раздела 103, "[Подготовка к экзамену LPI 101 \(раздел 103\): команды GNU и UNIX](#)" вы узнали, что файлы и каталоги хранятся в наборе **блоков**, а информация о файлах и каталогах хранится в узлах *inode*.

*Жесткими ссылками* называются указатели на inode. Так, фактически наименование файла является ссылкой на узел inode, содержащий информацию о файле. Как вы уже знаете, с помощью команды **ls** с параметром **-i** можно отобразить номера inode для записей файлов и каталогов.

Вы можете использовать команду **ln** для создания дополнительных жестких ссылок на существующие файлы (но не на каталоги, даже несмотря на то, что в системе . и .. определены как жесткие ссылки). Если на один узел inode указывает несколько ссылок, этот узел будет удален только тогда, когда количество ссылок на него станет равным нулю.

В листинге 58 показано, как создать файл и жесткую ссылку на него. Также там показано, что даже при удалении исходного наименования файла вторая жесткая ссылка предотвращает стирание при этом узла inode.

## Листинг 58. Жесткие ссылки

```
ian@pinguino:~$ echo testing > file1
ian@pinguino:~$ ls -l file*
-rw-r--r-- 1 ian ian 8 2005-12-26 15:35 file1
ian@pinguino:~$ ln file1 file2
ian@pinguino:~$ ls -l file*
-rw-r--r-- 2 ian ian 8 2005-12-26 15:35 file1
-rw-r--r-- 2 ian ian 8 2005-12-26 15:35 file2
ian@pinguino:~$ rm file1
ian@pinguino:~$ ls -l file*
-rw-r--r-- 1 ian ian 8 2005-12-26 15:35 file2
ian@pinguino:~$ cat file2
testing
```

Жесткие ссылки могут существовать только в рамках определенной файловой системы. Они не могут связывать несколько файловых систем, так как ссылка на файл происходит по номеру inode, который уникален только в рамках файловой системы.

### Поиск жестких ссылок

Если вам необходимо узнать, какие файлы ссылаются на определенный узел inode, вы можете использовать команду **find** и параметр **-samefile** с указанием имени файла или параметр **-inum** с указанием номера inode, как показано в листинге 59.

## Листинг 59. Поиск жестких ссылок

```
ian@pinguino:~$ ln file2 file3
ian@pinguino:~$ ls -il file2
172 -rw-r--r-- 2 ian ian 8 2005-12-26 15:35 file2
ian@pinguino:~$ find . -samefile file2
./file2
./file3
ian@pinguino:~$ find . -inum 172
./file2
./file3
```

### Символические ссылки

Другой формой ссылок, используемых в файловой системе Linux, является *символические ссылки* (чаще называемых просто *symlink*). В этом случае ссылка указывает на наименование другого объекта, а не на его узел inode. Символические ссылки могут указывать на каталоги и на файлы, расположенные в других файловых системах. Они часто используются для присвоения альтернативных имен системным командам. С помощью длинного листинга каталога можно увидеть объекты, являющиеся символическими ссылками; они обозначены маленькой буквой **l** в первом символе, как показано в листинге 60.

## Листинг 60. Примеры символьических ссылок

```
ian@pinguino:~$ ls -l /sbin/mkfs.*
-rwxr-xr-x 1 root root 14160 2005-09-20 12:43 /sbin/mkfs.cramfs
-rwxr-xr-x 3 root root 31224 2005-08-23 09:25 /sbin/mkfs.ext2
-rwxr-xr-x 3 root root 31224 2005-08-23 09:25 /sbin/mkfs.ext3
-rwxr-xr-x 2 root root 55264 2005-06-24 07:48 /sbin/mkfs.jfs
-rwxr-xr-x 1 root root 13864 2005-09-20 12:43 /sbin/mkfs.minix
lrwxrwxrwx 1 root root 7 2005-12-14 07:40 /sbin/mkfs.msdos -> mkdosfs
-rwxr-xr-x 2 root root 241804 2005-05-11 09:40 /sbin/mkfs.reiser4
```

```
-rwxr-xr-x 2 root root 151020 2004-11-25 21:09 /sbin/mkfs.reiserfs
lrwxrwxrwx 1 root root      7 2005-12-14 07:40 /sbin/mkfs.vfat -> mkdosfs
-rw-rxr-x 1 root root 303788 2005-04-14 01:27 /sbin/mkfs.xfs
```

В дополнение к типу `l` вы можете видеть справа стрелку `->`, за которой следует имя файла, на которое указывает ссылка. Например, команда `mkfs.vfat` является символьической ссылкой на команду `mkdosfs`. Вы найдете множество подобных ссылок в `/sbin` и других системных каталогах. Еще одной отличительной особенностью символьических ссылок является размер файла, равный длине имени файла, на который указывает эта ссылка.

Вы можете создать символьическую ссылку с помощью команды `ln` с параметром `-s`, как показано в листинге 61.

### Листинг 61. Создание символьических ссылок

```
ian@pinguino:~$ touch file5
ian@pinguino:~$ ln -s file5 file6
ian@pinguino:~$ ln -s file5 file7
ian@pinguino:~$ ls -l file*
-rw-r--r-- 2 ian ian 8 2005-12-26 15:35 file2
-rw-r--r-- 2 ian ian 8 2005-12-26 15:35 file3
-rw-r--r-- 1 ian ian 0 2005-12-26 17:40 file5
lrwxrwxrwx 1 ian ian 5 2005-12-26 17:40 file6 -> file5
lrwxrwxrwx 1 ian ian 5 2005-12-26 17:40 file7 -> file5
```

Заметьте, что количество ссылок в листинге каталога не обновилось. Удаление ссылки не влияет на файл, на который она ссылается. Символьическая ссылка не защищает файл от удаления; в случае, если файл, на который указывает ссылка, перемещается или удаляется, эта ссылка будет испорчена. По этой причине во многих системах для обозначения символьических ссылок в листинге каталогов используются различные цвета, чаще всего – светло-голубой для хороших ссылок и красный для испорченных.

### Поиск символьических ссылок

Если вам необходимо узнать, какие файлы символьски ссылкуются на определенный файл, вы можете использовать команду `find` и параметр `-lname` с указанием имени файла, как показано в листинге 62. В ссылках могут использоваться относительные и абсолютные пути, поэтому полезно поставить подстановочный символ «звездочка» перед именем искомого файла.

### Листинг 62. Поиск символьических ссылок

```
ian@pinguino:~$ mkdir linktest1
ian@pinguino:~$ ln -s ~/file3 linktest1/file8
.ian@pinguino:~$ find . -lname "*file3"
./linktest1/file8
ian@pinguino:~$ find . -lname "*file5"
./file7
./file6
```

## Пути и символические ссылки

В большинстве рассмотренных ранее примеров символические ссылки находились в той же каталоги, что и файлы, на которые они указывают, а пути в ссылках, соответственно, были относительные. В листинге 62 мы создали ссылку в подкаталоге linktest1, которая указывала на абсолютное расположение файла which (~file3). При создании символьческих ссылок вам необходимо решить, какие пути вы будете использовать, относительные или абсолютные, так как вам доступны оба варианта. На рисунке 3 показано действие перемещения нескольких файлов и символьческих ссылок в подкаталог.

**Рисунок 3. Символьческие ссылки и пути**

```
ian@pinguino:~$ mkdir linktest2
ian@pinguino:~$ ls file?
file2 file3 file5 file6 file7
ian@pinguino:~$ mv file? linktest2
ian@pinguino:~$ ls -l linktest1 linktest2
linktest1:
total 0
linktest2:
total 8
-rw-r--r-- 2 ian ian 8 2005-12-26 15:35 file2
-rw-r--r-- 2 ian ian 8 2005-12-26 15:35 file3
-rw-r--r-- 1 ian ian 0 2005-12-26 17:40 file5
lrwxrwxrwx 1 ian ian 5 2005-12-26 17:40 file6 -> file5
lrwxrwxrwx 1 ian ian 5 2005-12-26 17:40 file7 -> file5
```

Красный цвет означает, что ссылка linktest1/file8 теперь испорчена. Это неудивительно, поскольку файла ~/file3 больше не существует. Однако две символьческие ссылки на file5 остаются хорошими, так как файл находится по тому же относительному пути, даже несмотря на то, что сам файл и обе ссылки были перемещены. Жестких и точных правил, которые позволяют выбрать, какие пути использовать в символьческих ссылках, абсолютные или относительные, не существует; отчасти это определяется вероятностью перемещения ссылок и файлов, на которые они указывают. Просто не забывайте учитывать этот вопрос при создании символьческих ссылок.

## Испорченные символьческие ссылки

Одно завершающее замечание по нашей испорченной ссылке. Попытка чтения из файла завершится ошибкой, так как файл не существует. Однако попытка записи сработает, если у вас есть достаточно полномочий для записи в файл, на который указывает ссылка, как показано в листинге 63.

**Листинг 63. Чтение и запись в испорченную символьческую ссылку**

```
ian@pinguino:~$ cat linktest1/file8
cat: linktest1/file8: No such file or directory
ian@pinguino:~$ echo "test file 8" >> linktest1/file8
ian@pinguino:~$ cat linktest1/file8
test file 8
ian@pinguino:~$ find . -name file3
./linktest2/file3
./file3
```

Поскольку я могу создавать файлы в моем каталоге home, запись в символьческую ссылку приведет к созданию отсутствующего файла, на который она указывает.

| [предыдущая](#) |

## **Раздел 9. Поиск и расположение системных файлов**

Этот раздел охватывает материалы темы 1.104.8 для экзамена 101 на младший уровень администрирования (LPIC-1). Тема имеет пятый уровень значимости.

Из этой темы Вы узнаете о:

- Стандарте Filesystem Hierarchy Standard и классификации файлов и каталогов
- Поиске файлов и команд

### **Стандарт Filesystem Hierarchy Standard**

Стандарт Filesystem Hierarchy Standard – это документ, который определяет структуру каталогов в Linux и других UNIX-подобных системах. Он был создан для формирования общей структуры, которая помогает упростить разработку программного обеспечения, независимого от дистрибутивов, путем расположения файлов на всех дистрибутивах Linux в одни и те же общие каталоги. Этот документ также используется в базе стандартов Linux (см. "[Ресурсы](#)").

### **Две независимые категории стандарта FHS**

В основе стандарта Filesystem Hierarchy Standard лежат две характеристики файлов:

#### **Файлы общего или локального использования**

Файлы общего использования могут быть расположены на одной системе и использоваться на другой, а файлы локального использования должны храниться на той системе, где они используются.

#### **Переменные или статические**

В число статических файлов входит документация, библиотеки и двоичные файлы, которые не изменяются без вмешательства системного администратора. Все остальные файлы являются переменными.

Такое разделение позволяет хранить файлы с разными характеристиками в различных файловых системах. В таблице 6 приведен пример из документа стандарта Filesystem Hierarchy Standard, где показана структура, соответствующая этому стандарту.

*Таблица 6. Пример реализации стандарта Filesystem Hierarchy Standard*

	<b>Общего использования</b>	<b>Локального использования</b>
<b>Статический</b>	/usr	/etc
	/opt	/boot
<b>Переменный</b>	/var/mail	/var/run
	/var/spool/news	/var/lock

### **Корневая файловая система**

Целью стандарта Filesystem Hierarchy Standard является максимально возможное сокращение корневой файловой системы. В ней должны содержаться все файлы, необходимые для загрузки, восстановления или исправления системы, в том числе утилиты, которые могут понадобиться опытным администраторам для выполнения их задач. Следует отметить, что для загрузки системы необходимо, чтобы в корневой файловой системе находились все средства, необходимые для монтирования других файловых систем.

### **Каталоги корневой файловой системы**

В таблице 7 показано назначение каталогов, наличие которых в корневой файловой системе (/) обуславливает стандарт Filesystem Hierarchy Standard. В ней должны присутствовать перечисленные каталоги или символические ссылки на них, за исключением отмеченных как необязательные, которые необходимы только в случаях, когда имеется соответствующая подсистема.

*Таблица 7. Корневая файловая система стандарта Filesystem Hierarchical Standard*

<b>Каталог</b>	<b>Назначение</b>
bin	Двоичные коды необходимых команд
boot	Статические файлы загрузчика
dev	Файлы устройств
etc	Конфигурация системы для этого узла
lib	Необходимые модули ядра и библиотеки общего пользования
media	Точка монтирования для съемных носителей
mnt	Точка временного монтирования файловых систем
opt	Дополнительные пакеты программных приложений
sbin	Двоичные коды необходимых системных файлов
srv	Данные служб, предоставляемых этой системой
tmp	Временные файлы
usr	Дополнительная иерархия
var	Переменные данные
home	Домашние каталоги пользователей (необязательно)
lib<qual>	Важные библиотеки общего пользования в альтернативном формате (необязательно)
root	Домашний каталог пользователя root (необязательно)
<b>/usr и /var</b>	

Иерархии /usr и /var достаточно сложны, для их описания выделены целые разделы стандарта Filesystem Hierarchical Standard. Файловая система /usr является вторым важным разделом файловой системы, в котором содержатся данные только для чтения, находящиеся в общем доступе. Они могут использоваться различными системами, однако в современной практике это делается нечасто. В файловой системе /var содержатся переменные файлы данных, в том числе каталоги и файлы подкачки, данные администрирования и журналов, а также промежуточные и временные файлы. Некоторая часть /var не может использоваться другими системами, однако другие части, например, /var/mail, /var/cache/man, /var/cache/fonts и /var/spool/news, могут находиться в общем доступе.

Для того чтобы полностью понять стандарт Filesystem Hierarchical Standard, ознакомьтесь его описанием (см. раздел "[Ресурсы](#)").

### Где искать файл?

В системах Linux зачастую содержатся сотни тысяч файлов. Только что установленная система Ubuntu, которую мы используем в этом руководстве, насчитывает около 50000 файлов только в иерархии /usr. В системе Fedora, с которой я работал одно время, насчитывается около 175000 файлов. В оставшейся части этого раздела мы рассмотрим инструменты, которые помогут вам найти файлы (в частности, программы) в этом необозримом море данных.

### Ваш PATH

Если вы работали с несколькими системами Linux, вы могли отметить, что если вы входите как пользователь root, вы можете запускать такие команды, как `fdisk`, которые недоступны вам, если вы обычный пользователь. Когда вы запускаете программу в командной строке, командный процессор bash (или любой другой) производит поиск запрошенной вами программы по списку каталогов. Список каталогов указывается в переменной среды PATH, и нет ничего необычного во включении каталога /sbin в путь пользователя root, в то время как для остальных пользователей этого каталога в пути нет. В листинге 64 показаны два различных примера путей пользователя, а также пример пути для пользователя root.

## Листинг 64. Несколько примеров PATH

```
ian@pinguino:~$ echo $PATH  
/  
usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11:/usr/games  
[ian@attic4 ~]$ echo $PATH  
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/ian/bin  
[ian@attic4 ~]$ su -  
Password:  
[root@attic4 ~]# echo $PATH  
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:  
/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
```

Как вы можете видеть, переменная PATH представляет собой просто список наименований каталогов, разделенных двоеточиями. Поскольку команда `fdisk` фактически расположена по пути `/sbin/fdisk`, только первый и последний примеры путей позволяют пользователю запускать эту команду, просто вводя `fdisk`, без указания полного имени (`/sbin/fdisk`).

Как правило, путь устанавливается в файле инициализации, например, в `.bash_profile` или `.bashrc`. Вы можете изменить его для текущей сессии путем указания нового пути. Для того чтобы новое значение было доступно другим процессам, которые вы запускаете, нужно не забывать экспортить переменную PATH. Пример приведен в листинге 65.

## Листинг 65. Изменение переменной PATH

```
[ian@attic4 ~]$ fdisk  
-bash: fdisk: command not found  
[ian@attic4 ~]$ export PATH=/sbin:$PATH  
[ian@attic4 ~]$ fdisk  
  
Usage: fdisk [-l] [-b SSZ] [-u] device  
E.g.: fdisk /dev/hda (for the first IDE disk)  
      or: fdisk /dev/sdc (for the third SCSI disk)  
      or: fdisk /dev/eda (for the first PS/2 ESDI drive)  
      or: fdisk /dev/rd/c0d0 or: fdisk /dev/ida/c0d0 (for RAID devices)  
...
```

### which, type, и whereis

В предыдущем примере мы узнали, что команда `fdisk` недоступна, только когда попытались её запустить. Существует несколько команд, которые помогут вам сделать это.

#### which

Вы можете использовать команду `which` для поиска по вашему пути программы, которая будет запускаться (если будет) при вводе команды. В листинге 66 показан пример поиска команды `fdisk`.

## Листинг 66. Использование команды which

```
[ian@attic4 ~]$ which fdisk  
/usr/bin/which: no fdisk in (/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:  
/usr/X11R6/bin:/home/ian/bin)  
[ian@attic4 ~]$ export PATH=/sbin:$PATH  
[ian@attic4 ~]$ which fdisk
```

```
/sbin/fdisk
```

Команда **which** показывает вам первую найденную команду в вашем пути. Если вы хотите узнать, нет ли нескольких соответствий, вы можете добавить параметр **-a**, как показано в листинге 67.

### Листинг 67. Использование команды **which** для поиска нескольких файлов

```
[root@attic4 ~]# which awk  
/bin/awk  
[root@attic4 ~]# which -a awk  
/bin/awk  
/usr/bin/awk
```

Здесь мы нашли команду **awk** в каталоге **/bin** (который содержит команды, которые могут быть использованы и пользователями, и системным администратором, но не обязательные при отсутствии других смонтированных файловых систем), а также в каталоге **/sbin** (в котором содержатся исполняемые файлы, необходимые для загрузки, восстановления и исправления системы).

### **type**

Существует ряд команд, которые не сможет найти команда **which**. К их числу, например, относятся встроенные команды командной оболочки. Встроенная команда **type** сообщает вам, как поданная вами команда будет оцениваться при выполнении. В листинге 68 показан пример использования команды **type** для самой себя.

### Листинг 68. Использование команды **type**

```
[root@attic4 ~]# which type  
/usr/bin/which: no type in (/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:  
/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin)  
[root@attic4 ~]# type type  
type is a shell builtin
```

### **whereis**

Если вам нужна более подробная информация, а не просто место расположения программы, можно использовать команду **whereis**. Например, можно искать документацию **man** и иную информацию, как показано в листинге 69.

### Листинг 69. Использование команды **whereis**

```
[root@attic4 ~]# whereis awk  
awk: /bin/awk /usr/bin/awk /usr/libexec/awk /usr/share/awk  
/usr/share/man/man1p/awk.1p.gz /usr/share/man/man1/awk.1.gz
```

Заметьте, что копия **awk**, расположенная в каталоге **/sbin**, не была найдена командой **whereis**. Число каталогов, используемых **whereis**, зафиксировано, поэтому команда не

всегда может найти то, что вы ищете. Кроме того, команда `whereis` позволяет выполнять поиск исходных файлов, указывать альтернативные пути поиска, а также искать необычные элементы. Чтобы узнать, как изменить такое поведение или изменить фиксированные пути, используемые `whereis`, обратитесь к документации `man`.

## find

Из руководства к теме 103 "[Подготовка к экзамену LPI 101 \(раздел 103\): команды GNU и UNIX](#)" вы узнали, как искать файлы по имени (в том числе – с использованием подстановочных знаков), пути, размеру и дате создания. В предыдущем разделе, посвященном [жестким и символическим ссылкам](#), вы узнали, как найти ссылки на определенный файл или узел inode.

Команда `find` – своего рода «швейцарский армейский нож» среди инструментов поиска файлов в системах Linux. У нее есть еще две возможности, которые могут быть вам полезны – это поиск файлов по имени пользователя и группы, а также поиск файлов по полномочиям.

В листинге 70 показан листинг каталога для рабочей группы `~greg/xml101`, которую мы уже рассматривали ранее, а также пример того, как можно найти все файлы, владельцем которых является пользователь `ian`, и все файлы, не принадлежащие группе `xml-101`. Заметьте, что в команде `find` восклицательный знак `!` инвертирует критерий поиска.

### Листинг 70. Поиск файлов по пользователю и группе

```
ian@pinguino:~$ ls -l ~greg/xml101/*
-rw-r--r-- 1 greg xml-101 0 2005-12-27 07:38 /home/greg/xml101/file1.c
-rw-r----- 1 greg xml-101 0 2005-12-27 07:39 /home/greg/xml101/file2.c
-rw-r--r-- 1 tom  xml-101 0 2005-12-27 07:41 /home/greg/xml101/file3.c
-rw-r--r-- 1 ian   ian    0 2005-12-27 07:40 /home/greg/xml101/file4.c
-rw-r--r-- 1 tom  xml-101 0 2005-12-27 07:41 /home/greg/xml101/file5.c
-rw-r--r-- 1 ian   xml-101 0 2005-12-27 07:40 /home/greg/xml101/file6.c
-rw-r--r-- 1 tom  xml-101 0 2005-12-27 07:43 /home/greg/xml101/file7.c
-rwxr-xr-x 1 tom  xml-101 0 2005-12-27 07:42 /home/greg/xml101/myprogram
ian@pinguino:~$ find ~greg/xml101 -user ian
/home/greg/xml101/file4.c
/home/greg/xml101/file6.c
ian@pinguino:~$ find ~greg/xml101 ! -group xml-101
/home/greg/xml101/file4.c
```

Чтобы найти файлы по полномочиям, можно использовать критерий `-perm` вместе с символическим выражением, подобным используемому в командах `chmod` и `umask`. Вы можете выполнять поиск по точным полномочиям, однако чаще более полезным является использование в выражении полномочия префикса в виде дефиса, что означает указание искать файлы с установленными данными полномочиями, не обращая внимания на остальные полномочия. В листинге 71 показано с использованием файлов из листинга 70, как найти исполняемые файлы, и приведены два различных способа поиска файлов, которые не могут прочитать другие пользователи.

### Листинг 71. Поиск файлов по полномочиям

```
ian@pinguino:~$ find ~greg/xml101 -perm -ug=x
/home/greg/xml101
/home/greg/xml101/myprogram
ian@pinguino:~$ find ~greg/xml101 ! -perm -o=r
/home/greg/xml101/file2.c
ian@pinguino:~$ find ~greg/xml101 ! -perm -0004
```

```
/home/greg/xml101/file2.c
```

Мы рассмотрели несколько основных видов поиска, которые можно выполнить с помощью команды **find**. Чтобы сузить результаты поиска, вы можете сочетать несколько расширений, а также добавлять регулярные выражения. Чтобы узнать больше об этой разносторонней команде, используйте документацию **man**, или лучше, если у вас установлена система **info**, введите команду **info find**.

В листинге 72 показан пример поиска с помощью команды **find**. В этом примере выполняется переход в каталог **/usr/include**, чтобы сохранить длину листинга в разумных пределах, после чего выполняется поиск всех файлов, содержащих в наименовании пути буквы **xt** с учетом регистра. Во втором примере вывод ограничивается ещё больше, до файлов, которые не являются каталогами и минимальный размер которых составляет 2 килобайта. Информация, которая будет выведена на вашей системе, может отличаться от этой в зависимости от того, какие пакеты у вас установлены.

### Листинг 72. Заключительный пример использования команды **find**

```
ian@pinguino:/usr/include$ find . -iregex ".*xt.*"  
./X11/xterm  
./X11/xterm/ptyx.h  
.irssi/src/fe-common/core/printtext.h  
.irssi/src/fe-common/core/hilight-text.h  
ian@pinguino:/usr/include$ find . -iregex ".*xt.*" ! -type d -size +2k  
./X11/xterm/ptyx.h  
.irssi/src/fe-common/core/printtext.h
```

Следует помнить, что регулярные выражения должны соответствовать полному пути, возвращаемому командой **find**, а также не забывать о разнице между регулярными выражениями и подстановочными символами.

### **locate** и **updatedb**

Команда **find** выполняет поиск по всем каталогам, которые вы укажете, при каждом запуске. Для ускорения процесса вы можете использовать другую команду, **locate**, которая использует базу данных сохраненной информации о путях, а не выполняет всякий раз поиск по файловой системе.

### **locate** и **slocate**

Команда **locate** выполняет поиск файлов по базе данных, которая обычно обновляется ежедневно с помощью задания **cron**. В современных системах Linux эта команда обычно замещается командой **slocate**, которая хранит полномочия вместе с путями, таким образом, не давая пользователям просматривать каталоги, права на просмотр которых у них нет. В этих системах вы можете увидеть, что **locate** является символьской ссылкой на **slocate**, поэтому вы можете использовать любую команду.

Команда **locate** выполняет поиск по любой части пути, а не только по имени файла. В листинге 73 показано, что **locate** является символьской ссылкой на **locate**, после чего приведен пример поиска пути, содержащего строку **bin/ls**.

### Листинг 73. Использование команды **locate**

```
[ian@attic4 ~]$ ls -l $(which locate)
```

```
lrwxrwxrwx 1 root slocate 7 Aug 24 23:04 /usr/bin/locate -> slocate
[ian@attic4 ~]$ locate bin/ls
/bin/ls
/usr/bin/lsb_release
/usr/bin/lskatproc
/usr/bin/lspgpot
/usr/bin/lsattr
/usr/bin/lskat
/usr/bin/lshal
/usr/bin/lsdiff
/usr/sbin/lsof
/sbin/lsmod
/sbin/lsusb
/sbin/lspci
```

## updatedb

База данных, используемая командой **slocate**, хранится в файловой системе /var, в файле /var/lib/slocate/slocate.db. Если вы увидите вывод, похожий на листинг 74 – значит, это задание в вашей системе не выполняется.

### Листинг 74. Отсутствие базы данных slocate

```
[ian@attic4 ~]$ locate bin/ls
warning: locate: could not open database: /var/lib/slocate/slocate.db: No such file or
directory
warning: You need to run the 'updatedb' command (as root) to create the database.
Please edit /etc/updatedb.conf to enable the daily cron job.
```

База данных создается и обновляется с помощью команды **updatedb**. Обычно она выполняется ежедневно как задание cron. Файлом конфигурации updatedb является /etc/updatedb.conf. Для того чтобы включить ежедневное обновление, пользователь root должен изменить файл /etc/updatedb.conf и установить параметр DAILY\_UPDATE=yes. Чтобы создать базу данных, запустите команду **updatedb** под пользователем root.

| [предыдущая](#) |

## Ресурсы

### Научиться

- Чтобы узнать основы Linux и подготовиться к сертификации в качестве системного администратора, ознакомьтесь со всей [серий руководств для подготовки к экзаменам LPI](#).
- В [программе LPIC](#) вы можете найти перечни заданий, примеры вопросов и подробные цели для трех уровней сертификации системных администраторов Linux института Linux Professional Institute.
- Из статьи "[Базовые задачи для новых разработчиков Linux](#)" (developerWorks, март 2005 г) вы узнаете, как открывать окна терминалов и приглашения командной оболочки, а также многое другое.
- В [Докладах симпозиума Linux Symposium, том первый](#) (июль 2005, Оттава, Канада), имеется статья М. ЧАО (M. Cao) и др. "Уровень развития: Что нового в файловой

системе Ext3", содержащая подробную информацию о системе ext3.

- [XFS: высокопроизводительная журналируемая файловая система](#) - домашняя страница проекта XFS на сайте SGI.
- [Reiser4](#) - новая версия файловой системы ReiserFS.
- [qtparted](#) - инструмент работы с разделами с графическим интерфейсом, использующий инструментарий Qt.
- [gparted](#) - инструмент работы с разделами с графическим интерфейсом для GNOME; в нем используется библиотека GTK+GUI.
- На сайте [Проекта документации Linux](#) можно найти множество полезных документов, особенно практические инструкции.
- В руководстве [Quota mini-HOWTO](#) можно найти ответы на вопросы, касающиеся квот.
- Посетите главную страницу стандарта [Filesystem Hierarchy Standard](#) (FHS).
- Из "[Расширенного руководства по реализации файловых систем, Часть 3](#)" вы узнаете подробнее о файловой системе tmpfs, хранящейся в виртуальной памяти.
- [На главной странице LSB](#) можно узнать о базе стандартов Linux (LSB), проекте Группы открытых стандартов (FSG), направленном на разработку стандартной двоичной операционной среды.
- [\*Краткий справочник по сертификации Linux LPI\*](#) (O'Reilly, 2001) и [\*Материалы к экзамену LPIC 1 2: Сертификационные экзамены института Linux Professional Institute 101 и 102 \(материалы к экзамену 2\)\*](#) (Que, 2004) - ссылки для тех, кто предпочитает книжный формат.
- Много других [учебных пособий для разработчиков Linux](#) можно найти в разделе [Linux](#) сайта developerWorks.
- Следите на последними новостями [на портале Web-трансляций и технических новостей developerWorks](#).

## Получить продукты и технологии

- Загрузите [ознакомительные версии программного обеспечения IBM](#) напрямую с developerWorks.

## Обсудить

- [Примите участие в обсуждении материала на форуме](#).
- Примите участие в обсуждении этой статьи на [форуме](#).

# Подготовка к экзамену LPI 101: X Window System

*Администрирование Linux для начинающих (LPIC-1), тема 110*

Ян Шилдс, Старший программист, EMC

**Описание:** Ян Шилдс продолжает готовить вас к сдаче экзамена LPI 101

Профессионального Института Linux (Linux Professional Institute): Администрирование Linux для начинающих (Junior Level Administration). В этом учебнике Ян знакомит вас с X Window System для Linux

[Больше статей из этой серии](#)

**Дата:** 25.01.2007

**Уровень сложности:** средний

## Раздел 1. Предварительные замечания

Узнайте чему может научить вас этот учебник и как извлечь из него максимум.

### Об этой серии учебников

Профессиональный Институт Linux (Linux Professional Institute - LPI) осуществляет сертификацию системных администраторов Linux по двум уровням: для *начинающих* (также называемый "первый уровень сертификации (certification level 1)") и для *специалистов* (также называемый "Второй уровень сертификации (certification level 2)"). Для достижения первого уровня сертификации вы должны сдать экзамены LPI 101 и LPI 102; для достижения второго уровня - экзамены LPI 210 и LPI 202.

developerWorks предоставляет учебники, помогающие в подготовке к каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем и для каждой темы существует соответствующий учебник для самостоятельного изучения на developerWorks. Экзамен LPI 101 содержит пять тем, которым соответствуют учебники от developerWorks:

*Таблица 1. Экзамен LPI 101: Учебники и темы*

Тема экзамена LPI 101	Учебник от developerWorks	Краткое содержание учебника
Тема 101	<a href="#">Учебник для экзамена LPI 101 (тема 101)</a> <a href="#">Аппаратное обеспечение и архитектура</a>	Обучает конфигурированию ваших аппаратных ресурсов в Linux. Из этого учебника, вы узнаете, как Linux конфигурирует устройства, обнаруженные на современном компьютере и где искать решения возникших проблем.
Тема 102	<a href="#">Учебник для экзамена LPI 101:</a> <a href="#">Установка Linux и управление пакетами</a>	Представляет собой введение в установку Linux и управление пакетами. К концу этого учебника вы узнаете, как Linux использует разделы жесткого диска, как Linux загружается, как устанавливать и управлять пакетами программного обеспечения.
Тема 103	<a href="#">Учебник для экзамена LPI 101:</a>	Представляет собой введение в основы GNU и команды UNIX. Из данного учебника вы узнаете,

## GNU и команды UNIX

Тема 104	<a href="#">Учебник для экзамена LPI 104: Устройства, файловые системы Linux, и стандарты иерархии файловых систем [Filesystem Hierarchy Standard].</a>	как использовать команды в командной оболочке bash, включая использование команд текстовых процессоров и фильтров, как искать файлы и каталоги и как управлять процессами.
Тема 110	Учебник для экзамена LPI 101 Система X Window	Содержит информацию о том, как создавать файловые системы на разделах диска, а также как сделать их доступными для пользователей, управлять квотами пользователей и правами доступа к файлам, восстанавливать при необходимости файловую систему. О жестких и символьических ссылках, как располагать файлы в вашей файловой системе и где их следует размещать.

Для сдачи экзаменов LPI 101 и LPI 102 (и достижения первого уровня сертификации), вы должны уметь:

- Работать в командной строке Linux.
- Выполнять простые операции сопровождения: помогать пользователям, управлять учетными записями пользователей, производить резервирование и восстановление информации, а также выключать и перезагружать компьютер.
- Устанавливать и настраивать рабочую станцию (включая X), подсоединяться к локальной сети или подключать отдельно стоящий компьютер в сеть Internet посредством модема

Для продолжения подготовки к сертификации первого уровня смотри [Учебники для экзамена LPI 101 на developerWorks](#). Узнай больше [о полном наборе LPI-учебников на developerWorks](#).

Профессиональный Институт Linux (The Linux Professional Institute) не одобряет любых учебных материалов или технологий для подготовки к экзаменам от третьих лиц. За разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## **Об этом руководстве**

"Знакомство с X Window System" завершает серию из пяти учебных пособий для подготовки к сдаче экзамена LPI 101. Из этого пособия вы узнаете о том, как настраивать X Window System в Linux. Данное руководство охватывает обе основные реализации X для Linux: XFree86 и X.Org.

Это учебное пособие организовано в соответствии с программой экзамена LPI по данной теме. Следует ожидать большее число вопросов по темам, имеющим более высокий рейтинг.

*Таблица 2. X Window System: экзаменационные темы, рассмотренные в этом учебном пособии*

### **Тема экзамена**

LPI	Рейтинг	Краткое содержание темы
1.110.1 <a href="#">Установка и настройка X</a>	5	Конфигурирование и установка X и сервера шрифтов. Как убедиться в том, что X-сервер поддерживает ваши видеоадаптер и монитор. Тонкая настройка монитора и видеокарты. Установка сервера шрифтов, установка шрифтов и конфигурирование X для использования

		сервера шрифтов.
1.110.2 <a href="#"><u>Настройка менеджера экрана</u></a>	3	Настройка менеджера экрана. Включение и выключение менеджера экрана, изменение его приветствия и числа отображаемых цветов. Настройка менеджеров экрана для работы с GNOME и KDE.
1.110.4 <a href="#"><u>Установка и настройка среды управления окнами</u></a>	5	Общесистемная настройка графической оболочки и оконного менеджера, включая его меню и панельные меню виртуального рабочего стола. Выбор и настройка эмулятора терминала, проверка и решение проблем с зависимостью от различных библиотек для X-приложений. Передача X-экрана на рабочую станцию клиента.

## Предварительные требования

Для максимально эффективного использования этого руководства вы должны обладать базовыми знаниями по Linux и иметь рабочую систему Linux для отработки показанных здесь команд. Желательно чтобы вы свободно работали с приложениями, имеющими графический интерфейс, предпочтительно под управлением X Window System.

Это учебное пособие базируется на материале предыдущих четырёх руководств и возможно вы захотите [освежить их в памяти](#).

Разные версии программ могут по-разному форматировать выходные данные, поэтому полученные вами результаты, могут выглядеть не совсем так, как в листингах и на рисунках из этого учебника.

# Подготовка к экзамену LPI 101: X Window System

*Администрирование Linux для начинающих (LPIC-1), тема 110*

[Ян Шилдс](#), Старший программист, EMC

**Описание:** Ян Шилдс продолжает готовить вас к сдаче экзамена LPI 101

Профессионального Института Linux (Linux Professional Institute): Администрирование Linux для начинающих (Junior Level Administration). В этом учебнике Ян знакомит вас с X Window System для Linux

[Больше статей из этой серии](#)

**Дата:** 25.01.2007

**Уровень сложности:** средний

## Раздел 2. Установка и настройка X

Этот раздел охватывает материал по теме 1.110.1 экзамена «Администрирование Linux для начинающих». Рейтинг темы – 5.

В данном разделе вы узнаете:

- как убедиться в том, что X-сервер поддерживает ваш монитор и видеокарту;
- как устанавливать и конфигурировать X-среду;

- как настраивать X-среду под вашу видеокарту и монитор;
- как конфигурировать и устанавливать сервер шрифтов;
- как устанавливать шрифты.

## История X Window System

X Window System называемая также просто X или X11 – оконная среда для графических (растровых) дисплеев. Начало X было положено в Массачусетском технологическом институте (MIT) в 1984 году. X была реализована как часть проекта Афина (Project Athena), предоставлявшего вычислительную среду, функционирующую на разнотипном оборудовании. В X-среде за вывод информации отвечает *сервер экрана* (display server), а логику приложения предоставляют *клиенты*. Взаимодействие между ними является "прозрачным" для сети, поэтому сервер и клиент могут работать на разных машинах. Следует отметить, что термины "клиент" и "сервер" несколько отличаются от обыденного представления. Помимо вывода информации, сервер обрабатывает ввод информации от различных устройств, таких как клавиатуры, мыши, графических планшетов и сенсорных экранов.

X-среда предоставляет набор средств для приложений с графическим интерфейсом, но не определяет конкретный интерфейс пользователя. В Linux обычно можно выбирать между графическими оболочками KDE и GNOME, а также несколькими другими оконными менеджерами. Поскольку X не определяет интерфейс пользователя, то эти среды и оконные менеджеры выглядят по-разному.

X разрабатывалась для большого сообщества пользователей, имеющих различные типы оборудования, поэтому разные версии X-клиентов и серверов обычно неплохо взаимодействуют между собой.

## XFree86 and X.Org

К 1987 году MIT решил передать управление разработкой X отдельной организации. В результате был создан MIT X Consortium – некоммерческая группа по надзору за разработкой X-среды. После ещё нескольких изменений в управлении Open Group в 1999-м сформировал X.Org. С 1992 года большая часть разработки X выполнялась в XFree86, первоначально перенесшей X на платформу Intel® 386, откуда и название. Организация XFree86 присоединилась к X.Org в качестве члена, освобожденного от уплаты взносов.

Хотя первоначально XFree86 был создан для платформы 386, последующие версии поддерживали несколько различных платформ и стали наиболее широко используемым вариантом X-среды в Linux. После ряда споров о новых условиях лицензирования и модели разработки XFree86 была создана X.Org Foundation. Отталкиваясь от последней версии XFree86 с предыдущей лицензией X.Org создала X11R6.7 и X11R6.8. Многие дистрибутивы до сих пор используют XFree86, в то время как многие выбрали X.Org.

## Поддержка видео оборудования

Оба пакета XFree86 и X.Org поддерживают широкий спектр современных видеокарт. Обратитесь к онлайновой документации по вашему релизу (см. [Ресурсы](#)). Некоторые производители не выпускают драйверы с открытым исходным кодом для всех целей использования, так что вам может потребоваться интегрировать драйвер от производителя в вашу XFree86-среду. За усовершенствованными или обновленными драйверами для Linux обратитесь на Web-сайт производителя. Это часто необходимо для 3D ускорителей. Но даже если аппаратные возможности вашей видеокарты не могут быть использованы XFree86, вероятно, вы сможете запустить её в режиме VESA (Video Electronics Standards Association ассоциация по стандартизации в области видеотехники) с буфером кадров.

Современные мониторы реализуют протокол VESA *Display Data Channel (DDC)* (Канал Отображения Данных), который позволяет программно определять информацию о мониторе

и его характеристиках. Средства конфигурации XFree86 (отличные от `xf86config`) используют эту информацию для настройки вашей X- среды.

Одним из способов посмотреть на то, как X-среда работает на вашем оборудовании является загрузка с live CD, например Knoppix или Ubuntu. Эти дистрибутивы имеют хорошие возможности по определению и использованию вашего оборудования. Многие дистрибутивы предлагают графический интерфейс установки, что также требует корректного определения и использования вашего оборудования.

## XFree86

Большинство дистрибутивов включают XFree86 или X.Org. Если нужных пакетов нет, то вы можете найти RPM или .deb пакеты и установить их, пользуясь знаниями, полученными в теме 102 "[Подготовка к экзамену LPI 101: установка Linux и управление пакетами.](#)"

### Установка XFree86

Если у вас нет пакета XFree86, то вам придётся загрузить файлы с Web-сайта проекта XFree86 (см. [Ресурсы](#)). Имеются собранные пакеты для Linux для нескольких популярных платформ или вы можете произвести установку из исходных кодов. В этом руководстве предполагается, что вы устанавливаете текущий релиз (версия 4.5.0) из скомпилированного пакета.

Вам потребуется загрузить несколько двоичных пакетов. Для проверки загруженных файлов воспользуйтесь контрольными суммами md5 или ключами GPG. В таблице 3 приведен список файлов, необходимых для XFree86.

*Таблица 3. Необходимые файлы для XFree86*

Файл	Описание
Xinstall.sh	Скрипт установки
extract	Утилита распаковки Tarball
Xbin.tgz	Х-клиенты, утилиты и библиотеки времени выполнения (run-time libraries)
Xlib.tgz	Файлы данных, необходимых во время выполнения
Xman.tgz	Интерактивное справочное руководство (ман страницы)
Xdoc.tgz	Документация по XFree86
Xfnts.tgz	Базовый набор шрифтов
Xfenc.tgz	Данные о кодировках шрифтов
Xetc.tgz	Файлы конфигурации времени выполнения - часть 1
Xrc.tgz	Файлы конфигурации времени выполнения - часть 2
Xvar.tgz	Данные времени выполнения
Xxserv.tgz	XFree86 X-сервер
Xmod.tgz	Модули X-сервера

Для определения версии необходимых файлов, используйте Xinstall.sh с параметром `-check` из того пакета, который вам кажется наиболее подходящим как показано в листинге 1.

### Листинг 1. Определение правильного двоичного пакета XFree86

```
root@pinguino:~/xfree86# sh Xinstall.sh -check
Checking which OS you're running...
uname reports 'Linux' version '2.6.12-10-386', architecture 'i686'.
libc version is '6.3.5' (6.3).
```

```
Binary distribution name is 'Linux-ix86-glibc23'  
If you don't find a binary distribution with this name, then  
binaries for your platform are not available from XFree86.org.
```

В этом примере вы должны выбрать пакет с названием "Linux-ix86-glibc23".

В таблице 4 перечислены дополнительные файлы для XFree86. При работе с учебником вам потребуется сервер шрифтов и любые другие компоненты, которые вы захотите установить.

*Таблица 4. Дополнительные файлы для XFree86*

Файл	Описание
Xdrm.tgz	Исходные коды модулей ядра менеджера прямого рендеринга (DRM)
Xfsrv.tgz	Сервер шрифтов
Xnest.tgz	Вложенный X-сервер
Xprog.tgz	X заголовочные файлы, конфигурационные файлы и библиотеки для разработки X-приложений
Xprt.tgz	X сервер печати
Xvfb.tgz	X-сервер виртуального буфера кадров
Xtinyx.tgz	TinyX сервера
Xf100.tgz	Шрифты с разрешением 100dpi
Xfcyr.tgz	Кириллические шрифты
Xfscl.tgz	Масштабируемые шрифты (Speedo, Type1, and TrueType)
Xhtml.tgz	Документация в формате HTML
Xps.tgz	Документация в формате PostScript
Xpdf.tgz	Документация в формате PDF
Перед установкой XFree86 сделайте резервные копии каталогов /usr/X11R6, /etc/X11, и /etc/fonts поскольку их содержимое может измениться в ходе установки. Для этого вы можете воспользоваться командами <a href="#">tar</a> , <a href="#">cp</a> или <a href="#">zip</a> . Когда вы будете готовы к установке XFree86 перейдите в каталог, в который вы загрузили файлы XFree86 и запустите скрипт Xinstall.sh как показано в листинге 2. Вас попросят ответить на несколько вопросов, перечень которых может варьироваться в зависимости от того была ли установлена X-среда ранее или нет. После установки основных компонентов вас спросят об установке дополнительных компонентов индивидуально.	

## Листинг 2. Установка XFree86

```
root@pinguino:~/xfree86# sh Xinstall.sh
```

Вас попросят ответить на несколько вопросов, перечень которых может варьироваться в зависимости от того была ли установлена X-среда ранее или нет. После установки основных компонентов вас спросят об установке дополнительных компонентов индивидуально.

После установки файлов скрипт выполнит команду [ldconfig](#) и предложит вам установить несколько символических ссылок.

Простейший способ установки XFree86 – установка всех желаемых компонентов используя Xinstall.sh. В противном случае, вам придётся либо переустанавливать весь пакет, что может привести к потере любых сделанных вами изменений в настройках, либо вручную устанавливать остальные компоненты.

## Настройка XFree86

Исторически конфигурирование XFree86 включало создание файла XF86Config, который содержал информацию о видеокарте, мыши, клавиатуре и мониторе наряду с настроочными параметрами, например, желаемом разрешении экрана. Первоначальное средство конфигурирования **xf86config** требовало от пользователя знания и ввода подробной информации о частотах синхронизации видеокарты и монитора. Последние версии XFree86 способны динамически определять доступное аппаратное обеспечение и могут работать с небольшим количеством конфигурационной информации или без неё.

Доступны следующие средства конфигурирования.

### XFree86 -autoconfig

Запуск XFree86 с параметром **-autoconfig** приведет к попытке автоматически настроить X-сервер. Если настройки определены правильно, то у вас появится возможность перемещать X-курсор по экрану. Удерживая одновременно клавиши **Ctrl** и **Alt** нажмите **Backspace** для выхода из экрана. Этим вы подтверждаете, что автоконфигурация будет работать. Конфигурационный файл не записывается.

### XFree86 -configure

Запуск XFree86 с параметром **-configure** может работать если не работает **-autoconfig**. Этот параметр может вызвать проблемы в некоторых системах.

### xf86cfg

Команда **xf86cfg** пытается запустить драйверы дисплея и устройств ввода. Если это удастся, то вы увидите окно с диаграммой вашей системы. Для просмотра и изменения настроек конкретного элемента щелкните по нему правой кнопкой мыши. В некоторых системах вам потребуется использовать цифровую клавиатуру вместо мыши, поскольку мышь не была корректно определена. Перед запуском **xf86cfg** вы можете захотеть создать символическую ссылку на `/dev/mouse` с вашего устройства ввода - мышь.

Например,

```
ln -s /dev/input/mice /dev/mouse.
```

Когда вы нажмёте **Quit**(Выход), вам предложат сохранить конфигурационные файлы `/etc/X11R6/lib/X11/XF86Config` и `/etc/X11R6/lib/X11/xkb/X0-config.keyboard`.

### xf86config

Команда **xf86config** использует текстовый интерфейс для интерактивного запроса информации о вашей мыше, клавиатуре, видеокарте и дисплее. Вам потребуется информация о частотах горизонтальной и вертикальной развёртки вашего монитора. Вы можете выбрать большинство видеокарт и базы данных по известным видеокартам. Если сделать это не удалось, то вам может потребоваться специфическая информация о наборе микросхем и частотах синхронизации вашей видеокарты.

## Замечания:

- Если ваша система включает XFree86, то, возможно, ваш поставщик включил средство типа **sax2**, используемое в SUSE или **redhat-config-xfree86** для некоторых систем Red Hat®. Всегда просматривайте документацию по вашей системе о подобных средствах.
- Другое средство конфигурирования **XF86Setup** более не распространяется с Xfree86.

## X.Org

Большинство дистрибутивов включают пакеты XFree86 или X.Org. Если в вашем дистрибутиве их нет, то вы можете отыскать RPM или .deb пакет и установить его, используя навыки, полученные при изучении темы 102 "[LPI exam 101 prep: Linux installation and package management](#)".

### Установка X.Org

Если необходимый пакет X.Org недоступен, то вам потребуется загрузить и собрать его из исходных кодов доступных на Web-сайте X.Org или его зеркалах (см. [Ресурсы](#)). В момент написания учебника, эти сайты не содержали собранных двоичных пакетов для X11R6.9.0 или X11R7.0. Исходный код доступен из репозитория CVS(система управления версиями) или архивов, сжатых gzip либо bzip2. Вам необходимо получить файлы gz или bz2, но не то и другое одновременно. Вы обнаружите, что *X.Org Modular Tree Developer's Guide* (см. [Ресурсы](#)) является неоценимым помощником при загрузке и самостоятельной сборке X.Org. Обратите внимание на то, что для полнофункциональной сборки рекомендуется использовать дополнительные пакеты такие как freetype, fontconfig, и Mesa

### Конфигурирование X.Org

Пакет X.Org основан на последней версии XFree86 и имеет похожие конфигурационные возможности, включая динамическое определение доступного аппаратного обеспечения. Конфигурационный файл предпочтительнее именовать xorg.conf, а не XF86Config. Вы можете его найти в нескольких местах /etc/xorg.conf, /etc/X11/xorg.conf, /usr/X11R6/etc/xorg.conf, /usr/X11R6/lib/X11/xorg.conf.hostname, или /usr/X11R6/lib/X11/xorg.conf.

Доступны следующие средства конфигурирования:

#### X -configure

Запуск X с параметром **-configure** заставляет X-сервер загружать каждый драйверный модуль, пробовать драйвер и создавать конфигурационный файл, который сохраняется в домашнем каталоге пользователя, запустившего сервер (обычно /root). Файл называется xorg.conf.new.

#### xorgcfg

Это средство подобно xf86cfg

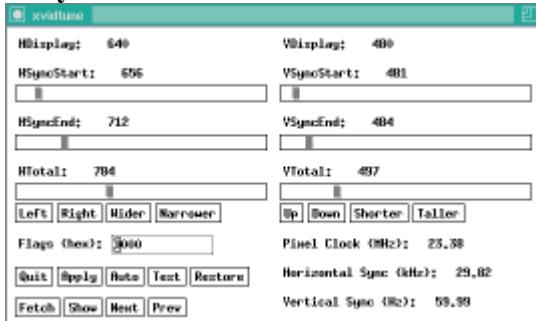
#### xorg86config

Команда **xorgconfig** текстовый интерфейс для интерактивного закрашивания информации о вашей мыши, клавиатуре, видеоадаптере и дисплее. Как и при использовании xf86config вам необходимо знать частоты горизонтальной и вертикальной развертки вашего дисплея. Вы можете выбрать большинство видеоадаптеров из базы данных известных видеокарт. Если этого не удается сделать, то вам потребуется информация о наборе микросхем и частотах синхронизации для вашего видеоадаптера.

## Настройка X

Современные многочастотные ЭЛТ мониторы обычно имеют элементы управления для задания размера изображения и его позиции на экране. Если ваш монитор не имеет этой возможности, то вы можете использовать команду **xvidtune** для настройки размера и положения вашего X-экрана. При запуске **xvidtune** из сеанса виртуального терминала, вы увидите окно, похожее на изображенное на рисунке 1. Откорректируйте настройки и нажмите **Test** (Тест) для того, чтобы посмотреть как они работают или **Apply** (Применить) для изменения параметров. Если вы нажмете **Show** (Показать) текущие настройки будут выведены в ваше терминальное окно в том формате, который вы можете использовать как настройку Modeline в файлах CF86Config или xorg.conf.

**Рисунок 1: Работа xvidtune**



За более подробной информацией обратитесь к интерактивной справке.

### Общая информация о шрифтах в X.

Долгие годы за работу со шрифтами отвечала *встроенная система шрифтов X11 (core X11 fonts system)*. Последние версии X-серверов XFree86 (и X.Org) включают *систему шрифтов Xft*. Встроенная система шрифтов первоначально поддерживала монохромные растровые шрифты, но с течением времени была усовершенствована. Система Xft была разработана с учетом современных требований, включая сглаживание и субпиксельное растирование (sub-pixel rasterization) и позволяет приложениям иметь всесторонний контроль над рендерингом глифов. Основное отличие между этими двумя системами шрифтов состоит в том, что встроенная система работает на сервере, а Xft шрифты обрабатываются клиентами, которые отсылают необходимые глифы серверу.

X первоначально использовала шрифты Type 1 (или Adobe Type 1) – формат описания шрифтов, разработанный Adobe. Система Xft может работать с ними наряду со шрифтами OpenType, TrueType, Speedo и CID.

### Сервер шрифтов xfs

Со встроенной системой шрифтов X11 X Server получает шрифты и информацию о них от *сервера шрифтов*. Сервер шрифтов **xfs** обычно запускается как демон при старте системы, хотя возможно запустить его как обычную задачу. Как правило вы будете устанавливать сервер шрифтов в ходе установки X. Тем не менее, поскольку X является сетевым протоколом, имеется возможность получать шрифты и информацию о них через сеть, а не с вашей локальной машины.

X-сервер шрифтов использует обычно конфигурационный файл /usr/X11R6/lib/X11/fs/config. Пример файла конфигурации шрифтов показан в листинге 3. Конфигурационный файл может быть также расположен в или связан с /etc/X11/fs.

### Листинг 3. Пример /usr/X11R6/lib/X11/fs/config

```
# allow a max of 10 clients to connect to this font server
client-limit = 10

# when a font server reaches its limit, start up a new one
clone-self = on

# alternate font servers for clients to use
#alternate-servers = foo:7101,bar:7102

# where to look for fonts
#
catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,
            /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
```

```

/usr/X11R6/lib/X11/fonts/100dpi:unscaled,
/usr/X11R6/lib/X11/fonts/misc,
/usr/X11R6/lib/X11/fonts/Type1,
/usr/X11R6/lib/X11/fonts/Speedo,
/usr/X11R6/lib/X11/fonts/cyrillic,
/usr/X11R6/lib/X11/fonts/TTF,
/usr/share/fonts/default/Type1

# in 12 points, decipoints
default-point-size = 120

# 100 x 100 and 75 x 75
default-resolutions = 75,75,100,100

# how to log errors
use-syslog = on

# don't listen to TCP ports by default for security reasons
no-listen = tcp

```

Этот пример типичен для установки Linux на рабочую станцию где сервер шрифтов не предоставляет шрифты по TCP-соединениям (`no-listen = tcp`).

## Библиотека Xft

Библиотека Xft предоставляет функции, позволяющие клиентским приложениям выбирать шрифты по заданному образцу и генерировать глифы для отправки их на сервер. Образцы учитывают семейство шрифтов (Helvetica, Times и тд.), кегль, начертание и множество других характеристик. В то время как встроенная система шрифтов позволяет клиенту найти лишь первый подходящий шрифт на сервере, Xft находит лучший шрифт по всем критериям и затем отсылает информацию о глифах на сервер. Xft взаимодействует с FreeType для создания картинки из глифа и расширением Render сервера X, ускоряющим процесс рендеринга. Xft входит в состав текущих версий как XFree86 так и X.Org.

**Замечание:** Если ваш X-сервер работает через сеть и используется видеокарта не поддерживающая расширение Render, то производительность сети может стать проблемой в данной ситуации и вы можете захотеть отключить сглаживание. Вы можете использовать команду [xdpyinfo](#) для просмотра информации о вашем X-сервере. Листинг 4 содержит часть информации выводимой [xdpyinfo](#). Поскольку объём информации, создаваемой [xdpyinfo](#) велик, можно воспользоваться командой [grep](#) для поиска 'RENDER'.

## Листинг 4. Проверка наличия расширения RENDER программой xdpyinfo

```
[ian@lyrebird ian]$ xdpyinfo
name of display:      :0.0
version number:      11.0
vendor string:        The XFree86 Project, Inc
vendor release number: 40300000
XFree86 version:     4.3.0
maximum request size: 4194300 bytes
motion buffer size:   256
bitmap unit, bit order, padding:    32, LSBFirst, 32
image byte order:      LSBFirst
number of supported pixmap formats: 7
supported pixmap formats:
    depth 1, bits_per_pixel 1, scanline_pad 32
    depth 4, bits_per_pixel 8, scanline_pad 32
```

```
depth 8, bits_per_pixel 8, scanline_pad 32
depth 15, bits_per_pixel 16, scanline_pad 32
depth 16, bits_per_pixel 16, scanline_pad 32
depth 24, bits_per_pixel 32, scanline_pad 32
depth 32, bits_per_pixel 32, scanline_pad 32
keycode range: minimum 8, maximum 255
focus: window 0x2000011, revert to Parent
number of extensions: 30
    BIG-REQUESTS
    DOUBLE-BUFFER
    DPMS
    Extended-Visual-Information
    FontCache
    GLX
    LBX
    MIT-SCREEN-SAVER
    MIT-SHM
    MIT-SUNDRY-NONSTANDARD
    RANDR
    RECORD
RENDER
    SECURITY
    SGI-GLX
    SHAPE
    SYNC
    TOG-CUP
    X-Resource
    XC-APPGROUP
    XC-MISC
    XFree86-Bigfont
    XFree86-DGA
    XFree86-DRI
    XFree86-Misc
    XFree86-VidModeExtension
    XInputExtension
    XKEYBOARD
    XTEST
    XVideo
default screen number: 0
number of screens: 1
```

Использование Xft вместо встроенной системы шрифтов X требует внесения изменений в приложения, поэтому вы можете обнаружить, что некоторые приложения не используют преимуществ улучшенного рендеринга шрифтов в Xft. В момент написания этого учебника примерами приложений, использующих Xft являются Qt (используется в KDE), GTK+ (используется в GNOME), Mozilla 1.2.

## **Установка шрифтов**

Существует два метода установки шрифтов один для Xft и более сложный для встроенной системы шрифтов X11.

### **Шрифты для Xft**

Xft использует шрифты, расположенные в ряде хорошо известных каталогов шрифтов, также как и в подкаталоге .fonts домашнего каталога пользователя. Хорошо известные каталоги шрифтов включают подкаталоги /usr/X11R6/lib/X11/lib/fonts, как перечислено в разделе каталогов в /usr/X11R6/lib/X11/fs/config. Другие каталоги шрифтов могут быть заданы в разделе FontPath файлов XF86Config или xorg.conf (в зависимости от используемого вами

X-сервера).

Просто скопируйте ваши шрифты в пользовательский каталог .fonts или для использования во всей системе в /usr/local/share/fonts. Сервер шрифтов должен выбрать новые шрифты и сделать их доступными при следующем запуске. Вы можете провести обновление без перезапуска сервера командой **fc-cache**.

Текущая технология работы со шрифтами в X использует загружаемые модули для поддержки различных типов шрифтов как показано в таблице 5.

*Таблица 5. Модули работы со шрифтами X-сервера*

Модуль	Описание
bitmap	Растровые шрифты (.bdf, .pcf, and .snf)
freetype	TrueType (.ttf and .ttc), OpenType (.otf and .otc) и шрифты Type 1 (.pfa and .pfb)
type1	Альтернативная поддержка Type 1 (.pfa and .pfb) и CID шрифтов.
xtt	Альтернативный модуль TrueType (.ttf and .ttc)
speedo	Шрифты Speedo(.spd)

Если у вас возникли проблемы с установкой и использованием шрифта, то проверьте журнал сервера (например /var/log/XFree86.0.log) чтобы убедиться в том, что соответствующий модуль был загружен. Имена модулей чувствительны к регистру. Для просмотра (и изменения) настроек X-сервера, включая путь к шрифтам, расположение конфигурационных файлов и журналов вы можете использовать команду **xset** как показано в листинге 5.

### Листинг 5. Отображение настроек X-сервера командой xset

```
[ian@lyrebird ian]$ xset -display 0:0 -q
Keyboard Control:
  auto repeat: on    key click percent: 0    LED mask: 00000000
  auto repeat delay: 500   repeat rate: 30
  auto repeating keys: 00ffffffffffffbbff
                        fadfffffffffe5ff
                        ffffffffffffffff
                        ffffffffffffffff
  bell percent: 50    bell pitch: 400    bell duration: 100
Pointer Control:
  acceleration: 2/1    threshold: 4
Screen Saver:
  prefer blanking: yes    allow exposures: yes
  timeout: 0    cycle: 0
Colors:
  default colormap: 0x20    BlackPixel: 0    WhitePixel: 16777215
Font Path:
  /home/ian/.gnome2/share/cursor-fonts,unix/:7100,/home/ian/.gnome2/share/fonts
Bug Mode: compatibility mode is disabled
DPMS (Energy Star):
  Standby: 7200    Suspend: 7200    Off: 14340
  DPMS is Enabled
  Monitor is Off
Font cache:
  hi-mark (KB): 5120  low-mark (KB): 3840  balance (%): 70
File paths:
  Config file: /etc/X11/XF86Config
  Modules path: /usr/X11R6/lib/modules
```

```
Log file:      /var/log/XFree86.0.log
```

Если вам требуется дополнительный контроль поведения Xft, вы можете использовать либо общесистемный (/etc/fonts/fonts.conf) или пользовательский (.fonts.conf в домашнем каталоге пользователя) конфигурационный файл. Кроме прочего, вы можете включить или отключить сглаживание и управлять субпиксельным рендерингом (используется в ЖК-дисплеях). Это XML файлы поэтому вы должны убедиться в том, что после редактирования они остались корректными. За дополнительной информацией о содержании и формате этих файлов обратитесь к интерактивной справке.

## Встроенные шрифты X11

Перед установкой шрифтов в формате Bitmap Distribution Format (.bdf) (двоичный формат распространения) желательно преобразовать их в Portable Compiled Format (.pcf) (переносимый скомпилированный формат) и сжать их, используя [gzip](#). После этого, вы можете скопировать новые шрифты в каталог, например, /usr/local/share/fonts(bitmap/ и затем выполнить команду [mkfontdir](#) для создания каталога шрифтов, который будет использоваться сервером. Эти шаги показаны в листинге 6.

### Листинг 6. Установка растровых шрифтов

```
[root@lyrebird root]# bdftopcf courier12.bdf -o courier12.pcf
[root@lyrebird root]# gzip courier12.pcf
[root@lyrebird root]# mkdir -p /usr/local/share/fonts(bitmap/
[root@lyrebird root]# cp *.pcf.gz /usr/local/share/fonts(bitmap/
[root@lyrebird root]# mkfontdir /usr/local/share/fonts(bitmap/
[root@lyrebird root]# ls /usr/local/share/fonts(bitmap/
courier12.pcf.gz  fonts.dir
```

Обратите внимание на то, что команда [mkfontdir](#) создаёт файл fonts.dir

Для установки масштабируемых шрифтов типа TrueType или Type1 требуется дополнительный шаг. После копирования файлов шрифтов в целевой каталог выполните команду [mkfontscale](#) а затем [mkfontdir](#). Команда [mkfontscale](#) создаст перечень масштабируемых шрифтов в файле fonts.scale.

Теперь, когда вы задали каталог шрифтов и информацию по их масштабированию вы должны указать серверу, где искать новые шрифты. Это делается включением нового каталога в путь поиска шрифтов. Вы можете сделать это на временной (используя [xset](#)) или постоянной (включением записи FontPath в файл XF86Config или xorg.conf) основе. Для того, чтобы добавить новый каталог растрового шрифта в начало списка поиска шрифтов воспользуйтесь параметром [+fp](#) команды [xset](#), как показано в листинге 7.

### Листинг 7. Обновление пути поиска шрифтов командой xset

```
[ian@lyrebird ian]$ xset +fp /usr/local/share/fonts(bitmap/ -display 0:0
```

Хорошей идеей (хотя это здесь не показано) является включение масштабируемых шрифтов перед растровыми, поскольку это приводит к лучшему их подбору. Для добавления каталогов в конец списка используйте параметр [fp+](#). Аналогично параметры [-fp](#) и [fp-](#) приводят к удалению каталогов из начала и конца списка соответственно.

Вы можете сделать изменения постоянными, отредактировав XF86Config или xorg.conf. Вы можете добавить столько строк FontPath в раздел Files, сколько необходимо как показано в листинге 8.

#### Листинг 8. Изменение XF86Config или xorg.conf

```
Section "Files"
# RgbPath is the location of the RGB database. Note, this is the name of the
# file minus the extension (like ".txt" or ".db"). There is normally
# no need to change the default.

# Multiple FontPath entries are allowed (they are concatenated together)
# By default, Red Hat 6.0 and later now use a font server independent of
# the X server to render fonts.

    RgbPath      "/usr/X11R6/lib/X11/rgb"
    FontPath     "unix/:7100"
    FontPath     "/usr/local/share/fonts/bitmap/"
EndSection
[
```

Информацию о том, что ещё вы можете изменить в конфигурационных файлах X смотрите в интерактивном руководстве по XF86Config или xorg.conf.

### Раздел 3. Настройка менеджера экрана

В этом разделе рассматриваются вопросы темы 1.110.2 экзамена 101 "Начальный уровень администрирования (LPIC-1)". Тема имеет рейтинг 3.

В этом разделе вы узнаете как:

- настраивать менеджер экрана;
- изменять приветствие менеджера экрана;
- изменять глубину цвета для менеджера экрана;
- конфигурировать менеджеры экрана для использования X-станциями

Рассматриваются следующие менеджеры экрана: XDM (X Display Manager), GDM (GNOME Display Manager) и KDM (KDE Display Manager).

#### Менеджеры экрана

Если в предыдущем разделе вы установили и сконфигурировали X на компьютере, на котором ранее X отсутствовал, то вы, вероятно, заметили, что для доступа к любым графическим экранам приходиться регистрироваться в окне виртуального терминала и выполнять команду **startx**. Это неудобно даже для локального экрана, а для удалённого терминала не работает вообще.

Решение этой проблемы состоит в использовании *менеджера экрана* для предоставления графического окна регистрации и управления аутентификацией. После того, как пользователь аутентифицирован, менеджер экрана открывает для пользователя сессию на той системе, где он выполняется. Графический вывод производится на том экране, в котором пользователь ввел свои данные регистрации. Это может быть как локальный дисплей, так и X-дисплей, подключенный через сеть. И Xfree86 и X.Org поставляются с менеджером экрана XDM.

Есть ещё два популярных менеджера экрана KDE и GNOME. В этом разделе вы узнаете как устанавливать и настраивать все эти три менеджера экрана.

Для настройки графического окна регистрации вам необходимо понимать процесс инициализации Linux. Подробнее об этом процессе вы можете узнать из готовящегося пособия для подготовки к экзамену LPI 102 (тема 106): Загрузка, инициализация, выключение и уровни выполнения и в [учебнике для подготовки к экзамену LPI 201\(тема 202\): Запуск системы](#). В оставшейся части этого раздела вы получите достаточно информации для запуска системы с графическим экраном регистрации, но основное внимание будет уделено установке и настройке менеджера экрана.

В Red Hat® и SUSE X запускается с уровнем выполнения 5. Debian рассматривает уровни выполнения со 2 по 5 как эквивалентные и по умолчанию использует уровень 2. Определение уровня выполнения производится в файле /etc/inittab как показано в листинге 9.

#### **Листинг 9. Установка уровня выполнения в /etc/inittab.**

```
# The default runlevel is defined here  
id:5:initdefault:
```

Ещё одна строка показанная в листинге 10 (для SUSE) или листинге 11(для Ubuntu) определяет какая программа или скрипт должна выполняться первой.

#### **Листинг 10. Начальный скрипт для SUSE (или Red Hat)**

```
# First script to be executed, if not booting in emergency (-b) mode  
si::bootwait:/etc/init.d/boot
```

#### **Листинг 11. Начальный скрипт для Ubuntu (или Debian)**

```
# Boot-time system configuration/initialization script.  
# This is run first except when booting in emergency (-b) mode.  
si::sysinit:/etc/init.d/rcS
```

Скрипты инициализации (/etc/init.d/boot или /etc/init.d/rcS) далее запустят другие скрипты. В итоге, будет запущен набор скриптов для заданного уровня выполнения. Для приведённых примеров это набор мог включать скрипты etc/rc2.d/S13gdm (Ubuntu) или /etc/init.d/rc5.d/S16xdm (SUSE), предназначенные для запуска менеджера экрана. Вы обнаружите, что каталоги rcn.d и /etc/init.d обычно содержат символические ссылки на скрипты в /etc/init.d и префиксом S(или K) и числом. S означает, что скрипт должен выполняться при входе на уровень выполнения, а K – при завершении уровня выполнения. Цифры от 1 до 99 определяют порядок выполнения скриптов.

**Подсказка:** Если вы пытаетесь понять, как запускается менеджер экрана, то обратите внимание на скрипты, имена которых заканчиваются на **dm**.

Вы можете обнаружить, что скрипт для запуска менеджера экрана, скажем /etc/init.d/rc5.d/S16xdm, может быть короткой программой, содержащей дополнительную логику для определения того, какой менеджер экрана на самом деле будет запущен. Так что, хотя многие системы позволяют это делать через настройки, вы также можете определить, какой менеджер экрана будет запущен изучая ваши файлы инициализации.

Вас не должно удивлять, что для управления запуском менеджера экрана достаточно

символических ссылок в соответствующем каталоге rcn.d. Более того, если вам необходимо остановить или запустить менеджер экрана, вы можете напрямую использовать скрипт из каталога /etc/init.d как показано в листинге 12.

### Листинг 12. Остановка и запуск менеджера экрана

```
root@pinguino:~# /etc/init.d/gdm stop
 * Stopping GNOME Display Manager... [ ok ]
root@pinguino:~# /etc/init.d/gdm start
 * Starting GNOME Display Manager... [ ok ]
```

Теперь когда вы знаете как запускать и останавливать менеджер экрана, давайте обратимся к вопросу конфигурирования каждого из этих трёх менеджеров.

### XDM

X Display Manager (XDM) включен в пакеты Xfree86 и X.Org. В соответствии со стандартом File System Hierarchy Standard, файлы конфигурации должны располагаться в /etc/X11/xdm. Главный файл конфигурации - /etc/X11/xdm/xdm- config. В этом файле находится информация о других файлах, используемых XDM, о требованиях авторизации, имена скриптов для выполнения различных задач пользователя и некоторая другая информация о конфигурации.

Файл Xservers определяет какой локальный дисплей или дисплеи должен управляться XDM. Как правило, он состоит из одной строки как показано в листинге 13.

### Листинг 13. Пример файла Xservers

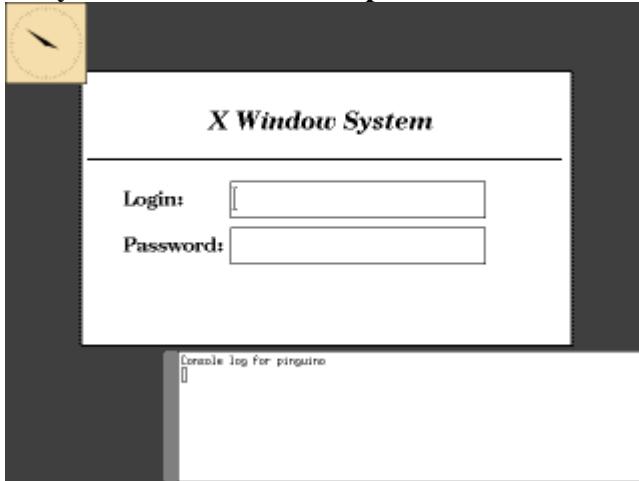
```
:0 local /usr/X11R6/bin/X :0 vt07
```

Листинг 13 показывает, что X должен работать на 7-м виртуальном терминале. Для переключения между виртуальными терминалами большинство систем используют сочетания клавиш от Ctrl-Alt-F1 до Ctrl-Alt-F7, где терминалы от vt01 до vt06 – текстовые терминалы, а vt07 – X-терминал.

Для поддержки работы удалённых X-терминалов вам понадобится файл Xaccess. Этот файл управляет взаимодействием XDM с терминалами, поддерживающими *X Display Manager Control Protocol (XDCMP)*(X протокол управления менеджером экрана). Терминалы, не поддерживающие данный протокол определены в файле Xservers. XDCMP использует хорошо известный UDP порт 177. Из соображений безопасности вы должны ограничить использование XDCMP только доверенной внутренней сетью соответствующей настройкой брандмауэра.

Вы можете настраивать работу XDM обновляя скрипты в /etc/X11/xdm. В частности, скрипт Xsetup (или Xsetup\_0) позволяет вам настроить приветствие. На рисунке 2 показано приветствие XDM с добавленными часами.

**Рисунок 2. Изменённое приветствие XDM**



Исходный код изменённого файла Xsetup\_0 показан в листинге 14.

#### **Листинг 14. Пример файла Xsetup\_0**

```
#!/bin/sh
xclock -geometry 80x80 -bg wheat&
xconsole -geometry 480x130-0-0 -daemon -notify -verbose -fn fixed -exitOnFail
```

Приветствие, показанное на рисунке 2 имеет разрешение 640x480 пикселов и 256 цветов. XDM использует разрешение, установленное по умолчанию в файле XF86Config или xorg.conf. Для изменения разрешения экрана во всей системе, вы можете отредактировать этот файл или воспользоваться утилитами, которые могут иметься в вашей системе. Листинг 15 показывает раздел Screen (Экран) файла XF86Config. Обратите внимание на то, что значение параметра DefaultDepth (глубина цвета по умолчанию) равно 16 и X-сервер попробует запустить экран с первым указанным для этого случая разрешением (в примере 1024x768).

#### **Листинг 15. Настройка разрешения экрана**

```
Section "Screen"
    DefaultDepth 16
    SubSection "Display"
        Depth      15
        Modes     "1280x1024" "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth      16
        Modes     "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth      24
        Modes     "1280x1024" "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth      32
        Modes     "1280x1024" "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth      8
```

```

Modes      "1280x1024" "1024x768" "800x600" "640x480"
EndSubSection
Device     "Device[0]"
Identifier  "Screen[0]"
Monitor    "Monitor[0]"
EndSection

```

Параметр **Depth** (Глубина) указывает количество бит для представления каждого пикселя. Этот параметр также называют числом *бит на пиксель* (*bits per pixel*) или *bitplanes*. Таким образом, использование 8 бит на пиксель (8 бит на каждый цвет) даёт 256 цветов, а 16 бит на пиксель позволяет получить до 65536 цветов. На современных графических картах сейчас используется глубина цвета 24 или 32.

Вы можете уточнить разрешение экрана используя команду **xwininfo** с параметром **-root** для отображения характеристик работающего X-сервера как показано в листинге 16.

#### **Листинг 16. Проверка разрешения экрана**

```

ian@lyrebird:~> xwininfo -display 0:0 -root
xwininfo: Window id: 0x36 (the root window) (has no name)

Absolute upper-left X:  0
Absolute upper-left Y:  0
Relative upper-left X:  0
Relative upper-left Y:  0
Width: 1024
Height: 768
Depth: 16
Visual Class: TrueColor
Border width: 0
Class: InputOutput
Colormap: 0x20 (installed)
Bit Gravity State: NorthWestGravity
Window Gravity State: NorthWestGravity
Backing Store State: NotUseful
Save Under State: no
Map State: IsViewable
Override Redirect State: no
Corners: +0+0  -0+0  -0-0  +0-0
-geometry 1024x768+0+0

```

## **KDM**

KDM это K Desktop Manager (Менеджер рабочего стола K) для K Desktop Environment(KDE) (интегрированная рабочая среда K). KDE версии 3 использует файл конфигурации **kmrc**, в отличии от предыдущих версий, использовавших конфигурационную информацию, основанную на файлах конфигурации **xdm**. Этот файл расположен в **\$KDEDIR/share/config/kdm**, где **\$KDEDIR** может соответствовать **/etc/kde3/kdm/** или чему-то ещё. Например, в SUSE SLES8 он находится в **/etc/opt/kde3/share/config/kdm**.

#### **Листинг 17. Файл конфигурации KDM - kdmrc**

```
[Desktop0]
BackgroundMode=VerticalGradient
```

```

Color1=205,205,205
Color2=129,129,129
MultiWallpaperMode=NoMulti
Wallpaper=UnitedLinux-background.jpeg
WallpaperMode=Scaled

[X-*-Greeter]
GreetString=UnitedLinux 1.0 (%h)
EchoMode=OneStar
HiddenUsers=nobody,
BackgroundCfg=/etc/opt/kde3/share/config/kdm/kdmrc
MinShowUID=500
SessionTypes=kde,gnome,twm,failsafe

[General]
PidFile=/var/run/kdm.pid
Xservers=/etc/opt/kde3/share/config/kdm/Xservers

[Shutdown]
HaltCmd=/sbin/halt
LiloCmd=/sbin/lilo
LiloMap=/boot/map
RebootCmd=/sbin/reboot
UseLilo=false

[X-*-Core]
Reset=/etc/X11/xdm/Xreset
Session=/etc/X11/xdm/Xsession
Setup=/opt/kde3/share/config/kdm/Xsetup
Startup=/etc/X11/xdm/Xstartup
AllowShutdown=Root

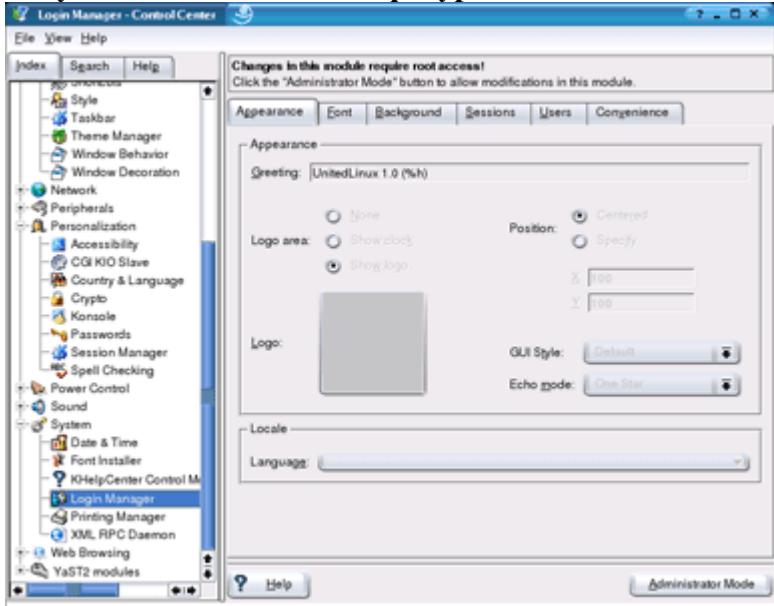
[Xdmcp]
Willing=/etc/X11/xdm/Xwilling
Xaccess=/etc/X11/xdm/Xaccess

```

Многие разделы содержат тот же тип конфигурационной информации что и для XDM, но существуют некоторые отличия. Например, поле SessionTypes (типы сессий) позволяет KDM запускать сессии нескольких различных типов, другие команды позволяют KDM выключать или перезагружать систему.

Вы можете сконфигурировать KDM редактируя файл kdmrc. Вы также можете изменить многие настройки менеджера регистрации (Login Manager), используя центр управления KDE, как показано на рисунке 3.

**Рисунок 3. Изменение конфигурации KDM с использованием kcontrol**



Справочник по KDM (см. [Ресурсы](#)) содержит обширную информацию по настройке KDM.

## GDM

GDM – это GNOME Desktop Manager (Менеджер виртуального рабочего стола GNOME) для GNOME Desktop Environment (интегрированной рабочей среды GNOME). Этот менеджер рабочего стола был написан с нуля, а не основывался на XDM. GDM использует конфигурационный файл gdm.conf, обычно расположенный в каталоге /etc/X11/gdm. В листинге 18 показана часть файла gdm.conf.

### Листинг 18. Часть файла конфигурации GDM - gdm.conf

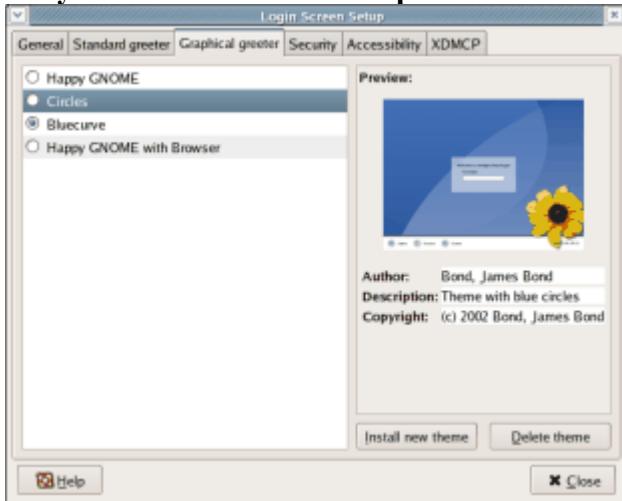
```
# You should probably never change this value unless you have a weird setup
PidFile=/var/run/gdm.pid
# Note that a post login script is run before a PreSession script.
# It is run after the login is successful and before any setup is
# run on behalf of the user
PostLoginScriptDir=/etc/X11/gdm/PostLogin/
PreSessionScriptDir=/etc/X11/gdm/PreSession/
PostSessionScriptDir=/etc/X11/gdm/PostSession/
DisplayInitDir=/etc/X11/gdm/Init

...
# Probably should not touch the below this is the standard setup
ServAuthDir=/var/gdm
# This is our standard startup script. A bit different from a normal
# X session, but it shares a lot of stuff with that. See the provided
# default for more information.
BaseXsession=/etc/X11/xdm/Xsession
# This is a directory where .desktop files describing the sessions live
# It is really a PATH style variable since 2.4.4.2 to allow actual
# interoperability with KDM. Note that <sysconfdir>/dm/Sessions is there
# for backwards compatibility reasons with 2.4.4.x
#SessionDesktopDir=/etc/X11/sessions/:/etc/X11/dm/Sessions/:/usr/share/gdm/Buil\
tInSessions/:/usr/share/xsessions/
# This is the default .desktop session. One of the ones in SessionDesktopDir
DefaultSession=default.desktop
```

Снова вы можете заметить сходство в конфигурационной информации, используемой GDM, KDM, XDM, но файл gdm.conf больше с богатым выбором опций.

Вы можете настроить GDM отредактировав файл gdm.conf. Также большую часть этих настроек можно изменить командой **gdmsetup**. На рисунке 4 приведён выбор различных вариантов приветствия в системе Fedora.

**Рисунок 4. Изменение настроек GDM с использованием gdmsetup**



Справочное руководство по GDM (см. gdmsetup help или [Ресурсы](#)) содержит подробную информацию по настройке GDM.

#### **Раздел 4. Настройка менеджера экрана**

Этот раздел касается темы 1.110.4 экзамена 101. Рейтинг темы - 5

В данном разделе вы узнаете как:

- настраивать виртуальный рабочий стол или менеджер окон;
- настраивать меню оконного менеджера и экранную панель виртуального рабочего стола;
- конфигурировать X-терминал;
- решать вопросы, связанные с использованием различных библиотек X приложениями;
- экспортировать X-экран.

#### **Менеджеры окон**

В предыдущем разделе вы узнали о менеджерах окон и о том, как их устанавливать. Так же из данного пособия вы узнали, что хотя X предоставляет набор средств для создания приложений с графическим интерфейсом, она не определяет сам интерфейс. В этом разделе вы получите дополнительные знания об интерфейсах пользователя и о том, как конфигурировать то, что происходит после запуска X-сессии.

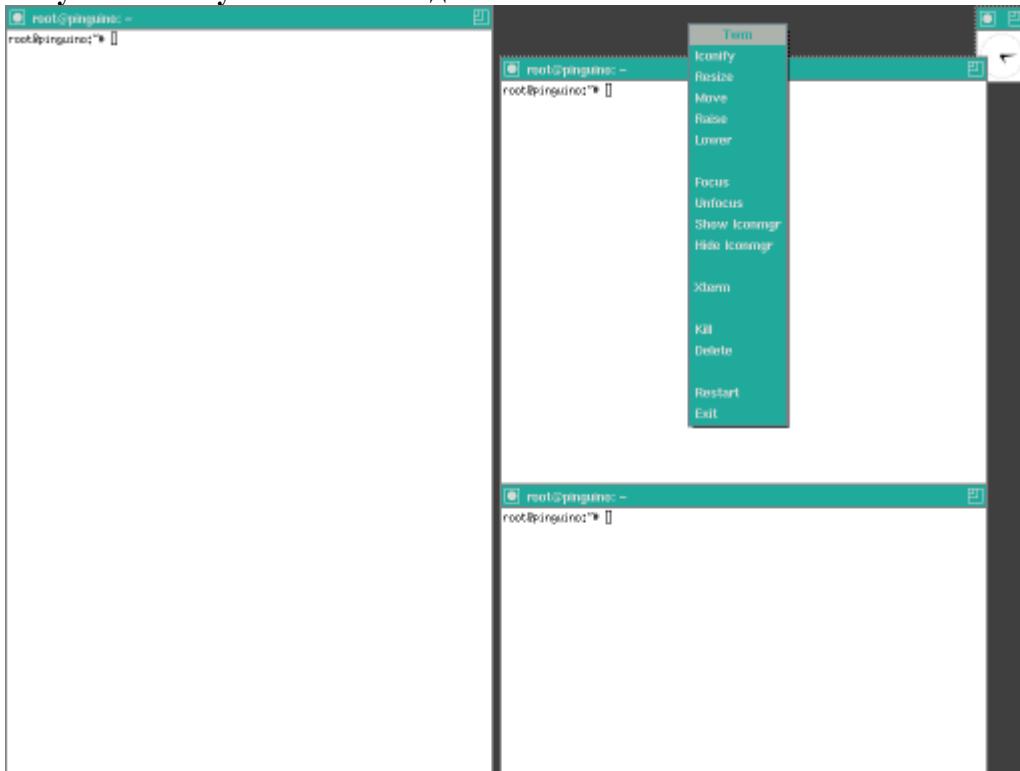
Вы можете подумать, что без задания пользовательского интерфейса фантазия разработчиков приведет к множеству различных стилей окон, которые борются за экранное пространство и все имеют разные сочетания клавиш, действия мыши и стили для кнопок, диалоговых окон и т.д. Для привнесения некоторого порядка в этот хаос были созданы высокоуровневые наборы инструментов. Они породили менеджеры окон типа twm, fvwm, и fvwm2 и в конечном итоге привели к графическим оболочкам KDE и GNOME.

Графические оболочки предоставляют целостную схему поведения пользователя, но также потребляют значительные ресурсы центрального процессора и памяти. До того как

компьютеры стали достаточно мощными для работы с KDE или GNOME менеджеры окон были популярны и многие пользователи до сих пор любят их за лёгкость и быстродействие.

Если вы только что установили X и выполнили команду **startx**, то увидите экран похожий на тот, что изображен на рисунке 5.

**Рисунок 5. Запуск twm командой startx**



Это менеджер окон twm с меню появившимся после нажатия первой кнопки мыши (обычно это правая кнопка для пользователей-правшей) поверх фона. Вы видите три окна виртуальных терминалов и аналоговые часы, но без панелей задач и запуска или другой атрибутики виртуального рабочего стола.

На самом деле команда **startx** представляет внешнюю оболочку для **xinit** запускающей процесс X-сервера и некоторые клиенты. Обычно она располагается в каталоге /usr/X11R6/bin также как **xinit** и многие другие X утилиты. X приложения могут брать настройки из базы данных ресурсов X также как и из командной строки. Таблица 6 резюмирует имена и назначения каждого файла конфигурации, который используется **startx** либо **xinit**. Обратите внимание на то, что некоторые или все эти файлы могут отсутствовать в конкретном системном и домашнем каталоге.

*Таблица 6. Файлы конфигурации для startx и xinit*

<b>Файл</b>	<b>Описание</b>
\$HOME/.xinitrc	Определяемый пользователем скрипт, объединяющий файлы ресурсов и запускающий клиентские приложения.
\$HOME/.xserverrc	Задаваемый пользователем скрипт, позволяющий переопределить конфигурацию X-сервера по умолчанию.
/usr/X11R6/lib/X11/xinit/xinitrc	Системный скрипт, объединяющий файлы ресурсов и запускающий клиентские приложения.
/	Системный скрипт, предоставляющий

```

usr/X11R6/lib/X11/ возможность переопределить конфигурацию
xinit/xserverrc      X-сервера по умолчанию.
$HOME/.Xresource Задаваемый пользователем файл ресурсов для
s                  X-приложений.
$HOME/.Xmodmap Пользовательский файл, определяющий настройки
                  мыши и клавиатуры.
/
usr/X11R6/lib/X11/ Системный файл ресурсов X- приложений.
xinit/.Xresources
/
usr/X11R6/lib/X11/ Системный файл настроек мыши и клавиатуры.
xinit/.Xmodmap
Обратите внимание на то, что системные файлы xinitrc и xserverrc не имеют точки перед
именем, а у всех остальных она есть.

```

Каждое окно на экране и конечно каждый виджет (графический интерфейсный элемент) на экране имеет атрибуты, такие как высота, ширина и размещение (геометрия), цвета или изображения переднего и заднего фона, текст заголовка и его цвет и т.д. Для новых клиентских приложений большинство этих параметров может быть задано в командной строке. Поскольку атрибутов много, то проще использовать параметры по умолчанию. Такие параметры хранятся в *базе данных ресурсов*, которая создаётся из файлов ресурсов командой [xrdb](#).

В листинге 19 приведён файл xinit поставляемый с XFree86 4.5.0

#### **Листинг 19. Пример файла xinit - /usr/X11R6/lib/X11/xinit/xinitrc**

```

#!/bin/sh
# $Xorg: xinitrc.cpp,v 1.3 2000/08/17 19:54:30 cpqbl Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

# merge in defaults and keymaps

if [ -f $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi

if [ -f $userresources ]; then
    xrdb -merge $userresources
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi

# start some nice programs

twm &
xclock -geometry 50x50-1+1 &
xterm -geometry 80x50+494+51 &

```

```
xterm -geometry 80x20+494-0 &
exec xterm -geometry 80x66+0+0 -name login
```

Заметьте, что команда **xrdb** используется для объединения ресурсов, а **xmodmap** для обновления определений мыши и клавиатуры. Наконец, несколько программ запускаются в фоновом режиме, а последняя в обычном, с использованием команды **exec** которая прерывает выполнение текущего скрипта (**xinitrc**) и передаёт управление окну xterm с геометрией 80x66+0+0. Это окно регистрации в системе, его закрытие приведет к остановке X-сервера. Должно быть только одно такое приложение, хотя некоторые пользователи предпочитают, чтобы эту роль выполнял менеджер окон. Все остальные приложения должны запускаться в фоновом режиме, чтобы скрипт мог завершиться.

Первые два значения в определении геометрии задают размер окна. Для часов размер указан в пикселях, а для окон xterm в количестве строк и столбцов. Следующие два значения (если заданы) определяют расположение окна. Если первое значение – «плюс», то позиция отсчитывается относительно левого края экрана, а если «минус» – относительно правого. Аналогично следующие «плюс» и «минус» обозначают соответственно верх и низ экрана.

Допустим, вы хотите увеличить размер часов, изменить их цвет и поместить их правый нижний угол экрана вместо правого верхнего. Если вы хотите сделать это только для одного пользователя скопируйте приведенный выше файл под именем .xinitrc (не забудьте точку) в домашний каталог пользователя и измените определение часов так, как показано в листинге 20. Названия всех цветов находятся в файле **rgb.txt** в дереве каталогов вашей установки X (например /usr/X11R6/lib/X11/rgb.txt).

#### Листинг 20. Изменение параметров запуска xclock в xinitrc

```
xclock -background mistyrose -geometry 100x100-1-1 &
```

Если вы желаете изменить настройки по умолчанию для всей системы, вы должны обновить файлы /usr/X11R6/lib/X11/xinit/Xresources и /usr/X11R6/lib/X11/xinit/Xmodmap, а не пользовательские файлы.

Вот несколько средств, которые помогут вам настроить окна и сочетания клавиш.

#### **xrdb**

Объединяет ресурсы из файла ресурсов в базу данных ресурсов для работающего X-сервера. По умолчанию программа прогоняет файлы через компилятор C++. Если у вас этого компилятора нет, то укажите параметр **-nospp**.

#### **xmodmap**

Задаёт настройки клавиатуры и мыши. Например, вы можете перенастроить мышь для левши или задать привычное для вас поведение для клавиш delete и backspace.

#### **xwininfo**

Выдаёт вам информацию об окне, включая его геометрию.

#### **editres**

Позволяет настраивать ресурсы для окон на вашем экране, просматривать и сохранять изменения в файле, который вы в дальнейшем можете использовать с **xrdb**.

#### **xev**

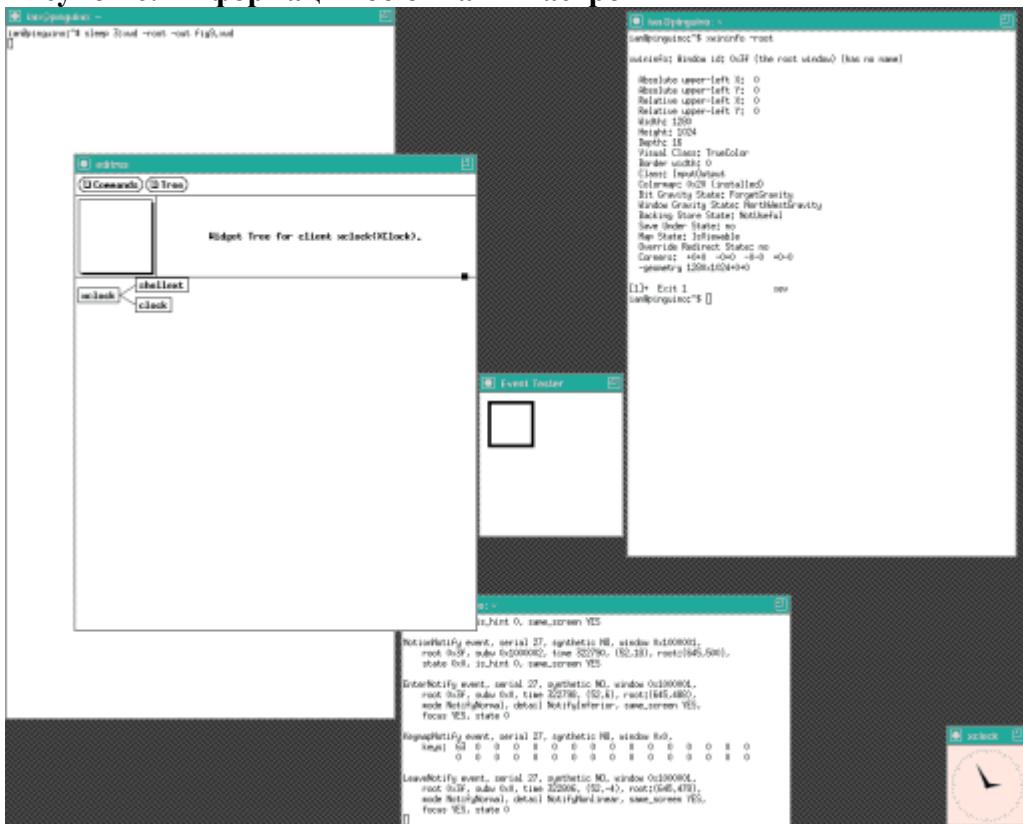
Запускает окно и перехватывает X-события, которые отображаются в окне эмулятора терминала xterm. Используйте эту возможность при настройке клавиатуры или проверки событий мыши.

За дополнительной информацией по каждой из этих команд обращайтесь к интерактивной документации.

На рисунке 6 показан экран, в котором выполняются некоторые из этих команд.

- В левом верхнем окне терминала (окно регистрации) используется `xwd` для сохранения копии экрана в файле.
- В следующем окне выполняется `editres` для модификации ресурсов окна часов.
- В маленьком центральном окне работает `xev` и вывод информации производится в окно терминала. В окне справа показан вывод результатов `xwininfo` для корневого окна(весь экран).
- В правом нижнем углу показаны изменённые часы, окрашенные в розовый цвет.

Рисунок 6. Информация об окнах и настройки



Кроме окон можно настраивать и сам оконный менеджер. Например, меню на рисунке 5 сконфигурировано в файле настройки `twm`. Этот файл по умолчанию расположен в дереве каталогов установки X (`/usr/X11R6/lib/X11/twm/system.twmrc`), а каждый пользователь может иметь собственный `.twmrc` файл. Если у пользователя имеется несколько дисплеев, то могут быть файлы (например `.twmrc.0` или `.twmrc.1`) для каждого номера дисплея. В листинге 21 показана часть файла `system.twmrc`, определяющего меню, показанное на рисунке 5.

## Листинг 21. Настройка меню в `twm`

```
menu "defops"
{
    "Twm"      f.title
    "Iconify"   f.iconify
    "Resize"    f.resize
    "Move"     f.move
    "Raise"     f.raise
    "Lower"    f.lower
```

```

"""
        f.nop
"Focus"      f.focus
"Unfocus"    f.unfocus
>Show Iconmgr" f.showiconmgr
"Hide Iconmgr" f.hideiconmgr
"""
        f.nop
"Xterm"      f.exec "exec xterm &"
"""
        f.nop
"Kill"       f.destroy
"Delete"     f.delete
"""
        f.nop
"Restart"    f.restart
"Exit"       f.quit
}

```

За дополнительной информацией по twm или предпочтаемому вами менеджеру окон обращайтесь к интерактивной справке.

## Графические оболочки

Если вы используете менеджер экрана или графическую оболочку, вы обнаружите, что это также можно настраивать. Конечно, вы уже видели файл Xsetup\_0 для XDM в предыдущем разделе. Настройки графической оболочки, как и только что виденные вами настройки менеджера окон, могут быть общесистемными или пользовательскими.

## Настройка GNOME

GNOME конфигурируется, в основном, посредством XML файлов. Системные настройки по умолчанию находятся в /etc например /etc/gconf, /etc/gnome, и /etc/gnome-vfs2..0 вместе с другими каталогами для конкретных GNOME приложений. Пользовательские настройки обычно располагаются в подкаталогах домашнего каталога пользователя, имена которых начинаются с .g. В листинге 22 показаны некоторые из возможных мест расположения конфигурационной информации.

### Листинг 22. Расположение настроек GNOME

```
[ian@lyrebird ian]$ ls -d /etc/g[cn]*
/etc/gconf /etc/gnome /etc/gnome-vfs-2.0 /etc/gnome-vfs-mime-magic
[ian@lyrebird ian]$ find . -maxdepth 1 -type d -name ".g[nc]*"
./.gnome2
./.gconfd
./.gconf
./.gnome
./.gnome2_private
./.gnome-desktop
./.gnome_private
```

Вместо громоздких страниц интерактивного руководства GNOME имеет online руководство до которого можно добраться командой [gnome-help](#) или выбором пункта меню такого как Desktop > Help. В момент написания этого материала руководство имело три основных раздела: Desktop (Рабочий стол), Applications (Приложения) и Other Documentation (Другая документация). Содержание раздела Desktop показано на рисунке 7.

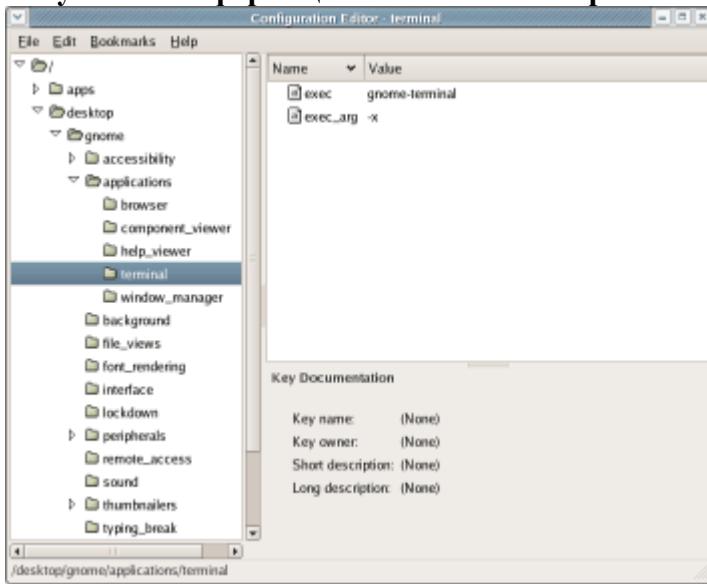
**Рисунок 7. Информация об окнах и настройках**



Информацию по средствам конфигурирования вы найдёте в подразделе System Administration Guide (Руководство по системному администрированию) раздела Desktop, а также в Configuration Editor Manual (руководство по редактору настроек) в теме приложений раздела Desktop.

Вы можете запустить графический редактор конфигурации командой **gconf-editor** или выбором пункта Configuration Editor в меню Applications > System Tools. Конфигурация gnome-терминала показана на рисунке 8.

**Рисунок 8. Информация об окнах и настройках**



Кроме графических средств существует также программа командной строки **gconftool-2** для получения и обновления настроек GNOME. За подробностями обращайтесь к упомянутому выше System Administration Guide.

### Настройка KDE

KDE настраивается посредством простых текстовых файлов, использующих UTF-8 для

представления символов не входящих в ASCII. Как и в GNOME может быть множество различных конфигурационных файлов. Если в дереве конфигурации присутствуют несколько файлов с одинаковыми именами информация из них объединяется.

Конфигурационный файл состоит из одной или нескольких групп имен записей, заключенных в квадратные скобки, за которыми идут пары ключ-значение. Ключи могут содержать пробелы, поскольку разделяются знаком равенства. Файл конфигурации броузера konqueror показан в листинге 23.

### Листинг 23. Файл конфигурации KDE для броузера konqueror

[HTML Settings]

```
[Java/JavaScript Settings]
ECMADomainSettings=localhost::Accept
JavaPath=/usr/lib/java2/jre/bin/java
EnableJava=true
EnableJavaScript=true
```

```
[EmbedSettings]
embed-text=true
embed-audio=false
embed-video=false
```

[Reusing]

```
MaxPreloadCount=1
PreloadOnStartup=true
```

Конфигурационные файлы можно редактировать вручную. Большинство систем включают графические средства редактирования, такие как KConfigEditor или настроенные под конкретный дистрибутив как SUSE Control Center.

## Различные виртуальные терминалы

Обычный эмулятор терминала `xterm`, устанавливаемый с графическими оболочками, обладает хорошими функциональными возможностями, но также потребляет значительные системные ресурсы. Если вы работаете с множеством клиентов X терминала, функционирующего на одном процессоре, вы можете захотеть использовать более легковесный терминал. Два примера `rxvt` и `aterm` (созданный как надстройка над `rxvt`). Это эмуляторы VT102 обычно не устанавливаются по умолчанию, так что вам придется установить их.

## Необходимые библиотеки

К этому моменту вы можете себе представить, что для X-приложений существует множество библиотек и наборов инструментов. Как же убедиться что, вы используете правильные библиотеки? Команда `ldd` выводит список зависимостей для любого приложения. В простейшей форме, она получает название программы и печатает список необходимых библиотек. Заметьте, `ldd` не просматривает автоматически переменную окружения PATH, так что обычно вам придётся кроме имени программы задавать относительный или абсолютный путь (за исключение случая, когда программа находится в текущем каталоге). В листинге 24 показаны зависимости для трёх эмуляторов терминала, обсуждавшихся ранее. Количество зависимостей для каждого, даёт вам общее представление об их системных требованиях.

## Листинг 24. Библиотечные зависимости для xterm, aterm, и rxvt

```
root@pinguino:~# ldd `which xterm`
    linux-gate.so.1 => (0xfffffe000)
    libXft.so.2 => /usr/X11R6/lib/libXft.so.2 (0xb7fab000)
    libfontconfig.so.1 => /usr/X11R6/lib/libfontconfig.so.1 (0xb7f88000)
    libfreetype.so.6 => /usr/X11R6/lib/libfreetype.so.6 (0xb7f22000)
    libexpat.so.0 => /usr/X11R6/lib/libexpat.so.0 (0xb7f06000)
    libXrender.so.1 => /usr/X11R6/lib/libXrender.so.1 (0xb7eff000)
    libXaw.so.7 => /usr/X11R6/lib/libXaw.so.7 (0xb7ead000)
    libXmu.so.6 => /usr/X11R6/lib/libXmu.so.6 (0xb7e99000)
    libXt.so.6 => /usr/X11R6/lib/libXt.so.6 (0xb7e4f000)
    libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0xb7e46000)
    libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0xb7e30000)
    libXpm.so.4 => /usr/X11R6/lib/libXpm.so.4 (0xb7e22000)
    libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0xb7e15000)
    libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0xb7d56000)
    libncurses.so.5 => /lib/libncurses.so.5 (0xb7d15000)
    libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7be6000)
    libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb7be3000)
    /lib/ld-linux.so.2 (0xb7fc3000)
root@pinguino:~# ldd `which aterm`
    linux-gate.so.1 => (0xfffffe000)
    libXpm.so.4 => /usr/X11R6/lib/libXpm.so.4 (0xb7f81000)
    libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0xb7ec1000)
    libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0xb7eb9000)
    libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0xb7ea3000)
    libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7d75000)
    libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb7d72000)
    /lib/ld-linux.so.2 (0x800000000)
root@pinguino:~# ldd `which rxvt`
    linux-gate.so.1 => (0xfffffe000)
    libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0xb7eb0000)
    libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7d81000)
    libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb7d7e000)
    /lib/ld-linux.so.2 (0x800000000)
```

## Экспорт дисплея

Х дисплей определяется по имени в форме *имя\_хоста:номер\_дисплея.номер\_экрана*. Для Linux, работающей на рабочей станции типа ПК обычно имеется только единственный дисплей с одним экраном. В этом случае *имя\_экрана* может быть опущено (обычно так и делается) и имя дисплея принимает вид :0.0. Переменная окружения DISPLAY обычно хранит имя экрана, поэтому его можно посмотреть командой `echo $DISPLAY`. В зависимости от вашей системы эта переменная может быть, а может не быть установлена при переключении пользователя командой `su -`. В таком случае вам необходимо задать и экспортировать переменную DISPLAY как показано в листинге 25. В этом листинге вы видите попытку запуска приложения xclock после переключения пользователя на root. Попытка заканчивается неудачей, поскольку переменная DISPLAY не задана. Даже если эта переменная установлена, вы не можете использовать экран без авторизации.

## Листинг 25. Попытка запуска xclock

```
ian@lyrebird:~> whoami
ian
ian@lyrebird:~> echo $DISPLAY
:0.0
```

```
ian@lyrebird:~> su -
Password:
lyrebird:~ # echo $DISPLAY

lyrebird:~ # xclock
Error: Can't open display:
lyrebird:~ # export DISPLAY=:0.0
lyrebird:~ # echo $DISPLAY
:0.0
lyrebird:~ # xclock
Xlib: connection to ":0.0" refused by server
Xlib: No protocol specified

Error: Can't open display: :0.0
lyrebird:~ # export XAUTHORITY=~ian/.Xauthority
lyrebird:~ # xclock
lyrebird:~ # ls -l ~ian/.Xauthority
-rw----- 1 ian users 206 Feb 18 16:20 /home/ian/.Xauthority
```

Давайте посмотрим на происходящее здесь. Пользователь `ian` зарегистрирован в системе и переменная `DISPLAY` установлена в `:0.0` как ожидалось. Когда `ian` переключился на `root`, переменная `DISPLAY` не была установлена и попытка запуска `xclock` не увенчалась успехом, поскольку приложение не знало о том, какой дисплей необходимо использовать.

Пользователь `root` устанавливает переменную `DISPLAY` и экспортирует её, чтобы она была доступна для других оболочек, которые могут быть запущены из этого окна терминала.

Заметьте, что для установки и экспорта переменной окружения не указывается символ `$` как при отображении или других вариантах её использования. Обратите внимание так же на то, что выполнение команды `su` без знака минус приведёт к установке значения переменной `DISPLAY` таким, каким оно было для пользователя `ian`. Тем не менее, даже при заданном значении переменной `DISPLAY` запуск `xclock` не удаётся.

Причина второй неудачи кроется в клиент/серверной природе `X`. Хотя пользователь `root` работает в окне одного (и единственного!) дисплея системы, дисплей фактически находится в распоряжении пользователя, который изначально зарегистрировался в системе, в нашем случае это `ian`. Давайте обратимся к процедуре авторизации `X`.

## Методы авторизации

Для локального дисплея авторизация в Linux обычно основана на **MIT-MAGIC-COOKIE-1**, которая обычно обновляется при перезапуске `X`-сервера. Пользователь может извлечь «магическое» cookie из файла `.Xauthority` в его домашнем каталоге (используя команду `xauth extract`) и передать это значение другому пользователя для включения в файл `.Xauthority` текущего пользователя командой `xauth merge`. Альтернативным способом предоставления пользователям полномочий на доступ к локальной системе является команда `xhost +local:`.

## XAUTHORITY

Ещё одной альтернативой является задание переменной окружения `XAUTHORITY` на файл, содержащий необходимое значение **MIT-MAGIC-COOKIE-1**. При переключении на `root` это просто сделать, поскольку `root` имеет возможность читать файлы других пользователей, что и было сделано нами в листинге 25. Таким образом, после установки и экспорта переменной `XAUTHORITY` на `~ian/.Xauthority`, `root` может открывать графические окна на рабочем столе. Ранее мы отмечали отличие систем Red Hat. Команда `su` в Red Hat немного отличается от SUSE систем, в которых установка экрана производится автоматически.

А что делать, если переключаемся на пользователя отличного от root? Из листинга 25 вы заметили, что файл .Xauthority пользователя ian позволяет доступ только для него. Даже члены группы не могут его прочесть. Это то, что надо, если только вы не хотите, чтобы кто-то выполнил приложение на вашем экране и лишил вас возможности что-либо делать. Так что если вы извлекли MIT-MAGIC-COOKIE-1 из .Xauthority, вам необходимо отыскать безопасный способ его передачи другому пользователю. Другой подход состоит в использовании команды **xhost** для предоставления полномочий любому пользователю конкретного хоста.

### Команда xhost

В связи с трудностями безопасной передачи значения MIT-MAGIC-COOKIE-1 другому пользователю в Linux системах с одним пользователем использовать **xhost** проще несмотря на то, что в общем случае **xauth** предпочтительнее. Не забывайте сетевое наследие X Window System и не предоставьте случайно больше полномочий, чем хотите и не откройте таким образом вашу систему для случайных сетевых пользователей.

Для предоставления полномочий работы с приложениями на дисплее (:0.0) пользователь ian может использовать команду **xhost**. Для этого откройте окно эмулятора терминала и введите команду:

```
xhost +local:
```

Обратите внимание на завершающее двоеточие. Оно позволяет другим пользователям системы подключаться к X-серверу и открывать окна. Для однопользовательской системы это означает, что вы можете переключаться на любого не root пользователя и запустить теперь **xclock** или другое X приложение.

Вы можете использовать **xhost** для авторизации удаленных хостов. Обычно это плохая идея, за исключением ограниченных сетей. Если вы используете эту возможность вам потребуется открыть соответствующие порты в межсетевом экране (если вы его используете).

Другой возможностью использования X приложений с другой системы является подключение через secure shell (**ssh**) (безопасная оболочка). Если по умолчанию ваш ssh клиент не поддерживает X, то может потребоваться параметр **-X** для команды **ssh**. На сервере ssh также должна быть включена поддержка X. В общем случае это более безопасный способ удаленной работы с X чем использование **xhost**.

Для получения подробной информации об использовании команд **xauth** и **xhost**, можете воспользоваться командами **info xauth**, **man xauth**, **info xhost** или **man xhost**. Если вас заинтересовал вопрос о безопасности X соединений начните с изучения документации по **Xsecure**.

## Ресурсы

### Научиться

- [LPI exam prep tutorial series](#). Знакомство с полным перечнем учебников на сайте developerWorks для изучения основ Linux и подготовки к сертификационным экзаменам по системному администрированию.
- [LPIC Program](#). В программе LPIC вы найдете перечень заданий, примеры вопросов, подробное описание программы для трёх уровней сертификации по системному администрированию Профессионального Института Linux (Linux Professional Institute).
- В статье "[Basic tasks for new Linux developers](#)" (Основные задачи для новых Linux разработчиков) (developerWorks, февраль 2006), вы узнаете как открывать окно

виртуального терминала или получать приглашение системы и многое другое.

- Получите [документацию по XFree86 версии 4.5.0](#) и другим версиям на сайте XFree86.
- [Документация по X Window System Version 11 выпуск 6.9 и 7.0](#) на сайте X.Org.
- Документ [\*X.Org Modular Tree Developer's Guide\*](#) поможет вам собрать X.Org из исходных файлов.
- [Учебное руководство по Xft](#) описывает механизм шрифтов Xft, представленный в XFree86 4.0.2.
- [kde.org](#) Домашняя страница K Desktop Environment и [документация по KDE](#), включая [Справочник по kdm](#) и [KDE для системных администраторов](#).
- [The GNOME Foundation](#) – домашняя страница GNOME Desktop Environment и [Руководство по Gnome Display Manager](#).
- [Linux Documentation Project](#) содержит массу полезных документов, особенно материалов типа HOWTO.
- Книга [\*The Concise Guide to Xfree86 for Linux\*](#) (Que, 1999) подробнее освещает вопросы установки, конфигурирования и использования X.
- Книги [\*LPI Linux Certification in a Nutshell\*](#) (O'Reilly, 2001) и [\*LPIC 1 Exam Cram 2: Linux Professional Institute Certification Exams 101 and 102 \(Exam Cram 2\)\*](#) (Que, 2004) предназначены для тех, кто предпочитает бумажный вариант.
- Узнайте больше о [руководствах для Linux разработчиков](#) на [developerWorks Linux zone](#).
- Следите за событиями с помощью [developerWorks technical events and Webcasts](#).

## Получить продукты и технологии

- [Загрузите XFree86](#) с сайта The XFree86 Project, Inc.
- [Загрузите X.Org](#) с сайта The X.Org Foundation.
- [Загрузите пробное программное обеспечение IBM](#) прямо с developerWorks.

# Экзамен LPI 102: Ядро

*Администрирование Linux для начинающих (LPIC-1) тема 105*

[Ян Шилдс](#), Старший программист, EMC

**Описание:** В этом учебном пособии Ян Шилдс начинает готовить вас к сдаче Экзамена 102 Linux Professional Institute® Администрирование Linux для начинающих (Junior Level Administration, LPIC-1). В этом первом в [серии из девяти пособий](#) Ян знакомит вас с ядром Linux. Прочтя это пособие, вы узнаете, как собрать и установить ядро Linux и его модули, а также получить информацию о ядре и его модулях.

[Больше статей из этой серии](#)

**Дата:** 21.03.2006

**Уровень сложности:** средний

## Прежде чем начать

Узнайте, чему может научить вас это учебное пособие и как извлечь из него максимум.

Об этой серии учебных пособий

[Linux Professional Institute](#) (LPI) осуществляет сертификацию системных администраторов Linux по двум уровням: *для начинающих* (также называемый "уровень сертификации 1") и *средний уровень* (также называемый "уровень сертификации 2"). Для достижения уровня сертификации 1 вы должны сдать экзамены 101 и 102; для достижения уровня сертификации 2 -- экзамены 201 и 202.

developerWorks предоставляет учебные пособия, которые помогут вам в подготовке к каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем, и для каждой темы на developerWorks существует соответствующее пособие для самостоятельного изучения. Экзамен LPI 102 содержит девять тем, которым соответствуют учебные пособия от developerWorks:

*Таблица 1. Экзамен LPI 102: Учебные пособия и темы*

Тема экзамена LPI 102	Учебное пособие от developerWorks	Краткое содержание пособия
Тема 105	Материалы к экзамену LPI 102: Ядро	(Это пособие). Установка и сопровождение ядра Linux и его модулей. Подробнее см. <a href="#">цели</a> ниже.
Тема 106	Материалы к экзамену LPI 102: Загрузка, инициализация системы, завершение работы, уровни выполнения	Скоро ожидается.
Тема 107	Материалы к экзамену LPI 102: Печать	Скоро ожидается.
Тема 108	Материалы к экзамену LPI 102: Документация	Скоро ожидается.
Тема 109	Материалы к экзамену LPI 102: Командные оболочки, написание	Скоро ожидается.

скриптов, программирование и компиляция

Тема 111	Материалы к экзамену LPI 102: Задачи администрирования	Скоро ожидается.
Тема 112	Материалы к экзамену LPI 102: Основы работы в сети	Скоро ожидается.
Тема 113	Материалы к экзамену LPI 102: Сетевые сервисы	Скоро ожидается.
Тема 114	Материалы к экзамену LPI 102: Безопасность	Скоро ожидается.

Чтобы сдать экзамены 101 и 102 (и достичь уровня сертификации 1), вы должны уметь:

- Работать в командной строке Linux
- Выполнять простые операции сопровождения: управлять учетными записями пользователей, производить резервирование и восстановление, а также завершать работу и перезагружать компьютер
- Устанавливать и настраивать рабочую станцию (в том числе систему X Window), подсоединяясь к локальной сети (LAN) или подключать отдельно стоящий компьютер к сети интернет посредством модема

Для продолжения подготовки к сертификации уровня 1 см. [Учебные пособия от developerWorks для экзаменов LPI 101 и LPI 102](#), а также [полный набор учебных пособий LPI от developerWorks](#).

Linux Professional Institute не одобряет использование при подготовке к экзаменам любых учебных материалов или технологий, разработанных третьими лицами. За разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

### Об этом учебном пособии

Добро пожаловать в учебное пособие "Ядро", первое из девяти пособий, разработанных для подготовки к экзамену LPI 102. Из этого пособия вы узнаете, как собрать и установить ядро Linux и его модули, а также получить информацию о ядре и его модулях.

Это учебное пособие организовано в соответствии с рабочей программой LPI по этой теме. Грубо говоря, экзамены с большим количеством вопросов имеют больший рейтинг.

*Таблица 2. Ядро: Цели экзамена, описанные в этом учебном пособии*

Цель экзамена LPI	Рейтинг	Описание цели
1.105.1 <a href="#">Управление ядром и его модулями и получение информации о них в ходе работы</a>	Рейтинг 4	Научиться управлять и получать информацию о ядре и загружаемых модулях ядра.
1.105.2 <a href="#">Переконфигурация, сборка и инсталляция собственного ядра и его модулей.</a>	Рейтинг 3	Имея исходники, научиться конфигурировать, собирать и устанавливать ядро и его загружаемые модули.

### Необходимые условия

Чтобы извлечь максимум из этого учебного пособия, вы должны иметь базовые знания о Linux и рабочую версию системы Linux, где вы сможете упражняться в выполнении команд,

приведенных в этом пособии.

Это пособие предполагает, что вы знакомы с предыдущими учебными пособиями серии LPI. Возможно, вы захотите сначала ознакомиться с [учебными пособиями для экзамена 101](#). В частности, вам необходимо изучить пособие [Экзамен LPI 101: Аппаратные средства и архитектура](#).

Формат вывода программы может быть различным в зависимости от ее версии, так что результат вашей работы может выглядеть не совсем так, как это представлено в листингах и на рисунках этого пособия.

## Экзамен LPI 102: Ядро

*Администрирование Linux для начинающих (LPIC-1) тема 105*

[Ян Шилдс](#), Старший программист, EMC

**Описание:** В этом учебном пособии Ян Шилдс начинает готовить вас к сдаче Экзамена 102 Linux Professional Institute® Администрирование Linux для начинающих (Junior Level Administration, LPIC-1). В этом первом в [серии из девяти пособий](#) Ян знакомит вас с ядром Linux. Прочтя это пособие, вы узнаете, как собрать и установить ядро Linux и его модули, а также получить информацию о ядре и его модулях.

[Больше статей из этой серии](#)

**Дата:** 21.03.2006

**Уровень сложности:** средний

### Управление ядром в ходе работы системы

В этом разделе мы рассматриваем материал по теме 1.105.1 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 4.

Из этого раздела вы узнаете, как:

- С помощью утилит командной строки получать информацию о ядре и его модулях в ходе работы системы
- Вручную загружать и удалять модули ядра
- Определять, когда модули могут быть удалены
- Конфигурировать систему для загрузки модулей, имена которых отличаются от названий соответствующих файлов

Формально Linux -- это ядро вашей системы. Ядро обеспечивает инфраструктуру для работы приложений и использования различных аппаратных средств. Это код низкого уровня, который взаимодействует с интерфейсами аппаратных средств, планирует и распределяет память и т.д. Выбор в пользу систем GNU/Linux часто обусловлен тем, что многие инструменты, создаваемые для большинства дистрибутивов, благодаря проекту GNU Фонда свободного программного обеспечения (Free Software Foundation) являются доступными. Однако, вы будете часто видеть вместо "GNU/Linux" только "Linux".

#### **uname**

Команда **uname** выдает информацию о вашей системе и ее ядре. В Листинге 1 показаны различные опции команды **uname** и получаемая с их помощью информация. Описание опций дается в Таблице 3.

## Листинг 1. Команда `uname`

```
ian@pinguino:~$ uname
Linux
ian@pinguino:~$ uname -s
Linux
ian@pinguino:~$ uname -n
pinguino
ian@pinguino:~$ uname -r
2.6.12-10-386
ian@pinguino:~$ uname -v
#1 Mon Jan 16 17:18:08 UTC 2006
ian@pinguino:~$ uname -m
i686
ian@pinguino:~$ uname -o
GNU/Linux
ian@pinguino:~$ uname -a
Linux pinguino 2.6.12-10-386 #1
Mon Jan 16 17:18:08 UTC 2006 i686 GNU/Linux
```

Таблица 3. Опции команды `uname`

Опция	Описание
-s	Показать имя ядра. Эта информация выдается по умолчанию, если ни одна опция не указана.
-n	Показать имя узла сети или имя хоста.
-r	Показать номер выпуска ядра. Эта опция часто используется с командами управления модулями.
-v	Показать версию ядра.
-m	Показать имя аппаратной платформы (CPU).
-o	Показать имя операционной системы.
-a	Показать всю возможную информацию.

Листинг 1 получен на операционной системе Ubuntu, запущенной на процессоре Intel®. Команда `uname` доступна в большинстве систем UNIX® и UNIX-подобных систем, таких как Linux. Выдаваемая ею информация может быть различной в разных дистрибутивах и версиях Linux, а также на разных типах компьютеров. В Листинге 2 показаны выводы команды `uname` на системе Fedora Core 4, запущенной на машине AMD Athlon 64, и, для сравнения, на Apple PowerBook.

## Листинг 2. Использование команды `uname` на других системах

```
Linux attic4 2.6.14-1.1656_FC4 #1
Thu Jan 5 22:13:55 EST 2006 x86_64
x86_64 x86_64 GNU/Linux filesystem
```

```
Darwin Ian-Shields-Computer.local 7.9.0 Darwin Kernel Version 7.9.0:
Wed Mar 30 20:11:17 PST 2005;
root:xnu/xnu-517.12.7.obj~1/RELEASE_PPC
Power Macintosh powerpc
```

## Модули ядра

Ядро управляет системой на низшем уровне, включая оборудование и интерфейсы. При большом разнообразии аппаратных средств и различных файловых систем ядро, способное поддерживать все их, будет слишком большим. К счастью, *модули ядра* позволяют при необходимости загрузить обеспечивающее поддержку программное обеспечение, такое как драйверы для аппаратных средств или файловые системы. Это позволяет запускать систему с небольшим ядром и затем подгружать модули по мере необходимости. Часто эта подгрузка происходит автоматически, например, при подключении устройств USB.

В оставшейся части этого раздела мы рассмотрим, как и с помощью каких команд производится конфигурирование модулей ядра.

Команды для выполнения задач загрузки и удаления модулей ядра требуют полномочий суперпользователя root. Команды, выдающие информацию о модулях, обычно могут быть выполнены обычным пользователем. Однако, в случае, если они расположены в каталоге /sbin, они будут недоступны для обычного пользователя, так как этот каталог не включается в путь поиска PATH. Таким образом, если вы не root, вам, вероятно, надо будет использовать полное наименование пути.

### lsmod

Воспользуйтесь командой `lsmod`, чтобы узнать, какие модули загружены на вашей системе в настоящее время (см. Листинг 3). Вероятно, у вас вывод этой команды будет другим, хотя некоторые записи должны совпадать.

### Листинг 3. Просмотр модулей ядра с помощью команды lsmod

```
[ian@attic4 ~]$ /sbin/lsmod
Module           Size  Used by
nvnet            74148  0
nvidia           4092336 12
forcedeth        24129  0
md5              4161   1
ipv6             268737 12
parport_pc       29189  1
lp                13129  0
parport          40969  2 parport_pc,lp
autofs4          29637  1
sunrpc           168453 1
ipt_REJECT       5825   1
ipt_state         1985   3
ip_conntrack     42009  1 ipt_state
iptable_filter   3137   1
ip_tables         19521  3 ipt_REJECT,
                  ipt_state,iptable_filter
dm_mod           58613  0
video            16069  0
button           4161   0
battery          9541   0
ac                4933   0
ohci_hcd         26977  0
ehci_hcd         41165  0
i2c_nforce2      7105   0
i2c_core          21825  1 i2c_nforce2
shpchp           94661  0
snd_intel8x0     34945  1
snd_ac97_codec   76217  1 snd_intel8x0
snd_seq_dummy    3781   0
snd_seq_oss       37569  0
```

```

snd_seq_midi_event      9409  1 snd_seq_oss
snd_seq                  62801  5 snd_seq_dummy,
snd_seq_oss,snd_seq_midi_event
snd_seq_device           9037  3 snd_seq_dummy,
snd_seq_oss,snd_seq
snd_pcm_oss              51569  0
snd_mixer_oss             18113  1 snd_pcm_oss
snd_pcm                  100553  3 snd_intel8x0,snd_ac97_codec,
snd_pcm_oss
snd_timer                33733  2 snd_seq,snd_pcm
snd                      57669  11 snd_intel8x0,snd_ac97_codec,
snd_seq_oss,snd_seq,
snd_seq_device,snd_pcm_oss,snd_mixer_oss,snd_pcm,snd_timer
soundcore                 11169  1 snd
snd_page_alloc            9925   2 snd_intel8x0,snd_pcm
floppy                   65397  0
ext3                      132681  3
jbd                      86233  1 ext3
sata_nv                  9541   0
libata                     47301  1 sata_nv
sd_mod                    20545  0
scsi_mod                  147977  2 libata, sd_mod
[ian@attic4 ~]$
```

Вы можете видеть, что в системе загружено множество модулей. Большинство из них поставляются вместе с ядром. Однако некоторые, такие как nvnet, nvidia и sata\_nv от корпорации NVIDIA, содержат проприетарный код и не являются частью стандартного ядра. Таким образом, модульный подход позволяет подключать проприетарный код к ядру с открытым кодом. При условии, что лицензия производителя разрешает это, в дистрибутив Linux могут быть внесены проприетарные модули, что позволит не тратить силы на получение их непосредственно от производителя и даст гарантию, что вы сможете воспользоваться ими.

В Листинге 3 вы также можете видеть, что соответствующими модулями осуществляется поддержка таких устройств как видео, SATA, SCSI, дискеты и звуковые карты, а также сетевые устройства, например, IPV6, поддержка файловых систем, такой как ext3, и Remote Procedure Call (RPC) компании Sun.

Помимо имени модуля, команда **lsmod** показывает также размер и число пользователей модуля. Если модуль используется более чем одним пользователем, будет представлен список этих пользователей. Так, например модуль **soundcore** используется модулем **snd**, который в свою очередь используется некоторыми другими звуковыми модулями.

### **modinfo**

Команда **modinfo** выдает информацию об одном или нескольких модулях. Как показано в Листинге 4, эта информация содержит полный путь до файла, имя автора, лицензию, другие параметры, которые можно передать модулю, версию, зависимости и другую информацию.

### **Листинг 4. Основная информация о модуле**

```
[ian@attic4 ~]$ /sbin/modinfo floppy
filename:      /lib/modules/2.6.12-1.
1456_FC4/kernel/drivers/block/floppy.ko
author:        Alain L. Knaff
license:       GPL
alias:         block-major-2-*
```

```

vermagic:      2.6.12-1.1456_FC4 686 REGPARM 4KSTACKS gcc-4.0
depends:
srcversion:   2633BC999A0747D8D215F1F
parm:          FLOPPY_DMA:int
parm:          FLOPPY_IRQ:int
parm:          floppy:charp
[ian@attic4 ~]$ /sbin/modinfo sata_nv
filename:     /lib/modules/2.6.12-1.1456_FC4
/kernel/drivers/scsi/sata_nv.ko
author:        NVIDIA
description:   low-level driver for NVIDIA nForce SATA controller
license:       GPL
version:      0.6
vermagic:      2.6.12-1.1456_FC4 686 REGPARM 4KSTACKS gcc-4.0
depends:
libata
alias:         pci:v000010DEd0000008Esv*sd*bc*sc*i*
alias:         pci:v000010DEd000000E3sv*sd*bc*sc*i*
alias:         pci:v000010DEd000000EEsv*sd*bc*sc*i*
alias:         pci:v000010DEd00000054sv*sd*bc*sc*i*
alias:         pci:v000010DEd00000055sv*sd*bc*sc*i*
alias:         pci:v000010DEd00000036sv*sd*bc*sc*i*
alias:         pci:v000010DEd0000003Esv*sd*bc*sc*i*
alias:         pci:v000010DEd*sv*sd*bc01sc01i*
srcversion:    3094AD48C1B869BCC301E9F

```

В Листинге 4 обратите внимание на строки, содержащие имена файлов модулей. Эти имена оканчиваются суффиксом .ko, что является характерным отличием модулей для ядра 2.6 от других объектных файлов и от модулей для ядра 2.4 и более ранних версий, в которых и для них и для других объектных файлов использовался один и тот же суффикс .o.

Вы можете также заметить, что путь включает в себя версию ядра. Например, частью пути /lib/modules/2.6.12-1.1456\_FC4/kernel/drivers/block/floppy.ko является 2.6.12-1.1456\_FC4. Это запись совпадает с выводом команды `uname -r`. Модули являются характерными для данного ядра, и эта структура каталогов отражает их взаимозависимость.

В системах с ядром версии 2.6 с помощью команды `modinfo` можно ограничить количество запросов определенной информации о модуле. Чтобы извлечь информацию одного типа, например, `parm`, `description`, `license`, `filename` или `alias`, воспользуйтесь опцией `-F`. Если вам необходимо получить информацию различных типов, используйте команду несколько раз с различными опциями. В ядрах версии 2.4 информацию о параметрах можно получить с помощью опции `-p`. Текущая версия команды `modinfo` поддерживает также параметры предыдущих версий. В Листинге 5 показаны несколько примеров.

## Листинг 5. Определенная информация о модуле

```

[ian@attic4 ~]$ /sbin/modinfo -F parm snd
cards_limit:Count of auto-loadable soundcards.
major:Major # for sound driver.
[ian@attic4 ~]$ /sbin/modinfo -F license nvidia floppy
NVIDIA
GPL
[ian@attic4 ~]$ /sbin/modinfo -p snd
major:Major # for sound driver.
cards_limit:Count of auto-loadable soundcards.

```

## Используйте ваши навыки работы в Linux

Вы можете использовать некоторые приемы, описанные в учебном пособии "[Экзамен LPI 101 \(тема 103\): Команды GNU и UNIX](#)" для получения информации, например, о количестве параметров, которые можно передать модулю. В Листинге 6 показан пример.

### Листинг 6. Количество параметров модуля

```
[ian@attic4 ~]$ for n in `sbin/lsmod | tail +2 | cut -d " " -f1`;  
> do echo "$n $(/sbin/modinfo -p $n |wc -l )"  
| grep -v " 0$"; done  
nvnet 12  
forcedeth 1  
parport_pc 5  
dm_mod 1  
ohci_hcd 2  
ehci_hcd 2  
shpchp 3  
snd_intel8x0 7  
snd_ac97_codec 1  
snd_seq_dummy 2  
snd_seq_oss 2  
snd_seq 7  
snd_pcm_oss 3  
snd_pcm 2  
snd_timer 1  
snd 2  
snd_page_alloc 1  
scsi_mod 6
```

### rmmod

Если "use count" модуля равен 0, вы можете без опасений удалить его. Например, вы можете сделать это при подготовке к загрузке обновленной версии. Это позволит не перезагружать компьютер только из-за того, что необходимо обновить поддержку какого-то отдельного устройства. Для удаления введите команду `rmmod` имя модуля, как показано в Листинге 7.

### Листинг 7. Удаление модуля из работающей системы

```
[root@attic4 ~]# rmmod floppy
```

Чтобы узнать о других опциях команды `rmmod`, обратитесь к ее странице man.

### insmod и modprobe

Удаленный вами модуль может понадобиться вновь. Чтобы загрузить модуль, введите команду `insmod`, полный путь для модуля, который необходимо загрузить, и все необходимые опции. При использовании этой команды вы, вероятно, захотите использовать режим подстановки вывода команды (command substitution), чтобы сгенерировать имя файла. В Листинге 8 показаны два способа загрузки модуля.

### Листинг 8. Загрузка модуля с использованием команды insmod

```
[root@attic4 ~]# insmod /lib/modules/`uname -r`/kernel/drivers/block/floppy.ko  
[root@attic4 ~]# rmmod floppy
```

```
[root@attic4 ~]# insmod $(modinfo -F filename floppy)
```

При использовании второго варианта нет необходимости помнить, в каком подкаталоге (в данном случае `drivers/block`) расположен модуль. Но существует и более удобный способ загрузки модуля. Команда `modprobe` предоставляет высокоуровневый интерфейс, оперирующий именами модулей, а не именами путей файлов. Она также управляет загрузкой дополнительных модулей, от которых зависит загружаемый модуль, и позволяет не только загружать, но и удалять модули.

В Листинге 9 показано, как использовать команду `modprobe` для удаления модуля `vfat` вместе с используемым им модулем `fat`. Затем показано, как будет вести себя система, если модуль будет загружен вновь, и в завершение показан результат загрузки модуля. Обратите внимание на опцию `-v`, которая позволяет получать подробный вывод. Если не использовать эту опцию, команда `modprobe` (а также команда `insmod`) выведет только сообщения об ошибках модуля. Между каждым действием используется команда `lsmod`, пропущенная через `grep`, чтобы проверить, загружены или нет модули `vfat` и `fat`.

### Листинг 9. Загрузка модуля с помощью команды modprobe

```
[root@lyrebird root]# modprobe -r vfat
vfat: Device or resource busy
[root@lyrebird root]# lsmod | grep fat
vfat                  13132   1
fat                   38744   1  [vfat]
[root@lyrebird root]# umount /windows/D
[root@lyrebird root]# modprobe -r vfat
[root@lyrebird root]# modprobe -v --show vfat
/sbin/insmod /lib/modules/2.4.21-37.0.1.EL/kernel/fs/fat/fat.o
/sbin/insmod /lib/modules/2.4.21-37.0.1.EL/kernel/fs/vfat/vfat.o
[root@lyrebird root]# lsmod | grep fat
[root@lyrebird root]# modprobe -v vfat
/sbin/insmod /lib/modules/2.4.21-37.0.1.EL/kernel/fs/fat/fat.o
Using /lib/modules/2.4.21-37.0.1.EL/kernel/fs/fat/fat.o
Symbol version prefix ''
/sbin/insmod /lib/modules/2.4.21-37.0.1.EL/kernel/fs/vfat/vfat.o
Using /lib/modules/2.4.21-37.0.1.EL/kernel/fs/vfat/vfat.o
[root@lyrebird root]# lsmod | grep fat
vfat                  13132   0  (unused)
fat                   38744   0  [vfat]
```

### depmod

Как вы только что видели, в случае, если один модуль зависит от другого, при помощи команды `modprobe` можно управлять автоматической загрузкой нескольких модулей. Зависимости прописаны в файле `modules.dep`, хранящемся в подкаталоге `/lib/modules` соответствующего ядра (имя ядра определяется при помощи команды `uname -r`). Этот файл вместе с несколькими тар-файлами генерируется при помощи команды `depmod`. Опция `-a` (от `all`) теперь необязательна.

Команда `depmod` просматривает модули в подкаталоге `/lib/modules` текущего ядра и обновляет информацию о зависимостях. В Листинге 10 показан пример выполнения команды `depmod` и полученных в результате файлов.

## Листинг 10. Использование команды depmod для создания файла modules.dep

```
[root@lyrebird root]# date
Thu Mar 16 10:41:05 EST 2006
[root@lyrebird root]# depmod
[root@lyrebird root]# cd /lib/modules/`uname -r`
[root@lyrebird 2.4.21-37.0.1.EF]# ls -l mod*
-rw-rw-r-- 1 root root 54194 Mar 16 10:41 modules.dep
-rw-rw-r-- 1 root root 31 Mar 16 10:41 modules.generic_string
-rw-rw-r-- 1 root root 73 Mar 16 10:41 modules.ieee1394map
-rw-rw-r-- 1 root root 1614 Mar 16 10:41 modules.isapnpmap
-rw-rw-r-- 1 root root 29 Mar 16 10:41 modules.parportmap
-rw-rw-r-- 1 root root 65171 Mar 16 10:41 modules.pcimap
-rw-rw-r-- 1 root root 24 Mar 16 10:41 modules.pnpbiosmap
-rw-rw-r-- 1 root root 122953 Mar 16 10:41 modules.usbmap
[root@lyrebird 2.4.21-37.0.1.EF]# cd -
/root
```

Вы можете изменить поведение команд **modprobe** и **depmod**, скорректировав файл **/etc/modules.conf**. Обычно это делается для создания псевдонимов (alias) для имен модулей и определения команд, которые должны быть запущены после загрузки модуля или перед его отключением. Однако, могут быть произведены и другие настройки. В Листинге 11 показан пример файла **/etc/modules.conf**. Более подробную информацию можно найти в странице man для **modules.conf**.

## Листинг 11. Пример файла /etc/modules

```
[root@lyrebird root]# cat /etc/modules.conf
alias eth0 e100
alias usb-controller usb-uhci
alias usb-controller1 ehci-hcd
alias sound-slot-0 i810_audio
post-install sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -L >/
dev/null 2>&1 || :
pre-remove sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -S >/
dev/null 2>&1 || :
```

Вы должны также знать, что некоторые системы используют другой конфигурационный файл, **modprobe.conf**, и есть системы, которые хранят информацию о конфигурации модулей в каталоге **/etc/modules.d**. В некоторых системах вы можете встретить файлы с названием **/etc/modules**. Эти файлы содержат имена модулей ядра, которые должны быть загружены в ходе загрузки системы.

## Модули USB

Когда вы подключаете к работающей системе Linux устройство USB, ядро должно определить, какие модули необходимо загрузить для управления устройством. Обычно это делается при помощи hot-plug скрипта, использующего команду **usbmodules** для поиска соответствующего модуля. Вы можете также запустить команду **usbmodules** (от имени root). В Листинге 12 показан пример.

## Листинг 12. Модули USB

```
root@pinguino:~# lsusb
```

```
Bus 005 Device 004: ID 1058:0401 Western Digital Technologies, Inc.
Bus 005 Device 003: ID 054c:0220 Sony Corp.
Bus 005 Device 001: ID 0000:0000
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 003: ID 04b3:310b IBM Corp. Red Wheel Mouse
Bus 001 Device 001: ID 0000:0000
root@pinguino:~# usbmodules --device /proc/bus/usb/005/003
usb-storage
root@pinguino:~# usbmodules --device /proc/bus/usb/001/003
usbmouse
usbhid
```

В следующем разделе показано, как собрать и настроить собственное ядро.

## Настройка и сборка ядра и его модулей

В этом разделе мы рассматриваем материал по теме 1.105.2 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 3.

Из этого раздела вы узнаете, как:

- Изменить текущую конфигурацию ядра
- Собрать новое ядро и соответствующие модули
- Установить новое ядро и любые модули
- Гарантировать, что загрузчик системы сможет определить местоположение нового ядра и связанных с ним файлов

Как вы узнали из предыдущего раздела, [Управление ядром в ходе работы системы](#), ядро обеспечивает поддержку низкого уровня для аппаратных средств и файловых систем. Современные образы ядра обычно обладают только необходимой минимальной функциональностью, но при необходимости могут быть настроены для поддержки дополнительных функций с помощью *модулей ядра* (*kernel modules*). Дополнительная поддержка включается только при необходимости, например, при подсоединении устройства или в других случаях.

Код модулей становится неотъемлемой частью ядра, динамично расширяя его функциональность. Если возможности загруженных модулей ядра в данное время не используется, ядро может самостоятельно отключить их от основной части и выгрузить из памяти при помощи процесса, известного как *autocleaning*.

Ядро, загруженное с диска без модулей, как отдельный бинарный файл, должно было бы обеспечивать всю необходимую функциональность. И при необходимости добавить системе функциональности вам пришлось бы каждый раз полностью пересобирать ядро.

Однако, вы не можете поместить в модули *все*. Для монтирования файловой системы необходим, как минимум, образ ядра. Но, как вы могли узнать из учебного пособия "[Экзамен LP1 101 \(тема 102\): Инсталляция Linux и управление пакетами](#)", загрузчик может загрузить *стартовый RAM-диск* (*initial RAM disk* или *initrd*), который может содержать необходимые для монтирования файловой системы модули. Тем не менее, образ ядра должен как минимум включать поддержку файловой системы RAM, используемой в initial RAM disk, иначе ваша система не сможет быть загружена.

В ходе загрузки системы происходит монтирование файловой системы и затем запускаются другие процессы инициализации. Через некоторое время система загрузится полностью и будет готова к использованию. Однако, ядро будет находиться в состоянии готовности

выполнять пользовательские процессы и распределять системные ресурсы между использующими их задачами.

Модульные ядра хорошо работают в современных системах с большим количеством RAM и дискового пространства. Однако, вы можете подключить новые аппаратные средства, например, видеокарту или запоминающее устройство, которые не поддерживаются ядром, входящим в дистрибутив. Некоторые устройства содержат проприетарный код, который, как было сказано выше *загрязняют* чистое ядро Linux, поэтому некоторые дистрибутивы не включают их, даже если лицензия производителя позволяет это. В таких случаях необходимо по крайней мере собрать новые модули, а возможно, и пересобрать ядро.

Linux может использоваться как во встроенных системах, таких как мобильные телефоны, сетевых устройствах, таких как маршрутизаторы, и компьютерных приставках, так и в компьютерах большинства традиционных конфигураций. Некоторые из этих устройств используют ядра, настроенные на поддержку только тех функций, которые обеспечивают поддержку системы. Например, бездисковой системе, предназначеннной для использования в качестве firewall, вероятно, не потребуется поддержка других типов файловых систем, кроме файловой системы read-only, с помощью которой она загружается, но может потребоваться поддержка продвинутых сетевых устройств, не входящих в стандартное ядро. К тому же потребуется специально настроенное ядро.

### Пакеты с исходными кодами

Исходные коды ядра Linux можно найти в Linux Kernel Archives (см. [Ресурсы](#)). Для начала воспользуйтесь пакетами для ядра, входящего в ваш дистрибутив Linux, так как производитель мог добавить в него индивидуальные патчи. Если вы уже умеете находить и извлекать исходные коды ядра, просмотрите учебное пособие "[Экзамен LPI 101 \(тема 102\): Инсталляция Linux управление пакетами](#)". Прежде чем вносить изменения, сделайте резервную копию системы, чтобы можно было восстановить ее, если что-то пойдет не так

Если вы загружаете исходники из публичных архивов ядра, вы получаете сжатые файлы, которые необходимо декомпрессировать с помощью команды `gzip` или `bzip2`, в случае, если вы получили исходники в формате .gz, или команды `bzip2` для исходников в формате .bz2. Каталог `pub/linux/kernel/` на сервере содержит каталоги с исходниками для ядра версий 2.4, 2.5 и 2.6. На время написания этого пособия последняя версия ядра 2.6 содержалась в `linux-2.6.15.tar.bz2`.

В каталоге с исходниками ядра также содержится соответствующий файл `ChangeLog-2.6.15.6`, в котором описаны все изменения для текущей версии, и `patch-2.6.15.bz2`, позволяющий внести изменения в исходники предыдущей версии, приведя их к версии 2.6.15. Также вы заметите файлы, содержащие подписи, которые могут быть использованы для того, чтобы убедиться, что загруженные файлы случайно или умышленно не были повреждены.

Распаковка сжатых исходников обычно производится в каталог `/usr/src`, при этом создается подкаталог для версии ядра, например `linux-2.6.15`, содержащий дерево файлов, необходимых для сборки ядра. Если у вас уже есть такой каталог, вы можете создать его резервную копию или переименовать его, прежде чем приступить к распаковыванию новых исходников ядра. Это позволит вам в случае необходимости вернуться назад, а также даст гарантию того, что в дереве исходников ядра не окажется случайных файлов. Вам понадобится приблизительно 40 Мбайт для архива tar и приблизительно 350 Мбайт для разворачивания исходных кодов.

В настоящее время некоторые производители, в особенности Red Hat, распространяют файлы заголовков (headers) ядра и исходники, необходимые для сборки модулей, в виде пакета `kernel development`. Документация может не входить в этот пакет. Это делается и этого достаточно для того, чтобы собирать отдельные модули, например модули для проприетарных графических карт, но недостаточно для пересборки ядра. Производитель вашего

дистрибутива должен иметь информацию, о том, как пересобрать ядро, и о том, где получить исходники. Обратитесь к соответствующей документации, например, посмотрите замечания к выпуску.

Предположим, вы получили через FTP или HTTP с сервера download.fedora.redhat.com из каталога pub/fedora/linux/core/updates/4/SRPMS/ исходный пакет kernel-2.6.15-1.1833\_FC4.src.rpm и поместили его в каталог /root. Номер версии ядра, используемый здесь для примера, может отличаться от номера версии в вашей системе. Если используется дистрибутив Fedora Core, необходимо установить исходный пакет RPM, затем перейти в каталог /usr/src/redhat/SPECS и в заключение пересобрать исходный RPM, для того чтобы создать дерево исходников ядра Linux, как показано в Листинге 13.

### Листинг 13. Создание дерева исходников ядра для Fedora Core

```
[root@attic4 ~]# uname -r
2.6.15-1.1833_FC4
[root@attic4 ~]# rpm -Uvh kernel-2.6.15-1.1833_FC4.src.rpm
 1:kernel
 ##### [100%]
[root@attic4 ~]# cd /usr/src/redhat/SPECS
[root@attic4 SPECS]# rpmbuild -bp --target $(arch) kernel-2.6.spec
Building target platforms: x86_64
Building for target x86_64
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.23188
+ umask 022
+ cd /usr/src/redhat/BUILD
+ LANG=C
+ export LANG
+ unset DISPLAY
+ '[' '!' -d kernel-2.6.15/vanilla ']'
+ cd /usr/src/redhat/BUILD
+ rm -rf kernel-2.6.15
+ /bin/mkdir -p kernel-2.6.15
+ cd kernel-2.6.15
+ /usr/bin/bzip2 -dc /usr/src/redhat/
SOURCES/linux-2.6.15.tar.bz2
+ tar -xf -
...
+ echo '# x86_64'
+ cat .config
+ perl -p -i -e 's/^SUBLEVEL.*$/SUBLEVEL = 15/' Makefile
+ perl -p -i -e 's/^EXTRAVERSION.*$/EXTRAVERSION = -prep/' Makefile
+ find . -name '*.orig' -o -name '*~' -exec rm -f '{}' ';' '
+ exit 0
```

Исходники ядра Linux для Fedora теперь расположены в /usr/src/redhat/BUILD/kernel-2.6.15/linux-2.6.15. В соответствии с принятыми соглашениями дерево /linux-2.6.15 часто перемещают в каталог /usr/src и делают символьную ссылку в /usr/src/linux, как показано в Листинге 14. Это не обязательно, но когда исходники ядра находятся в /usr/src/linux, проще ориентироваться по ссылкам.

### Листинг 14. Перемещение дерева исходников в /usr/src

```
[root@attic4 SPECS]# mv ..../BUILD/kernel-2.6.15/linux-2.6.15 /usr/src
[root@attic4 SPECS]# cd /usr/src
[root@attic4 src]# ln -s linux-2.6.15 linux
```

```
[root@attic4 src]# ls -ld lin*
lrwxrwxrwx 1 root root 12 Mar 20 18:23 linux -> linux-2.6.15
drwxr-xr-x 20 root root 4096 Mar 20 18:13 linux-2.6.15
```

Прежде чем попытаться что-нибудь пересобрать, просмотрите файл Changes, расположенный в каталоге Documentation. Он среди прочего содержит список пакетов, необходимых для сборки ядра, с указанием номеров версий. Убедитесь, что эти пакеты установлены.

В Листинге 13 вы могли заметить файлы Makefile и .config. make-файл содержит различные цели сборки для конфигурирования опций ядра, сборки ядра и его модулей, установки модулей и сборки пакетов RPM или deb. Наиболее свежие версии исходников ядра позволяют использовать **make help** для получения краткой справки для каждой цели. В более старых системах было необходимо обращаться к документации или просматривать make-файл. В Листинге 15 показана часть вывода **make help**.

### Листинг 15. Справка по make-файлу для сборки ядра

```
[ian@attic4 linux-2.6.15]$ make help
Cleaning targets:
  clean           - remove most generated files but keep the config
  mrproper        - remove all generated files + config + various backup files

Configuration targets:
  config          - Update current config utilising a line-oriented program
  menuconfig      - Update current config utilising a menu based program
  xconfig         - Update current config utilising a QT based front-end
  gconfig         - Update current config utilising a GTK based front-end
  oldconfig       - Update current config utilising a provided .config as base
  randconfig      - New config with random answer to all options
  defconfig       - New config with default answer to all options
  allmodconfig   - New config selecting modules when possible
  allyesconfig   - New config where all options are accepted with yes
  allnoconfig    - New minimal config

Other generic targets:
  all             - Build all targets marked with [*]
  * vmlinux       - Build the bare kernel
  * modules       - Build all modules
  modules_install - Install all modules
  dir/            - Build all files in dir and below
  dir/file.[ois]  - Build specified target only
  ...
```

## Конфигурация

Файл .config содержит информацию о конфигурации ядра, включая конфигурацию целевой платформы, какие компоненты будут включены, и должны ли компоненты быть включены в ядро или собраны в виде модулей. Создание файла .config является первым шагом на пути сборки или пересборки ядра. Этот файл создается при помощи одной из конфигурационных целей make-файла.

Основные опции конфигурации:

### **config**

Цель **config** использует интерфейс командной строки для получения ответов многие на вопросы, касающиеся создания или обновления файла .config. После появления

конфигурационных целей, использующих меню, интерфейс командной строки используется редко.

#### **menuconfig**

Цель **menuconfig** использует программу с меню-интерфейсом, построенную на базе ncurses, для создания или обновления файла .config. Вы должны только ответить на вопросы для элементов, которые хотите изменить. Этот подход заменил старую цель config. Выполняется в окне терминала удаленно или локально.

#### **xconfig**

Цель **xconfig** использует систему графического меню, основанную на QT front-end, используемом в KDE desktop.

#### **gconfig**

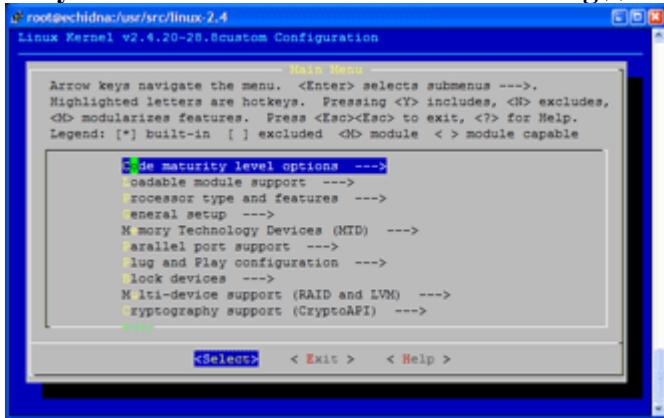
Цель **gconfig** использует систему графического меню, основанную на QT front-end, используемом в GNOME desktop.

#### **oldconfig**

Цель **oldconfig** позволяет создать конфигурацию с использованием существующего файла .config, созданного ранее или взятого из другой системы. Например, если вы устанавливали исходники ядра для Fedora, как описано выше, вы можете скопировать конфигурационный файл для вашей системы из `/lib/modules/$(uname -r)/build/.config` в `/usr/src/linux`. Сделав это, можно использовать одну из целей меню конфигурации, чтобы при необходимости внести изменения.

На Рисунке 1 показан пример того, что вы можете увидеть, запустив `make menuconfig` для ядра 2.4. Используйте **Enter**, чтобы перейти к меню ниже уровнем и **Esc** для возврата. Для каждого элемента можно получить справку. Для этого выберите **< Help >** и нажмите **Enter** или просто введите **h**. Нажмите **Esc** для возврата.

**Рисунок 1. Выполнение make menuconfig для ядра 2.4**



В Таблице 4 показаны различные опции, позволяющие включать компоненты в ядро или создавать специальные модули. Когда опция подсвечена, при помощи клавиши пробела можно перемещаться между возможными вариантами для данного компонента. Чтобы активировать опцию, нажмите **y**, чтобы отключить -- **n**, чтобы создать, если это возможно, модуль, нажмите **m**.

*Таблица 4. Опции menuconfig*

#### **Опция**

#### **Описание**

[\*] Компонент будет включен в ядро.

[ ] Компонент не будет включен в ядро.

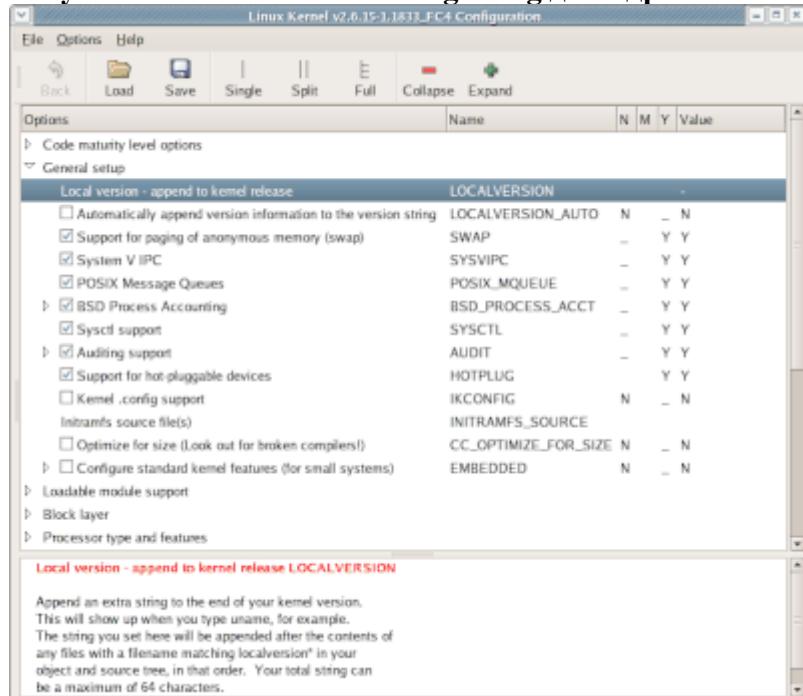
<M> Компонент будет оформлен в виде модуля.

<> Компонент не будет включен в ядро, но

может быть оформлен в виде модуля.

На Рисунке 2 показан пример того, что вы можете увидеть, запустив `make gconfig` для ядра 2.6. Щелчком по стрелкам можно развернуть или свернуть элемент меню. На нижней панели отображается справочная информация.

**Рисунок 2. Выполнение make gconfig для ядра 2.6**



Ниже дается описание главных конфигурационных разделов для ядра 2.6. В ядре 2.4 и более ранних версиях могут присутствовать не все эти разделы. Этот список дает краткий обзор того, где и что расположено.

### **Code maturity level options**

Этот раздел содержит опцию, определяющую, будет ли вам предоставляться возможность выбирать код, имеющий статус экспериментального. Если вы не выберете эту опцию, выбирать можно будет только опции, имеющие стабильный статус. Имейте в виду, что функции, которые вы выбрали, могут работать, а могут и нет в текущей версии ядра, так что у вас есть шанс помочь в его отладке.

### **General setup**

Этот раздел позволяет добавить идентификационную строку к вашему ядру, а также ряд атрибутов, которые не имеют отношения к каким-либо разделам, но тем не менее должны быть описаны.

### **Loadable module support**

Этот раздел содержит опции, определяющие, будет ли ваше ядро поддерживать модули и будут ли они подгружаться и выгружаться автоматически. Опцию "Enable loadable module support" следует включить.

### **Block layer**

Этот раздел управляет поддержкой дисков размером более 2ТБ и позволяет выбирать режимы обслуживания дисковых устройств.

### **Processor type and features**

Этот раздел содержит специфичные для данного типа процессора конфигурационные опции. Здесь вы можете выбрать процессор и семейство процессора, которые будут поддерживаться вашим ядром. Вы можете включать или отключать поддержку ядром различных возможностей, предоставляемых данным процессором. Убедитесь, что вы включили поддержку многопроцессорных систем (symmetric multi-processing support),

если в вашем системе установлено более одного процессора или процессор поддерживает технологию hyperthreading. Кроме того, для получения большей производительности графической подсистемы в системах с AGP или PCI видеокартами следует включить поддержку MTRR.

### **Power management options**

В этом разделе помещены опции, касающиеся управления питанием. Особенно они важны для ноутбуков. Кроме контроля состояния питания, вы сможете найти там средства для контроля и мониторинга таких параметров как температура или состояние охлаждающего вентилятора.

### **Bus options (PCI etc.)**

Этот раздел содержит опции для компьютерных шин, поддерживаемых вашей системой, таких как PCI, PCI Express и PC Card. Здесь вы можете включить поддержку файловой системы /proc/pci, которой можно пользоваться вместе с обычно используемой командой `lspci`.

### **Executable file formats / Emulations**

Этот раздел содержит опции, касающиеся поддержки различных форматов бинарных файлов. Следует включить поддержку "ELF binary". Кроме того, можно включить поддержку DOS binaries для запуска их под DOSEMU, также как и других поддерживаемых соответствующими wrapper'ами бинарных файлов, таких как Java™, Python, Emacs-Lisp и т.д. Наконец, для 64-битных систем, поддерживающих 32-битную эмуляцию, вы, возможно, захотите включить поддержку 32-битных приложений.

### **Networking**

Секция, касающаяся настроек сети, довольно велика. Здесь вы можете включить базовую поддержку сокетов, сетей TCP/IP, фильтрацию, маршрутизацию и bridging сетевых пакетов, а также поддержку различных протоколов, таких как IPV6, IPX, Appletalk и X.25. Кроме того, вы можете включить поддержку wireless, infrared и amateur radio.

### **Device drivers**

Этот раздел также очень велик. Здесь вы можете включить поддержку большого числа аппаратных устройств, включая IDE/ATAPI или SCSI диски, или flash-диски. Включите DMA для ваших IDE устройств; иначе они будут работать в более медленной PIO-моде. Если вы хотите иметь поддержку multiple devices, таких как RAID или LVM, соответствующие опции также надо включить. Здесь вы также можете включить поддержку параллельного порта для работы с принтером через этот интерфейс. Здесь происходит конфигурирование широкого набора поддерживаемых сетевых устройств для различных сетевых протоколов, которые мы конфигурировали ранее. Кроме того, здесь вы найдете опции поддержки устройств аудио- и видео-захвата, устройств USB и IEEE 1394 (Firewire), а также различного рода устройств аппаратного мониторинга. В разделе управления символьными устройствами (Character Devices) вы, возможно, захотите включить поддержку печати через параллельный порт и поддержку direct rendering.

### **Firmware drivers**

Этот раздел содержит несколько опций, относящихся к установке и обновлению BIOS, таких как использование функций Dell System Management на некоторых системах производства компании Dell.

### **File systems**

Этот раздел предназначен для конфигурирования файловых систем, поддержку которых вы хотите иметь в вашем ядре, скомпилированных в виде модулей или нет. Также вы сможете найти здесь файловые системы для съемных дисковых устройств (дискеты, CD и DVD устройства), а также сетевых файловых систем, таких как NFS, SMB или CIFS. Поддержка различных типов разделов и национальных кодировок Native Language Support также располагаются в этом разделе.

## Instrumentation support

Этот раздел позволяет вам включать экспериментальную поддержку профайлинга для профилирования работы вашей системы.

## Kernel hacking

Этот раздел позволяет включать режим отладки ядра и выбирать, какие дополнительные функции будут включены.

## Security options

Этот раздел предназначен для конфигурирования опций защиты, а также включения и конфигурирования SELinux (Security Enhanced Linux).

## Cryptographic options

В этом разделе можно сконфигурировать поддержку различных алгоритмов шифрования, таких как MD4, DES и SHA256.

## Library routines

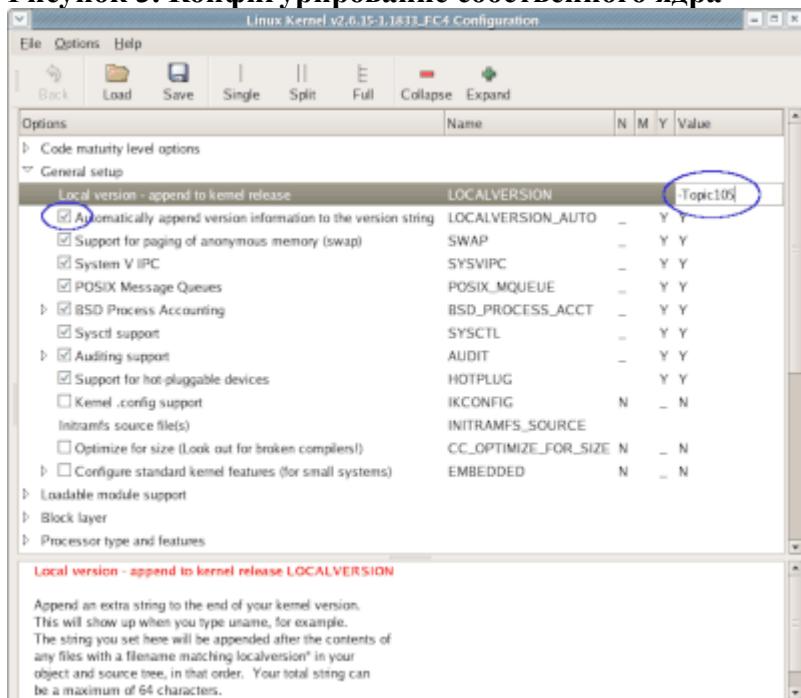
Здесь вы можете указать ряд алгоритмов вычисления контрольных сумм (CRC), которые будут включены в ядро или собраны как модули

## Сборка

Теперь, когда вы ознакомились с главными аспектами конфигурации ядра, вы готовы к его сборке. Если вы не знаете, каково состояние дерева для сборки, прежде чем приступить к конфигурированию нового ядра выполните `make clean`. Для более полной очистки выполните `make mrproper`. При этом будет удален файл `.config`, а также некоторые другие файлы, используемые в процессе сборки. Если вы сделаете это и затем захотите восстановить файл `.config` из резервной копии, вам надо будет выполнить `make oldconfig`, прежде чем приступить к конфигурированию.

В ходе эксперимента вы должны дать новому ядру специальное название, которое позволит вам легко его идентифицировать. Чтобы сделать это, установите значение Local version и активируйте опцию Automatically append version information to the version string в соответствующей строке раздела General setup, как показано на Рисунке 3.

Рисунок 3. Конфигурирование собственного ядра



В примерах, приведенных в оставшейся части этого учебного пособия, при сборке ядра

вносятся только те два изменения, которые показаны на Рисунке 3.

В принципе, для сборки ядра не требуются полномочия суперпользователя root, несмотря на то, что для установки нового ядра эти полномочия необходимы. Однако, если вы используете пакеты, установленные из дистрибутива, вам, вероятно, понадобятся привилегии суперпользователя root для доступа к необходимым файлам и каталогам. Выкачав исходники ядра из Linux kernel archives и распаковав их в своем домашнем каталоге или скопировав дерево сборки ядра и скорректировав права доступа, вы сможете поупражняться в сборке ядра, используя свою учетную запись пользователя.

Чтобы начать сборку ядра 2.6, выполните **make**.

Чтобы начать сборку ядра 2.4, выполните эти три команды:

```
make dep  
make bzImage  
make modules
```

Первая создает файлы необходимых зависимостей. Вторая собирает ядро. И последняя собирает модули.

Выполнение **make** на моей системе AMD Athlon 3500+ с целью полностью собрать ядро с нуля заняло около получаса. На более медленных системах выполнение этой задачи может занять до двух часов, так что сделайте перерыв или займитесь чем-нибудь другим. После запуска процесса сборки будут появляться сообщения подобные приведенным в Листинге 16.

### Листинг 16. Выполнение make

```
[root@attic4 linux]# make  
CHK      include/linux/version.h  
HOSTCC   scripts/basic/fixdep  
HOSTCC   scripts/basic/split-include  
HOSTCC   scripts/basic/docproc  
SPLIT    include/linux/autoconf.h -> include/config/*  
CC       arch/x86_64/kernel/asm-offsets.s  
GEN      include/asm-x86_64/asm-offsets.h  
.  
.  
LD [M]   sound/usb/snd-usb-lib.ko  
CC      sound/usb/usx2y/snd-usb-usx2y.mod.o  
LD [M]   sound/usb/usx2y/snd-usb-usx2y.ko
```

### Инсталляция

После того как у вас будет полностью собранное ядро, вы должны выполнить еще два действия. Сначала вам необходимо выполнить **make modules\_install** для установки модулей ядра в новый подкаталог /lib/modules.

Если вам необходимы проприетарные модули для видеокарты или сетевой карты, как это понадобилось мне для графической карты nVidia и для чипсета материнской платы nForce 4, сейчас подходящее время для того, чтобы собрать эти модули, используя предоставленные производителем средства.

И в заключение, вам необходимо выполнить **make install** для установки нового ядра и стартового RAM-диска (initial RAM disk) в каталог /boot и обновления конфигурации загрузчика. Листинг 17 иллюстрирует эти действия.

## Листинг 17. Инсталляция ядра и модулей

```
[root@attic4 linux]# make modules_install
  INSTALL arch/x86_64/crypto/aes-x86_64.ko
  INSTALL arch/x86_64/kernel/cpufreq/acpi-cpufreq.ko
  INSTALL arch/x86_64/kernel/microcode.ko
  INSTALL arch/x86_64oprofileoprofile.ko
  INSTALL crypto/aes.ko
  INSTALL crypto/anubis.ko
  INSTALL crypto/arc4.ko
...
[root@attic4 linux]# ls -lrt /lib/modules | tail -n 3
drwxr-xr-x 5 root root 4096 Mar 4 14:48 2.6.15-1.1831_FC4
drwxr-xr-x 5 root root 4096 Mar 20 18:52 2.6.15-1.1833_FC4
drwxr-xr-x 3 root root 4096 Mar 20 21:38 2.6.15-prep-Topic105
[root@attic4 linux]# sh /root/NFORCE-Linux-x86_64-1.0-0310-pkg1.run -a \
> -n -K -k 2.6.15-prep-Topic105
Verifying archive integrity...OK
Uncompressing NVIDIA nForce drivers for
Linux-x86_64 1.0-0310.....
[root@attic4 linux]# sh /root/NVIDIA-Linux-x86_64-1.0-8178-pkg2.run -a \
> -n -K -k 2.6.15-prep-Topic105
Verifying archive integrity... OK
Uncompressing NVIDIA Accelerated Graphics Driver for Linux-x86_64 1.0-8178..
[root@attic4 linux]# make install
  CHK      include/linux/version.h
  CHK      include/linux/compile.h
  CHK      usr/initramfs_list
Kernel: arch/x86_64/boot/bzImage is ready (#2)
sh /usr/src/linux-2.6.15/arch/x86_64/boot/install.sh 2.6.15-prep-Topic105
arch/x86_64/boot/bzImage System.map "/boot"
[root@attic4 linux]# ls -lrt /boot | tail -n 6
-rw-r--r-- 1 root root 1743149 Mar 20 21:45 vmlinuz-2.6.15-prep-Topic105
lrwxrwxrwx 1 root root      28 Mar 20 21:45 vmlinuz -> vmlinuz-2.6.15-prep-Topic105
-rw-r--r-- 1 root root  980796 Mar 20 21:45 System.map-2.6.15-prep-Topic105
lrwxrwxrwx 1 root root      31 Mar 20 21:45 System.map
-> System.map-2.6.15-prep-Topic105
-rw-r--r-- 1 root root 1318741 Mar 20 21:45 initrd-2.6.15-prep-Topic105.img
drwxr-xr-x 2 root root    4096 Mar 20 21:45 grub
```

## Стартовый RAM-диск

Обратите внимание, что в процессе сборки автоматически создается необходимый стартовый RAM-диск (initial RAM disk или initrd). Если у вас возникнет необходимость создать его вручную, это можно сделать при помощи команды [mkinitrd](#). Подробную информацию вы найдете в странице man для этой команды.

## Загрузчики

Если процесс идет без сбоев, в ходе выполнения `make install` будет также обновлена конфигурация загрузчика. В Листинге 18 приведены несколько строк из моего загрузчика.

## Листинг 18. Обновленный конфигурационный файл GRUB

```
default=1
timeout=10
splashimage=(hd0,5)/boot/grub/splash.xpm.gz
password --md5 $1$y.uQRs1W$Sqs30hDB3GtE957PoiDW0.
title Fedora Core (2.6.15-prep-Topic105)
```

```
root (hd0,11)
kernel /boot/vmlinuz-2.6.15-prep-Topic105 ro root=LABEL=FC4-64 rhgb quiet
initrd /boot/initrd-2.6.15-prep-Topic105.img
title Fedora Core -x86-64 (2.6.15-1.1833_FC4)
```

Запись для нового ядра расположена самой первой, но в качестве записи по умолчанию осталась та, которая была выбрана ранее. Если вы используете не GRUB, а LILO, выполните команду **grubby**, которая используется в скрипте, осуществляющем сборку ядра, для обновления конфигурации загрузчика LILO. Если по какой-то причине обновление конфигурации прошло некорректно обратитесь к учебному пособию "[Экзамен LPI 101 \(тема 102\): Инсталляция Linux и управление пакетами](#)". Там вы можете найти подробные инструкции по настройке загрузчика.

В заключение одно замечание. Вас могло удивить, что в примере конфигурации была использована запись **-Topic105**, а в результате все созданные файлы содержали вместо нее **-prep-Topic105**. Это одна из применяемых в дистрибутивах Fedora мер по обеспечению безопасности, которая предотвращает неумышленное разрушение имеющегося ядра. Контроль осуществляется путем установки переменной **EXTRAVERSION** в начальной части главного make-файла, как показано в Листинге 19. Если вы хотите отменить эту возможность, отредактируйте make-файл.

### Листинг 19. Начало главного make-файла

```
[root@attic4 linux]# head -n 6 Makefile
VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 15
EXTRAVERSION = -prep
NAME=Sliding Snow Leopard
```

### Перезагрузка

Если все прошло успешно, теперь вы сможете загрузить вашу новую систему. Вы должны выбрать конфигурационную запись для нового ядра, поскольку она не является (еще) записью по умолчанию. Если вас все устроит, можно будет сделать ее записью по умолчанию. После перезагрузки, чтобы проверить, какое загружено ядро, выполните команду **uname**, как показано в Листинге 20.

### Листинг 20. Проверка новой системы

```
[ian@attic4 ~]$ uname -rv
2.6.15-prep-Topic105 #2 Mon Mar 20 21:13:20 EST 2006
```

## Ресурсы

### Научиться

- [Оригинал этого учебного пособия](#) на developerWorks.
- Обзор всей [серии учебных пособий для экзамена LPI](#) на developerWorks для изучения основ операционной системы Linux и подготовки к сертификации по системному администрированию.
- В [Программе LPIC](#) вы найдете список заданий, типовые вопросы и подробные

программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.

- Из "[Basic tasks for new Linux developers](#)" (developerWorks, март 2005) вы узнаете о том, как открыть окно терминала или оболочку командной строки и о многом другом.
- [Linux Documentation Project](#) содержит ряд полезных документов, главным образом HOWTO.
- [The Linux Kernel Archives](#) - основной ресурс для ядра Linux. Прежде чем приступить к загрузке, выберите ближайшее зеркало.
- [kernelnewbies project](#) содержит массу информации для тех, кто плохо знаком с ядрами и процедурой их сборки.
- [Kernel Rebuild Guide](#) покажет, как сконфигурировать, собрать и установить новое ядро.
- [Linux Kernel Module Programming Guide](#) от [Linuxtopia](#) -- онлайн книга о модулях ядра Linux.
- [LPI Linux Certification in a Nutshell](#) (O'Reilly, 2001) и [LPIC 1 Exam Cram 2: Linux Professional Institute Certification Exams 101 and 102 \(Exam Cram 2\)](#) (Que, 2004) рекомендуются для тех, кто предпочитает печатные издания.
- Найдите другие [учебные пособия для Linux-разработчиков](#) на [developerWorks Linux zone](#).
- Ознакомьтесь с текущими [техническими событиями developerWorks](#) и [Webcasts](#).

#### **Получить продукты и технологии**

- Загрузите [IBM trial software](#) прямо с developerWorks.

# Учебник для экзамена LPI 102, Тема 106: Загрузка, инициализация, остановка и уровни выполнения

*Администрирование Linux для начинающих (LPIC-1), тема 106*

[Ian Shields](#), Старший программист, EMC

**Описание:** В этом учебнике Ян Шилдс (Ian Shields) продолжает готовить вас к сдаче Экзамена LPI 102 Профессионального Института Linux (Linux Professional Institute®): Администрирование Linux для начинающих (LPIC-1). В этом втором [в серии из девяти учебников](#) Ян знакомит вас с запуском и остановкой Linux®. К концу этого учебника вы будете знать, как провести загрузку системы, задать параметры ядра и остановить или перезагрузить систему.

[Больше статей из этой серии](#)

**Дата:** 04.04.2006

**Уровень сложности:** средний

## Перед тем как начать

Узнайте, чему вас могут научить эти учебники, и как извлечь из них максимум.

## Об этой серии

[Профессиональный Институт Linux](#) (LPI) сертифицирует системных администраторов Linux по двум уровням: *уровень для начинающих* (также называемый "уровень сертификации 1") и *средний уровень* (также называемый "уровень сертификации 2"). Для получения уровня сертификации 1, вы должны сдать экзамены 101 и 102; для получения уровня сертификации 2, вы должны сдать экзамены 201 и 202.

developerWorks предлагает учебники, чтобы помочь вам подготовиться к каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем, а для каждой темы существует соответствующий учебник на developerWorks. Экзамену LPI 102 соответствуют следующие девять тем и учебников от developerWorks:

Таблица 1. Экзамен LPI 102: Учебники и темы

Тема экзамена <b>LPI 102</b>	Учебник developerWorks	Краткое содержание учебника
Тема 105	<a href="#">Учебник для экзамена LPI 102: Ядро</a>	Узнайте, как устанавливать и управлять ядрами Linux и модулями ядра.
Тема 106	Учебник для экзамена LPI 102: Загрузка, инициализация, остановка и уровни выполнения	(Этот учебник). Узнайте, как загружать систему, задавать параметры ядра и выключать или останавливать систему. Смотрите подробную <a href="#">программу</a> ниже.
Тема 107	Учебник для экзамена LPI 102:	Скоро ожидается.

## Печать

Тема 108	Учебник для экзамена LPI 102: Документация	Скоро ожидается.
Topic 109	Учебник для экзамена LPI 102: Оболочки, скрипты, программирование и компиляция	Скоро ожидается.
Тема 111	Учебник для экзамена LPI 102: Задачи администрирования	Скоро ожидается.
Тема 112	Учебник для экзамена LPI 102: Основы сетей	Скоро ожидается.
Тема 113	Учебник для экзамена LPI 102: Сетевые службы	Скоро ожидается.
Тема 114	Учебник для экзамена LPI 102: Безопасность	Скоро ожидается.

Чтобы сдать экзамены 101 и 102 (и получить уровень сертификации 1), вы должны уметь:

- Работать в командной строке Linux
- Выполнять простые задачи обслуживания: помогать пользователям, добавлять пользователей в систему, выполнять резервирование и восстановление, останавливать и перезагружать систему
- Устанавливать и настраивать рабочую станцию (включая X) и подсоединять ее к локальной сети, или подключать изолированный компьютер с помощью модема к Интернет

Для продолжения подготовки к уровню сертификации 1, смотри [учебники developerWorks для экзаменов LPI 101 и 102](#), а также [полный набор учебников LPI от developerWorks](#).

Профессиональный Институт Linux не приветствует никаких учебных материалов и методов подготовки к экзаменам от третьих лиц. За подробными разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## Об этом учебнике

Добро пожаловать в "Загрузку, инициализацию, остановку и уровни выполнения" - второй в серии из девяти учебников, разработанных для подготовки вас к экзамену LPI 102. В этом учебнике вы научитесь загружать систему, задавать параметры ядра, останавливать или перезагружать систему.

Этот учебник построен в соответствии с программой LPI для этой темы. Для приблизительной оценки, на экзамене ожидайте больше вопросов по темам с более высоким рейтингом.

*Таблица 2. Загрузка, инициализация, остановка и уровни выполнения: Программы экзаменов, охватываемые этим учебником*

<b>Тема экзамена LPI</b>	<b>Рейтинг темы</b>	<b>Краткое содержание темы</b>
1.106.1 <u>Загрузка системы</u>	Рейтинг 3	Проведение загрузки системы, включая передачу команд загрузчику и установку параметров ядра во время загрузки. Изучение того, как просмотреть события загрузки в файлах журналов.
1.106.2 <u>Изменение уровней выполнения, остановка и перезагрузка системы</u>	Рейтинг 3	Управление уровнем выполнения системы, установка уровня выполнения по умолчанию, переход в однопользовательский режим, остановка и перезагрузка системы. Изучение того, как предупреждать пользователей перед сменой уровня выполнения, и как правильно завершать процессы.

## Требования

Чтобы извлечь максимум из этого учебника, вы должны иметь основные знания Linux и работающую систему Linux, чтобы выполнять команды, рассматриваемые в этом учебнике.

Этот учебник построен на основе содержания предыдущих учебников в серии LPI, поэтому вы, возможно, захотите сперва просмотреть [учебники для экзамена 101](#). Вы должны быть особенно хорошо знакомы с материалом из учебника "[Учебник для экзамена LPI 101 \(тема 102\): Установка Linux и управление пакетами](#)".

Разные версии программы могут форматировать вывод по-разному, поэтому ваши результаты могут не выглядеть в точности как на листингах и рисунках этого учебника.

# Учебник для экзамена LPI 102, Тема 106: Загрузка, инициализация, остановка и уровни выполнения

*Администрирование Linux для начинающих (LPIC-1), тема 106*

[Ian Shields](#), Старший программист, EMC

**Описание:** В этом учебнике Ян Шилдс (Ian Shields) продолжает готовить вас к сдаче Экзамена LPI 102 Профессионального Института Linux (Linux Professional Institute®): Администрирование Linux для начинающих (LPIC-1). В этом втором [в серии из девяти учебников](#) Ян знакомит вас с запуском и остановкой Linux®. К концу этого учебника вы будете знать, как провести загрузку системы, задать параметры ядра и остановить или перезагрузить систему.

[Больше статей из этой серии](#)

**Дата:** 04.04.2006

**Уровень сложности:** средний

## Загрузка системы

Этот раздел содержит материал для темы 1.106.1 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Тема имеет рейтинг 3.

В этом разделе вы научитесь:

- Проводить загрузку системы
- Давать команды загрузчику при загрузке
- Передавать параметры ядру во время загрузки
- Просматривать события загрузки в файлах журналов

## Обзор загрузки

Вкратце, процесс загрузки персональных компьютеров заключается в следующем:

1. При включении компьютера BIOS (*Basic Input/Output System - базовая система ввода-вывода*) проводит самотестирование.
2. Когда компьютер проходит самотестирование, BIOS загружает *главную загрузочную запись* (или *MBR*, обычно из первого 512-байтового сектора загрузочного устройства). Обычно это первый жесткий диск системы, но также может быть и дискетта, CD-диск или USB-ключ.
3. Для жесткого диска MBR загружает первичный загрузчик, которым обычно является загрузчик LILO или загрузчик GRUB для системы Linux. Это другая односекторная запись размером 512 байт.
4. Первичный загрузчик обычно загружает последовательность записей, называемую вторичным загрузчиком (или иногда "полуторным" ("stage 1.5") загрузчиком).
5. Вторичный загрузчик загружает операционную систему. Для Linux это ядро и, возможно, начальный RAM-диск (*initrd*).

К этому моменту ваша система должна быть готова к установке одного из двух популярных загрузчиков: LILO (LInux LOader) или GRUB (GRand Unified Boot loader). Вы должны быть способны использовать выбранный вами загрузчик для обычной загрузки, описанной выше. Отсылаем вас к учебнику "[учебник для экзамена LPI 101 \(тема 102\): установка Linux и управление пакетами](#)", если вам нужно просмотреть установку загрузчика или основную загрузку.

Для воздействия на процесс загрузки вашей системы, вы можете:

1. Изменять устройство, с которого вы загружаетесь. Обычно вы загружаетесь с жесткого диска, но иногда вам может потребоваться загрузиться с дискеты, ключа памяти USB, CD или DVD привода или через сеть. Установка таких альтернативных загрузочных устройств требует, чтобы ваша BIOS была соответственно сконфигурирована. Способ, как сделать это, зависит от вашей системы и ее BIOS. Это выходит за рамки этого учебника или требований этого объекта LPI, так что обращайтесь к вашей системной документации.
2. Вы можете взаимодействовать с загрузчиком, выбирая, какую из нескольких возможных конфигураций вы хотите загрузить. В этом учебнике вы научитесь делать это для загрузчиков LILO и GRUB.
3. Вы можете использовать GRUB или LILO, как только загрузчик его загрузит, для передачи параметров ядру, чтобы управлять тем, как ядро запускает систему.

## LILO

По умолчанию конфигурационный файл LILO находится в `/etc/lilo.conf`. Листинг 1 демонстрирует пример из системы, на которой работает Red Hat Enterprise Linux 3. Система имеет раздел Red Hat 9 с корневой файловой системой, монтированной на `/mnt/hda7`, и Windows® XP на `/dev/hda1`.

## Листинг 1. Пример LILO конфигурации

```
[root@lyrebird root]# cat /etc/lilo.conf
prompt
timeout=50
compact
default=latest-EL
boot=/dev/fd0
map=/boot/map
install=/boot/boot.b
message=/boot/message2
lba32
password=mypassword
restricted

image=/mnt/hda7/boot/vmlinuz-2.4.20-31.9
    label=redhat9
    alias=shrike
    initrd=/mnt/hda7/boot/initrd-2.4.20-31.9.img
    read-only
    append="hdd=ide-scsi root=LABEL=RH9"

image=/boot/vmlinuz-2.4.21-40.EL
    label=2.4.21-40.EL
    alias=latest-EL
    initrd=/boot/initrd-2.4.21-40.EL.img
    read-only
    append="hdd=ide-scsi root=LABEL=RHEL3"

image=/boot/vmlinuz-2.4.21-37.0.1.EL
    label=2.4.21-37a.EL
    initrd=/boot/initrd-2.4.21-37.0.1.EL.img
    read-only
    append="hdd=ide-scsi root=LABEL=RHEL3"

image=/boot/vmlinuz-2.4.21-37.EL
    label=2.4.21-37.EL
    initrd=/boot/initrd-2.4.21-37.EL.img
    read-only
    append="hdd=ide-scsi root=LABEL=RHEL3"

image=/boot/vmlinuz-2.4.21-32.0.1.EL
    label=2.4.21-32.EL
    alias=early
    initrd=/boot/initrd-2.4.21-32.0.1.EL.img
    read-only
    append="hdd=ide-scsi root=LABEL=RHEL3"

other=/dev/hda1
    loader=/boot/chain.b
    label=WIN-XP
    alias=xp
```

Помните, что каждый раз, когда вы вносите изменения в /etc/lilo.conf или устанавливаете новое ядро, вы **должны** запустить **lilo**. Программа **lilo** переписывает MBR или загрузочную запись раздела, чтобы отразить внесенные вами изменения, включая занесение абсолютного адреса ядра на диск. Если ваш конфигурационный файл включает образы Linux из нескольких разделов, вы должны смонтировать разделы, потому что команда **lilo**

требует доступа к разделу, чтобы найти образ.

Параметр **message** в конфигурационном файле LILO может ссылаться на текстовый файл или на специально созданный файл в формате PCX. Дистрибутив Red Hat Enterprise Linux включает графический файл /boot/message, который содержит заставку Red Hat. С иллюстративной целью конфигурация из Листинга 1 использует текстовый файл, представленный в Листинге 2. Изменение графических сообщений LILO выходит за пределы этого учебника. Имейте в виду, что это также не очень хорошо задокументировано.

## Листинг 2. Текстовое загрузочное сообщение LILO

```
[root@lyrebird root]# cat /boot/message2
Booting lyrebird
```

Если ваш файл конфигурации не включает параметра сообщения, то вы увидите очень простую строку приглашения: **LILO boot:**. В противном случае вы увидите или текстовое сообщение или графический фон и меню. Возможно, вам потребуется удерживать клавишу Shift в процессе загрузки, чтобы видеть приглашение, так как система может быть сконфигурирована таким образом, чтобы пропускать его.

Если вы видите пользовательскую текстовую строку приглашения или строку по умолчанию, вы можете нажать клавишу Tab, чтобы отобразить список доступных образов для загрузки. Вы можете или ввести имя образа, как показано в Листинге 3, или нажать **Enter** чтобы выбрать первый вариант. Если вам доступно графическое меню, используйте клавиши управления курсором, чтобы выделить вариант, который вы хотите загрузить.

## Листинг 3. Пример строки приглашения LILO

```
LILO
Booting lyrebird

boot:
latest-EL      shrike          redhat9        2.4.21-40.EL
2.4.21-37a.EL  2.4.21-37.EL   early          2.4.21-32.EL
xp              WIN-XP
boot: latest-EL
```

Помимо отображения конфигурационного файла LILO, можно указать ключ **-q** команды **lilo** для отображения информации о вариантах загрузки LILO. Добавьте ключ **-V** для более содержательного вывода. Два примера, использующих конфигурационный файл из Листинга 1, приведены в Листинге 4.

## Листинг 4. Отображение конфигурации LILO

```
[root@lyrebird root]# lilo -q
latest-EL      *
shrike
redhat9
2.4.21-40.EL
2.4.21-37a.EL
2.4.21-37.EL
```

```

early
2.4.21-32.EL
xp
WIN-XP
[root@lyrebird root]# lilo -q -v | tail +22 | head -n 9
    shrike
        Password is required for specifying options
        Boot command-line won't be locked
        No single-key activation
        VGA mode is taken from boot image
        Kernel is loaded "high", at 0x00100000
        Initial RAM disk is 149789 bytes
        No fallback
        Options: "ro BOOTFILE=/mnt/hda7/boot/vmlinuz-2.4.20-31.9 hdd=ide-scsi
root=LABEL=RH9"

```

## GRUB

Файл конфигурации GRUB по умолчанию находится в /boot/grub/grub.conf или /boot/grub/menu.lst. Если присутствуют оба файла, то один обычно является символьной ссылкой на другой. Листинг 5 содержит пример из той же системы, которую вы рассматривали для LILO, хотя здесь иллюстрируется только несколько вариантов.

### Листинг 5. Пример конфигурации GRUB

```

default=1
timeout=10
splashimage=(hd0,2)/boot/grub/fig1x.xpm.gz
foreground=23334c
background=82a6bc
password -md5 $1$H8LlM1$cI0Lf5.C06xFJYPQ8Ixz/
title Red Hat Linux (2.4.20-31.9)
    root (hd0,6)
    kernel /boot/vmlinuz-2.4.20-31.9 ro root=LABEL=RH9 hdd=ide-scsi
    initrd /boot/initrd-2.4.20-31.9.img
    savedefault
    boot

title Red Hat Enterprise Linux WS A (2.4.21-40.EL)
    root (hd0,10)
    kernel /boot/vmlinuz-2.4.21-40.EL ro root=LABEL=RHEL3 hdd=ide-scsi
    initrd /boot/initrd-2.4.21-40.EL.img

title Win/XP
    rootnoverify (hd0,0)
    chainloader +1

```

GRUB предоставляет интерфейс меню вместо строки приглашения LILO. Он также может использовать пароль, зашифрованный алгоритмом MD5, в противоположность простому текстовому паролю LILO. И, что, может быть, наиболее важно, изменения, сделанные в конфигурационном файле GRUB, не требуют переустановки GRUB в MBR. Заметьте, что многие дистрибутивы автоматически обновляют конфигурационный файл GRUB (или LILO) при переходе на новую версию ядра, но если вы устанавливаете новое ядро самостоятельно или создаете новый начальный RAM-диск, вам, возможно, потребуется отредактировать конфигурационный файл.

GRUB также не требует монтирования раздела, чтобы сконфигурировать для него загрузочный образ. Вы заметите строки типа `root (hd0,6)` и `splashimage=(hd0,2)/boot/grub/fig1x.xpm.gz`. GRUB обращается к жестким дискам как `hd $n$` , где  $n$  - целое число от 0 и выше. Подобным образом нумеруются, начиная с нуля, и разделы на диске.

Итак, в этой системе (hd0,2) представляет основной раздел /dev/hda3, а (hd0,6) - логический раздел /dev/hda7. Флоппи-дисковод - это обычно (fd0). Не забывайте брать это в кавычки, если вы вызываете GRUB с параметрами из оболочки bash, например, при установке GRUB на флоппи-диск или MBR.

Если вам хочется изменить фоновое изображение для GRUB, то вы ограничены 14 цветами. Ваше любимое JPEG-изображение может выглядеть несколько иначе, когда будет сведено к 14 цветам. Вы можете видеть результат на Рисунке 1, где показано изображение из конфигурационного файла выше, использующее фото, сделанное мной в Гласьер Бэй на Аляске. Вам может также захотеться выбрать подходящие основной и фоновый цвета для вашей текстовой командной строки из цветов на изображении; Рисунок 2 иллюстрирует измененный цвет текста и фона.

**Рисунок 1. Фото, сведенное к 14 цветам, для фонового изображения GRUB**



**Рисунок 2. Цвет текста и его фона, выбранные из цветов фотографии**

Red Hat Enterprise Linux WS A (2.4.21-40.EL)

Когда отображается меню GRUB, вы выбираете загрузочный образ, используя клавиши управления курсором для перемещения по списку вверх и вниз.

В отличие от LILO, GRUB ведет себя как маленькая оболочка с несколькими командами, которые позволяют вам делать такие вещи, как редактирование команд перед их выполнением, или нахождение и загрузка конфигурационного файла, или отображение файлов с использованием команды `cat`. Находясь в меню, вы можете нажать `e` на строке, чтобы изменить ее, `c` - чтобы переключиться на командную строку GRUB, `b` - чтобы загрузить систему, `p` - чтобы ввести пароль и `Esc` - чтобы вернуться в меню или на предыдущий шаг. Существует также команда `grub`, которая создает эмуляцию оболочки, в которой вы можете протестировать вашу конфигурацию GRUB или ваши навыки использования команд GRUB. Внутри оболочки GRUB команда `help` выводит список команд. Использование `help commandname` предоставляет справку по команде с именем `commandname`. Листинг 6 иллюстрирует вызов справки и доступные команды.

## Листинг 6. Использование оболочки GRUB

```
[root@lyrebird root]# grub
Probing devices to guess BIOS drives. This may take a long time.
find FILENAME                                geometry DRIVE [CYLINDER HEAD SECTOR [
halt [--no-apm]                               help [-all] [PATTERN ...]
hide PARTITION                                 initrd FILE [ARG ...]
```

```

kernel [--no-mem-option] [--type=TYPE] makeactive
map T0_DRIVE FROM_DRIVE md5crypt
module FILE [ARG ...] modulenounzip FILE [ARG ...]
pager [FLAG] partnew PART TYPE START LEN
parttype PART TYPE quit
reboot root [DEVICE [HDBIAS]]
rootnoverify [DEVICE [HDBIAS]] serial [--unit=UNIT] [--port=PORT] [--stage2=STAGE2_
setkey [T0_KEY FROM_KEY] setup [--prefix=DIR] [--stage2=STAGE2_
terminal [--dumb] [--no-echo] [--no-ed] terminfo [--name=NAME --cursor-address
testvbe MODE unhide PARTITION
uppermem KBYTES vbeprobe [MODE]

grub> help rootnoverify
rootnoverify: rootnoverify [DEVICE [HDBIAS]]
    Similar to `root', but don't attempt to mount the partition. This
    is useful for when an OS is outside of the area of the disk that
    GRUB can read, but setting the correct root device is still
    desired. Note that the items mentioned in `root' which derived
    from attempting the mount will NOT work correctly.

grub>

```

В качестве практического упражнения вы можете продолжить предыдущий пример и использовать команду GRUB **find** для поиска файлов конфигурации. Далее вы можете загрузить файл конфигурации из (hd0,2), то есть из /dev/hda3, как показано в Листинге 7.

#### **Листинг 7. Использование GRUB для нахождения и загрузки конфигурационного файла GRUB**

```

grub> find /boot/grub/menu.lst
(hd0,2)
(hd0,6)
(hd0,7)
(hd0,8)
(hd0,9)
(hd0,10)

grub> configfile (hd0,2)/boot/grub/menu.lst

```

Когда вы загружаете конфигурационный файл, вы можете увидеть меню, подобное тому, которое показано в Листинге 8. Помните, что это было выполнено из оболочки GRUB, которая эмулирует реальное окружение GRUB и не отображает фоновое изображение. Однако, это по сути дела то, что вы увидите наложенным на ваше фоновое изображение, когда действительно будете загружать систему, используя GRUB.

#### **Листинг 8. Меню GRUB**

```

GRUB version 0.93 (640K lower / 3072K upper memory)

+-----+
| Red Hat Linux (2.4.20-31.9)
| Red Hat Linux (2.4.20-6)
| Red Hat Enterprise Linux WS A (2.4.21-40.EL)
| Red Hat Enterprise Linux WS A (2.4.21-37.0.1.EL)
| Red Hat Enterprise Linux WS A (2.4.21-37.EL)
+-----+

```

```
| Red Hat Enterprise Linux WS A (2.4.21-32.0.1.EL)
| Red Hat Enterprise Linux WS A (2.4.21-27.0.4.EL)
| Red Hat Enterprise Linux WS A (2.4.21-27.0.2.EL)
| Red Hat Enterprise Linux WS A (2.4.21-27.0.1.EL)
| Red Hat Enterprise Linux WS A (2.4.21-20.EL)
| Red Hat Enterprise Linux WS (2.4.21-27.0.1.EL)
+-----+ Red Hat Enterprise Linux WS (2.4.21-15.0.2.EL) v
+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS or 'p' to enter a
password to unlock the next set of features.
```

The highlighted entry will be booted automatically in 5 seconds.

Предположим, вы выделили третью строчку, то есть **Red Hat Enterprise Linux WS A (2.4.21-40.EL)**, и нажали **e**, чтобы редактировать ее. Вы увидите что-то подобное Листингу 9.

### Листинг 9. Редактирование строки конфигурации GRUB

```
GRUB version 0.93 (640K lower / 3072K upper memory)

+-----+
| root (hd0,10)
| kernel /boot/vmlinuz-2.4.21-40.EL ro root=LABEL=RHEL3 hdd=ide-scsi
| initrd /boot/initrd-2.4.21-40.EL.img
|
+-----+

Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('0' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

Вы снова с помощью клавиш управления курсором выбираете строку для изменения и затем нажимаете **e** чтобы ее редактировать. Например, если вы удалили раздел /dev/hda5, то вашим корневым разделом станет /dev/hda10 или (hd0,9) вместо /dev/hda11 или (hd0,10). Сохраните и измените значение. После этого нажмите **Enter**, чтобы подтвердить изменения, или **Esc** для отмены. Наконец, нажмите **b**, чтобы загрузить систему.

GRUB имеет достаточно возможностей, чтобы отображать файлы вашей файловой системы, и он запускается с правами суперпользователя, так что вам действительно нужно защищать вашу систему паролем GRUB. Однако, помните, что если пользователь может загрузиться с переносного носителя, то он может использовать свою собственную конфигурацию GRUB. Смотрите часть, связанную с безопасностью, в руководстве GRUB (см. [Ресурсы](#)) для большей информации о безопасности и других аспектах GRUB. Вы также можете просмотреть эту информацию на вашей системе, используя команду [info grub](#).

### Параметры ядра

Параметры ядра (иногда называемые параметрами загрузки) снабжают ядро информацией о параметрах оборудования, которые оно не может определить самостоятельно, чтобы

переназначить значения, которые оно может в противном случае неверно определить, или избежать задания неподходящих значений. Например, вы можете захотеть загрузить симметричную многопроцессорную систему (SMP) в однопроцессорном режиме или указать альтернативную корневую файловую систему. Некоторые версии ядер требуют параметр для разрешения поддержки больших объемов памяти на системах с количеством памяти выше определенного.

Если вы используете LILO, вы указываете дополнительные (или перекрывающие) параметры после того, как вводите имя загружаемого ядра. Например, если вы только что добавили новое ядро с названием /boot/vmlinuz-2.4.21-40.EL-prep, вам следует ввести команду, чтобы сказать LILO использовать его, как показано в Листинге 10.

### Листинг 10. Задание параметров загрузки с помощью LILO

```
boot: latest-EL image=/boot/vmlinuz-2.4.21-40.EL-prep
```

Используя GRUB, вы можете вводить другой набор команд для ядра и операторы `initrd` или, что предпочтительней, использовать средства редактирования, которые вы только что изучили, чтобы изменить существующую строчку, добавив `-prep` после имени существующего образа ядра.

Когда ядро заканчивает загрузку, оно обычно запускает `/sbin/init`. Эта программа продолжает работать, пока система не будет остановлена. Ей обычно присвоен идентификатор процесса (PID) 1, как это видно из Листинга 11.

### Листинг 11. Процесс init

```
[root@lyrebird root]# ps --pid 1
 PID TTY          TIME CMD
   1 ?        00:00:04 init
```

Программа `init` загружает оставшуюся часть вашей системы, запуская серии скриптов. Эти скрипты обычно находятся в `/etc/rc.d/init.d` или `/etc/init.d`, они реализуют такие службы, как установка имени хоста системы, проверка файловой системы на ошибки, монтируют дополнительных файловых систем, включение сети, запуск служб печати, и так далее. После выполнения скриптов `init` запускает программу, называемую `getty`, которая выводит приглашение для ввода логина на консоль. Графические экраны входа в систему обрабатываются по-другому, как вы изучали в учебнике "[Учебник для экзамена LPI 102, тема 105: Ядро.](#)"

Если ваша система загрузит ядро, но не сможет успешно запустить `init`, вы можете попробовать восстановить систему, указав альтернативную программу инициализации. Например, указание `init=/bin/sh` загрузит вашу систему в оболочку с правами суперпользователя, из которой вы получите возможность восстановить систему.

Вы можете узнать больше про доступные параметры загрузки, используя страницы руководства `man` для `bootparam` или просматривая файл `/usr/src/linux/Documentation/ramdisk.txt`, который может быть назван `/usr/src/linux-$(uname -r)/Documentation/kernel-parameters.txt` на некоторых системах.

Не приходится и говорить, что, если каждый раз при загрузке вы вынуждены использовать один и тот же набор дополнительных параметров, то вам следует добавить их в

конфигурационный файл. Не забудьте при этом перезапустить [lilo](#), если вы используете LILO.

## События загрузки

Во время загрузки Linux, на консоль выводится большое количество сообщений, описывающих загружающееся ядро, аппаратные средства вашей системы и другие вещи, связанные с ядром. Эти сообщения обычно быстро проскаивают, и вы, вероятно, не сможете прочитать их, несмотря на то что при загрузке есть задержки, когда процесс загрузки ожидает чего-то, например, при невозможности найти сервер времени или при необходимости проверки файловой системы. С появлением проекта Linux Bootsplash (см. [Ресурсы](#)) эти сообщения могут накладываться на графический фон, или они могут быть скрыты и заменены простой строкой состояния. Если ваш дистрибутив поддерживает скрытый режим, у вас обычно есть возможность переключиться обратно на отображение загрузочных сообщений, нажав какую-нибудь клавишу, например, F2.

## dmesg

Приятно иметь возможность вернуться и просмотреть сообщения ядра. Поскольку стандартный вывод связан с процессом, а ядро не имеет идентификатора процесса, оно помещает сообщения ядра (и модулей) в *буфер кольца ядра*. Вы можете отобразить буфер кольца ядра, используя команду [dmesg](#), отображающую эти сообщения в стандартный вывод. Конечно, вы можете перенаправить этот вывод в файл для последующего анализа или направить его разработчику ядра для отладочных целей. Листинг 12 показывает некоторые варианты вывода, которые вы можете увидеть.

### Листинг 12. Неполный вывод dmesg

```
[root@lyrebird root]# dmesg | head -n 30
Linux version 2.4.21-40.EL (bhcompile@hs20-bc1-7.build.redhat.com) (gcc version 3.2.3
20030502 (Red Hat Linux 3.2.3-54)) #1 Thu Feb 2 22:32:00 EST 2006
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
BIOS-e820: 00000000000e0000 - 0000000000100000 (reserved)
BIOS-e820: 0000000000100000 - 0000000005f6f0000 (usable)
BIOS-e820: 0000000005f6f0000 - 0000000005f6fb000 (ACPI data)
BIOS-e820: 0000000005f6fb000 - 0000000005f700000 (ACPI NVS)
BIOS-e820: 0000000005f700000 - 0000000005f780000 (usable)
BIOS-e820: 0000000005f780000 - 00000000060000000 (reserved)
BIOS-e820: 00000000fec00000 - 00000000fec10000 (reserved)
BIOS-e820: 00000000fee00000 - 00000000fee01000 (reserved)
BIOS-e820: 00000000ff800000 - 00000000ffc00000 (reserved)
BIOS-e820: 00000000fffffc00 - 0000000100000000 (reserved)
631MB HIGHMEM available.
896MB LOWMEM available.
NX protection not present; using segment protection
On node 0 totalpages: 391040
zone(0): 4096 pages.
zone(1): 225280 pages.
zone(2): 161664 pages.
IBM machine detected. Enabling interrupts during APM calls.
Kernel command line: ro root=LABEL=RHEL3 hdd=ide-scsi
ide_setup: hdd=ide-scsi
Initializing CPU#0
Detected 2392.059 MHz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 4771.02 BogoMIPS
Page-cache hash table entries: 524288 (order: 9, 2048 KB)
```

```
Page-pin hash table entries: 131072 (order: 7, 512 KB)
```

Буфер кольца ядра также используется для некоторых событий после того, как система загружена. Это определенные программные сбои и события "горячего" подключения. Листинг 13 показывает строку для программы, которая вызвала ошибку сегментации, и несколько строк, связанных с подключением ключа памяти USB.

### Листинг 13. Позднейшие события в буфере кольца ядра

```
[root@attic4 ~]# dmesg |tail -n 19
main[15961]: segfault at 0000000000529000 rip 0000000000403b5d rsp 00007fffffd15d00
error 6
usb 1-4.3: new high speed USB device using ehci_hcd and address 4
scsi5 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 4
usb-storage: waiting for device to settle before scanning
    Vendor: Sony      Model: Storage Media      Rev: 0100
    Type:  Direct-Access          ANSI SCSI revision: 00
SCSI device sdb: 1014784 512-byte hdwr sectors (520 MB)
sdb: Write Protect is off
sdb: Mode Sense: 43 00 00 00
sdb: assuming drive cache: write through
SCSI device sdb: 1014784 512-byte hdwr sectors (520 MB)
sdb: Write Protect is off
sdb: Mode Sense: 43 00 00 00
sdb: assuming drive cache: write through
    sdb: sdb1
sd 5:0:0:0: Attached scsi removable disk sdb
usb-storage: device scan complete
SELinux: initialized (dev sdb1, type vfat), uses genfs_contexts
```

### /var/log/messages

Как только ваша система запустилась из точки `/sbin/init`, ядро начинает записывать события в буфер кольца, как вы только что видели, а процессы используют демон `syslog` для журналирования сообщений, обычно в `/var/log/messages`. В отличие от буфера кольца, каждая строка `syslog` имеет отметку времени, а файл продолжает существовать между перезапусками системы. Этот файл - то место, куда вам следует в первую очередь заглянуть при ошибках, возникших на стадии выполнения скриптов `init` при загрузке.

Большинство демонов имеют имена, заканчивающиеся на 'd'. Листинг 14 показывает, как посмотреть несколько последних сообщений о состоянии демонов после перезагрузки.

### Листинг 14. Сообщения демонов из /var/log/messages

```
[root@lyrebird root]# grep "^\Apr.*d\:" /var/log/messages|tail -n 14
Apr  2 15:36:50 lyrebird kernel: hdd: attached ide-scsi driver.
Apr  2 15:36:52 lyrebird apmd: apmd startup succeeded
Apr  2 15:36:26 lyrebird rc.sysinit: Setting hostname lyrebird: succeeded
Apr  2 15:36:26 lyrebird rc.sysinit: Initializing USB keyboard: succeeded
Apr  2 15:36:55 lyrebird sshd: succeeded
Apr  2 15:36:55 lyrebird xinetd: xinetd startup succeeded
Apr  2 15:36:56 lyrebird ntpd: succeeded
Apr  2 15:36:56 lyrebird ntpd: succeeded
Apr  2 15:36:56 lyrebird ntpd: succeeded
```

```
Apr  2 15:36:56 lyrebird ntpd: ntpd startup succeeded
Apr  2 15:36:57 lyrebird crond: crond startup succeeded
Apr  2 15:36:58 lyrebird atd: atd startup succeeded
Apr  2 15:36:58 lyrebird snastart: insmod: streams: no module by that name found
Apr  2 15:36:58 lyrebird rhnsd: rhnsd startup succeeded
```

Вы также найдете журналы для многих других системных программ в `/var/log`. Например, вы можете посмотреть журнал запуска для вашей системы X Window, которую вы изучали в учебнике "[Учебник для экзамена LPI 101, Тема 110: Система X Window](#)."

## Уровни выполнения, остановка и перезагрузка

Этот раздел содержит материал для темы 1.106.2 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Тема имеет рейтинг 3.

В этом разделе вы научитесь:

- Управлять уровнем выполнения системы
- Переходить в однопользовательский режим
- Устанавливать уровень выполнения по умолчанию
- Выключать или перезагружать систему
- Предупреждать пользователей перед переключением уровня выполнения
- Правильно завершать процессы

### Уровни выполнения

Уровни выполнения определяют, какие задачи могут быть выполнены в текущем состоянии (или на текущем уровне выполнения) системы Linux. Каждая система Linux поддерживает три базовых уровня выполнения и один или более уровней для обычной работы. Базовые уровни выполнения показаны в Таблице 3.

*Таблица 3. Базовые уровни выполнения Linux*

Уровень	Назначение
0	Выключение (или остановка) системы
1	Однопользовательский режим; обычно имеет псевдоним <i>s</i> или <i>S</i>
6	Перезагрузка системы

Использование уровней выполнения, за исключением базовых, различается среди дистрибутивов. Один из распространенных наборов показан в Таблице 4.

*Таблица 4. Другие распространенные уровни выполнения Linux*

Уровень	Назначение
2	Многопользовательский режим без поддержки сети
3	Многопользовательский режим с поддержкой сети
5	Многопользовательский режим с поддержкой сети и системы X Window

Дистрибутив Slackware использует уровень выполнения 4 вместо 5 для полного запуска системы X Window. Debian использует один уровень выполнения для любого многопользовательского режима, обычно это уровень 2. Обращайтесь к документации для вашего дистрибутива.

## **Уровень выполнения по умолчанию**

При запуске Linux уровень выполнения по умолчанию определяется из строки **id**: файла /etc/inittab. Листинг 15 показывает типичную строку для такой системы, как Red Hat Enterprise Linux, использующей уровень выполнения 5 для системы X Window.

### **Листинг 15. Уровень выполнения по умолчанию из ./etc/inittab**

```
[root@lyrebird root]# grep "^id:" /etc/inittab  
id:5:initdefault:
```

## **Изменение уровня выполнения**

Есть несколько способов изменить уровень выполнения. Чтобы сделать постоянное изменение, можно отредактировать /etc/inittab и изменить уровень выполнения по умолчанию, как вы только что видели.

Если вам нужно только перевести систему на другой уровень выполнения, у вас есть пара способов сделать это. Например, представьте, что вы только что установили новое ядро, и вам нужно добавить некоторые модули после того, как система загрузилась с новым ядром, но до загрузки системы X Window. Вы, возможно, захотите перевести систему на уровень выполнения 3, чтобы выполнить это. Вы делаете это во время загрузки, редактируя строку ядра (GRUB) или добавляя параметр после имени выбранной системы (LILO). Используйте одну цифру для указания желаемого уровня выполнения (в данном случае 3). Например, вы можете изменить строку из Листинга 5, который вы видели в предыдущем разделе, как показано в Листинге 16.

### **Листинг 16. Установка уровня выполнения по умолчанию во время загрузки**

```
kernel /boot/vmlinuz-2.4.21-40.EL ro root=LABEL=RHEL3 hdd=ide-scsi 3
```

Как только вы закончили вашу работу на уровне выполнения 3, вы, возможно, захотели переключиться на уровень выполнения 5. К счастью, вам не надо перезагружать систему. Вы можете использовать команду **telinit**, чтобы сообщить процессу init, на какой уровень выполнения он должен переключиться.

Вы можете определить текущий уровень выполнения, используя команду **runlevel**, которая помимо текущего показывает и предыдущий уровень выполнения. Если первый выведенный символ - 'N', то уровень выполнения не менялся с того момента, как система была загружена. Листинг 17 иллюстрирует проверку и изменение уровня выполнения.

### **Листинг 17. Проверка и изменение уровня выполнения**

```
[root@lyrebird root]# runlevel  
N 3  
[root@lyrebird root]# telinit 5  
[root@lyrebird root]# runlevel  
3 5
```

Если вы используете команду **ls**, чтобы отобразить длинный листинг команды **telinit**, вы

увидите, что в действительности она является символьной ссылкой на команду `init`.

Команда `init` знает, каким образом она была вызвана, и ведет себя соответственно.

Поскольку `init` обычно имеет PID 1, она также достаточно интеллектуальна, чтобы знать, если вы запустили ее с помощью `init` а не `telinit`. В этом случае она предполагает, что вы хотите, чтобы она вела себя так, как будто вы вызвали `telinit` вместо нее. Например, вы можете использовать `int 5` вместо `telinit 5`, чтобы переключиться на уровень выполнения 5.

## Однопользовательский режим

В отличие от таких операционных систем для персональных компьютеров, как DOS или Windows, Linux по своей сути является многопользовательской системой. Однако, бывают ситуации, когда это становится проблемой, например, когда вам нужно восстановить основную файловую систему или базу данных, или установить и протестировать какое-нибудь новое оборудование. Уровень выполнения 1, или *однопользовательский режим*, - это ваше решение для таких ситуаций. Фактическая реализация зависит от дистрибутива, но обычно вы попадаете в оболочку с минимальными возможностями. Обычно отсутствует поддержка сети, нет (или очень мало) запущенных демонов. На некоторых системах вы должны пройти аутентификацию, но на других вы заходите прямо в оболочку как суперпользователь. Однопользовательский режим может быть вашим спасителем, но вы можете также разрушить систему, поэтому будьте осторожны всякий раз, когда заходите в систему с правами суперпользователя.

Также как и с переключением на обычные многопользовательские уровни выполнения, вы можете переключиться на однопользовательский режим, используя `telinit 1`. Как видно из Таблицы 3, 's' и 'S' - это псевдонимы для уровня выполнения 1, так что вместо этого вы можете, например, использовать `telinit s`.

## Команды остановки

Помимо использования `telinit` или `init` для остановки работы пользователей и перехода в однопользовательский режим, вы также можете вызвать команду `shutdown`. Команда `shutdown` отсылает сообщения с предупреждением всем зарегистрированным пользователям и блокирует дальнейшие входы в систему. Затем она посыпает сигнал `init` о переключении уровней выполнения. После этого процесс `init` посыпает всем запущенным процессам сигнал SIGTERM, давая им шанс сохранить данные или корректно завершить работу. После 5 секунд или другой задержки, если она указана, `init` посыпает сигнал SIGKILL, чтобы насильно завершить все оставшиеся процессы.

По умолчанию `shutdown` переключается на уровень выполнения 1 (однопользовательский режим). Вы можете указать ключ `-h`, чтобы остановить систему, или ключ `-r`, чтобы перезагрузиться. Стандартное сообщение выводится вместе с любым указанным вами сообщением. Время можно задать как абсолютное время в формате `hh:mm` или как относительное время `n`, где `n` - количество минут до выключения. Для немедленной остановки используйте `now`, что эквивалентно `+0`.

Если вы вызвали остановку с задержкой, и время еще не прошло, у вас есть возможность отменить остановку, нажав `Ctrl-c`, если команда работает в настоящий момент, или вызвав `shutdown` с ключом `-c` для отмены отложенной остановки. Листинг 18 содержит несколько примеров использования `shutdown`, а также отмены этой команды.

## Листинг 18. Примеры остановки

```
[root@lyrebird root]# shutdown 5 File system recovery needed
Broadcast message from root (pts/0) (Mon Apr  3 22:44:29 2006):
```

```
File system recovery needed
The system is going DOWN to maintenance mode in 5 minutes!

Shutdown cancelled.
[root@lyrebird root]# shutdown -r 10 Reloading updated kernel&
[1] 5388

Broadcast message from root (pts/0) (Mon Apr 3 22:45:15 2006):

Reloading updated kernel
The system is going DOWN for reboot in 10 minutes!
[root@lyrebird root]# fg
shutdown -r 10 Reloading updated kernel

Shutdown cancelled.
[root@lyrebird root]# shutdown -h 23:59&
[1] 5390
[root@lyrebird root]# shutdown -c

Shutdown cancelled.
[1]+ Done shutdown -h 23:59
```

Вы, возможно, заметили, что наш последний пример не вызвал сообщения с предупреждением. Если время до остановки превышает 15 минут, то сообщение не посыпается раньше, чем за 15 минут до этого события, как показано в Листинге 19. Листинг 19 также показывает использование ключа [-t](#) для увеличения задержки между сигналами SIGTERM и SIGKILL от 5 секунд до 60.

### Листинг 19. Другой пример остановки

```
[root@lyrebird root]# date;shutdown -t60 17 Time to do backups
Mon Apr 3 22:51:45 EDT 2006

Broadcast message from root (pts/0) (Mon Apr 3 22:53:45 2006):

Time to do backups
The system is going DOWN to maintenance mode in 15 minutes!
```

### Листинг 20. Перезагрузка системы

```
[root@lyrebird root]# reboot

Broadcast message from root (pts/0) (Mon Apr 3 22:58:27 2006):

The system is going down for reboot NOW!
```

Убедимся, что система перезагрузилась обратно на уровень выполнения 3, что видно из использования команд [runlevel](#) и [uptime](#) в Листинге 21.

### Листинг 21. Другой пример перезагрузки системы

```
[ian@lyrebird ian]$ /sbin/runlevel  
N 3  
[ian@lyrebird ian]$ uptime  
23:05:51 up 6 min, 1 user, load average: 0.00, 0.06, 0.03
```

Также возможно использовать **telinit** (или **init**) для остановки или перезагрузки системы. Так же как и в других случаях использования **telinit**, пользователям не будет послано никаких сообщений, и команда выполнится немедленно, хотя между сигналами SIGTERM и SIGKILL существует задержка. Про дополнительные ключи **telinit**, **init** и **shutdown** смотрите в соответствующих страницах руководства man.

### Остановка, перезагрузка и отключение питания

Вам следует знать еще несколько команд, связанных с остановкой и перезагрузкой.

- Команда **halt** останавливает систему.
- Команда **poweroff** является символьной ссылкой на команду **halt**, останавливающей систему и делающей затем попытку отключения питания.
- Команда **reboot** - это другая символьная ссылка на команду **halt**, останавливающая систему и затем перезагружающая ее.

Если какая-то из этих команд вызывается, когда система не находится на уровне выполнения 0 или 6, то вместо нее выполняется соответствующая команда **shutdown**.

Для получения информации о ключах, которые можно использовать с этими командами, а также более подробной информации об их работе, обращайтесь к страницам руководства man.

### Конфигурация уровня выполнения

Теперь вы, возможно, будете удивлены, почему нажатие **Ctrl-Alt-Delete** на некоторых системах вызывает перезагрузку, и как вообще конфигурируется поведение уровня выполнения. Помните поле idb /etc/inittab?

Что ж, существует еще несколько полей в /etc/inittab, а также набор инициализирующих скриптов в таких каталогах, как rc1.d или rc5.d, где цифра обозначает уровень выполнения, к которому применяются скрипты из этого каталога. Листинг 22 содержит строку для **Ctrl-Alt-Delete**, таким образом, вы видите, почему это вызывает перезагрузку системы.

### Листинг 22. Отлавливание ctrl-alt-delete

```
[root@lyrebird root]# grep -i ctrl /etc/inittab  
# Trap CTRL-ALT-DELETE  
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Скрипты, используемые **init** при запуске системы, изменении уровня выполнения или остановке системы, обычно хранятся в каталогах /etc/init.d или /etc/rc.d.init.d. Наборы символьных ссылок в каталогах rcn.d, по одному каталогу для каждого уровня выполнения *n*, определяют запускается ли скрипт при входе на уровень выполнения, или останавливается при выходе из него. Эти ссылки начинаются с "K" или "S", за которыми следует число из двух цифр и имя службы, как показано в Листинге 23.

### Листинг 23. Скрипты init

```
[root@lyrebird root]# find /etc -path "*rc[0-9]*.d/???au*"
/etc/rc.d/rc0.d/K95audit
/etc/rc.d/rc0.d/K72autofs
/etc/rc.d/rc1.d/K95audit
/etc/rc.d/rc1.d/K72autofs
/etc/rc.d/rc2.d/S20audit
/etc/rc.d/rc2.d/K72autofs
/etc/rc.d/rc3.d/S20audit
/etc/rc.d/rc3.d/S28autofs
/etc/rc.d/rc4.d/K95audit
/etc/rc.d/rc4.d/S28autofs
/etc/rc.d/rc5.d/S20audit
/etc/rc.d/rc5.d/S28autofs
/etc/rc.d/rc6.d/K95audit
/etc/rc.d/rc6.d/K72autofs
[root@lyrebird root]# cd /etc/rc.d/rc5.d
[root@lyrebird rc5.d]# ls -l ???a*
lrwxr-xr-x 1 root      root          15 Jan 11 2005 S20audit -> ../init.d/audit
lrwxr-xr-x 1 root      root          14 Jan 11 2005 S26apmd -> ../init.d/apmd
lrwxr-xr-x 1 root      root          16 Jan 11 2005 S28autofs -> ../init.d/autofs
lrwxr-xr-x 1 root      root          13 Jan 11 2005 S95atd -> ../init.d/atd
```

Здесь вы видите, что службы **audit** и **autofs** имеют *Knn* вхождений во всех уровнях выполнения и *Snn* вхождений для обеих в уровнях выполнения 3 и 5. "S" означает, что служба запускается при входе на уровень выполнения, в то время как "K" означает, что она должна быть остановлена. Компонент *nn* в имени ссылки определяет приоритетный порядок, по которому служба должна быть запущена или остановлена. В этом примере **audit** запускается до **autofs** и останавливается после.

Для большей информации о командах **init** и **inittab** обращайтесь к страницам руководства man.

## Ресурсы

### Научиться

- Оригинал руководства "[LPI exam 102 prep, Topic 106: Boot, initialization, shutdown, and runlevels](#)".
- Обзор всей [серии учебников для экзамена LPI](#) на developerWorks для изучения основ Linux и подготовки к сертификации по системному администрированию.
- В [Программе LPIC](#) можно найти списки заданий, примерные вопросы и подробные программы для трех уровней сертификации по системному администрированию Профессионального Института Linux.
- В "[Основных задачах для начинающих разработчиков Linux](#)" (developerWorks, Март 2005) узнайте, как открыть окно терминала или оболочку командной строки и многое другое.
- [Проект Linux Documentation](#) содержит множество полезных документов, в том числе HOWTO.
- [Руководство системного администратора Linux](#) знакомит начинающих с системным администрированием Linux.

- "[Рассмотрение загрузчиков: знакомство с LILO и GRUB](#)" (developerWorks, Август 2005) поможет вам противопоставить и сравнить эти два варианта.
- [Руководство по GRUB](#) содержит массу информации по GRUB.
- [HOWTO по GRUB Splash Image](#) поможет вам создать свое собственное фоновое изображение GRUB.
- Посмотрите [проект Bootsplash](#) о графическом процессе загрузки ядра Linux.
- [Небольшое HOWTO по LILO](#) рассказывает вам, как использовать Linux Loader.
- "[Автоматизация переключения ОС на систему Linux с альтернативной загрузкой](#)" (developerWorks, Март 2006) демонстрирует вам, как переключаться между Linux и Windows на одном компьютере без ручного вмешательства.
- [Кратко о сертификации LPI по Linux](#) (O'Reilly, 2001) и [Экзаменационный сборник LPIC 2: экзамены 101 и 102 на сертификацию Профессионального Института Linux \(Экзаменационный сборник 2\)](#) (Que, 2004) - ссылки для читателей, предпочитающих формат книги.
- Найдите еще [обучающие руководства для разработчиков Linux](#) в [разделе Linux developerWorks](#).
- Следите за [техническими событиями developerWorks](#) и [Web-трансляциями](#).

## **Получить продукты и технологии**

- Скачайте [trial-версии ПО от IBM](#) непосредственно с developerWorks.

# Подготовка к экзамену 102 в LPI, Тема 108: Документация по Linux

*Junior Level Administration (LPIC-1), тема 108*

[Ян Шилдс](#), главный программист, IBM developerWorks

**Описание:** В данном руководстве Ян Шилдс продолжает готовить вас к Экзамену 102 Linux Professional Institute® Junior Level Administration (LPIC-1). В этой четвертой части [серии из девяти учебников](#) Ян знакомит вас с документацией по Linux®. К концу данного учебника вы будете знать, как использовать и управлять локальной документацией, находить документацию в Интернете и использовать автоматизированные сообщения во время регистрации в системе для уведомления пользователей о системных событиях.

[Больше статей из этой серии](#)

**Дата:** 20.09.2006

**Уровень сложности:** средний

## Перед началом работы

Узнайте, чему могут научить вас данные учебники и как использовать их наиболее эффективно.

О данной серии

[Linux Professional Institute](#) (LPI) сертифицирует системных администраторов Linux на двух уровнях: *младшем уровне* (известным также под названием "certification level 1") и *промежуточном уровне* (известным также под названием "certification level 2"). Для достижения сертификационного уровня 1 вы должны сдать экзамены 101 и 102; для достижения сертификационного уровня 2 вы должны сдать экзамены 201 и 202.

developerWorks предлагает учебники, помогающие подготовиться к каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем, а каждая тема имеет на developerWorks соответствующий учебник для самостоятельного обучения. Для экзаменов 102 в LPI на developerWorks девятью темами и соответствующими им учебниками являются::

*Таблица 1. Экзамен 102 в LPI: Учебники и темы*

Тема экзамена	Учебник developerWorks	Резюме по учебнику
Тема 105	<a href="#">Подготовка к экзамену 102 в LPI: Ядро</a>	Узнайте, как установить и поддерживать Linux-ядра и модули ядра.
Тема 106	<a href="#">Подготовка к экзамену 102 в LPI: Начальная загрузка, инициализация, останов и уровни запуска (runlevel)</a>	Узнайте, как загрузить систему, установить параметры ядра и остановить или перезагрузить систему.
Тема 107	<a href="#">Подготовка к экзамену 102 в LPI: Печать</a>	Узнайте, как управлять принтерами, очередями печати и пользовательскими заданиями на печать в Linux-системе.

Тема 108	Подготовка к экзамену 102 в LPI: Документация	(Данное руководство). Узнайте, как использовать и управлять локальной документацией, искать документацию в Интернет и использовать автоматизированные сообщения во время регистрации в системе для уведомления пользователей о системных событиях. Ниже приведено детальное описание <a href="#">целей</a> данного учебника.
Тема 109	Подготовка к экзамену 102 в LPI: Командные процессоры (shell), написание сценариев, программирование и компилирование	Готовится к публикации.
Тема 111	Подготовка к экзамену 102 в LPI: Задачи администрирования	Готовится к публикации.
Тема 112	Подготовка к экзамену 102 в LPI: Основы организации сетей	Готовится к публикации.
Тема 113	Подготовка к экзамену 102 в LPI: Сетевые службы	Готовится к публикации.
Тема 114	Подготовка к экзамену 102 в LPI: Система защиты	Готовится к публикации.

Для сдачи экзаменов 101 и 102 (и получения сертификационного уровня 1) вы должны уметь:

- Работать с командной строкой Linux.
- Выполнять простые задачи по обслуживанию: оказывать помощь пользователям, добавлять пользователей в систему, выполнять процедуры резервного копирования и восстановления, останавливать и перезагружать систему.
- Устанавливать и настраивать рабочую станцию (включая X) и подключать ее к LAN, либо подключать автономный компьютер через модем к Интернет.

Linux Professional Institute не дает рекомендаций по любым сторонним материалам или методикам для подготовки к экзаменам. За подробностями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## О данном руководстве

Добро пожаловать в "Документацию по Linux", четвертую часть из девяти учебников, предназначенных для подготовки к экзамену 102 в LPI. В данном руководстве вы узнаете, как использовать и управлять локальной документацией, искать документацию в Интернет и использовать автоматизированные сообщения во время регистрации в системе для уведомления пользователей о системных событиях.

Данное руководство организовано на основании заданий LPI для данной темы. Скорее всего, на экзамене вам следует ожидать больше вопросов для заданий с повышенным весом

(weight).

Таблица 2. Документация: Задания экзамена, охваченные в данном учебнике

Задание экзамена в LPI	Вес задания	Резюме по заданию
1.108.1 <u>Использовать и управлять локальной системной документацией</u>	Вес 4	Найти соответствующие оперативные справочные страницы и разделы в них. Найти команды и оперативные справочные страницы, относящиеся к ним. Настроить доступ к источникам справочной информации и справочной системе. Подготовить оперативные справочные страницы для вывода на печать. Использовать системную документацию, хранящуюся в /usr/share/doc/, и определить, какую документацию хранить в /usr/share/doc/.
1.108.2 <u>Найти документацию по Linux в Интернет</u>	Вес 3	Используйте документацию по Linux, размещенную в таких источниках как Linux Documentation Project (LDP), Web-сайты поставщиков и сторонних организаций, новостные группы, архивы новостных групп и списки рассылки.
1.108.5 <u>Уведомить пользователей о системных проблемах</u>	Вес 1	Уведомить пользователей о текущих проблемах, связанных с системой, посредством сообщений, отображаемых во время регистрации в системе.

### Предварительные требования

Для наиболее эффективного использования данного учебника вы должны иметь базовые знания Linux и работающую Linux-систему, на которой можно пробовать команды, описанные в данном учебнике. Вы также должны быть подключены к Интернет.

Данный учебник основан на содержании предыдущих руководств этой LPI-серии, поэтому вы, возможно, захотите ознакомиться сначала с [учебниками для экзамена 101](#).

Различные версии программы могут по-разному форматировать выводимую информацию, поэтому результаты вашей работы могут выглядеть не совсем так, как в листингах и на рисунках в данном учебнике.

## Подготовка к экзамену 102 в LPI, Тема 108: Документация по Linux

*Junior Level Administration (LPIC-1), тема 108*

[Ян Шилдс](#), главный программист, IBM developerWorks

**Описание:** В данном руководстве Ян Шилдс продолжает готовить вас к Экзамену 102 Linux Professional Institute® Junior Level Administration (LPIC-1). В этой четвертой части [серии из девяти учебников](#) Ян знакомит вас с документацией по Linux®. К концу данного учебника вы будете знать, как использовать и управлять локальной документацией, находить документацию в Интернете и использовать автоматизированные сообщения во время регистрации в системе для уведомления пользователей о системных событиях.

[Больше статей из этой серии](#)

**Дата:** 20.09.2006

**Уровень сложности:** средний

## Локальная документация

Данный раздел охватывает материал по теме 1.108.1 для экзамена 102 Junior Level Administration (LPIC-1). Тема имеет вес 4.

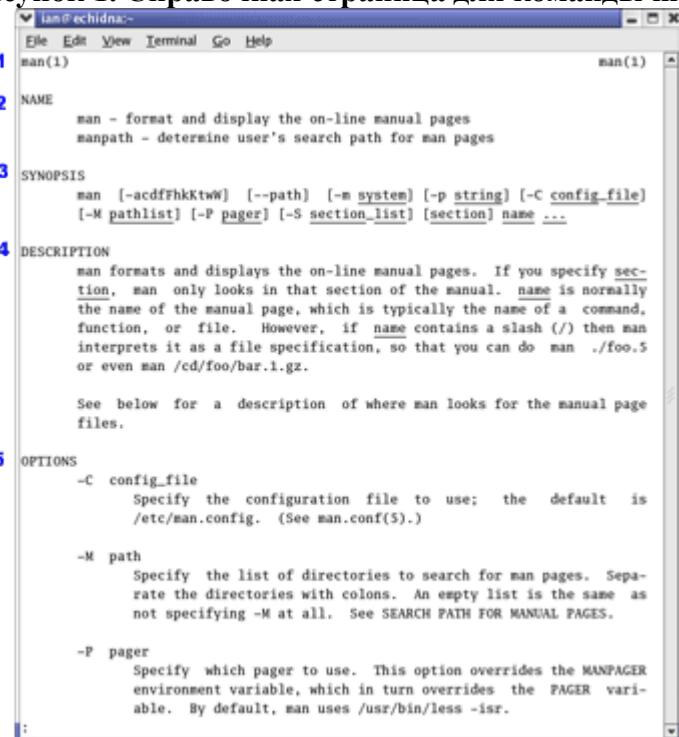
В данном разделе вы узнаете, как:

- Найти соответствующие оперативные справочные страницы.
- Найти разделы в оперативных справочных страницах.
- Найти команды и оперативные справочные страницы, относящиеся к ним.
- Настроить доступ к источникам справочной информации и к справочной системе.
- Подготовить оперативные справочные страницы для вывода на печать.
- Использовать системную документацию, размещенную в каталоге `/usr/share/doc/`, и определить, какую документацию оставить храниться в `/usr/share/doc/`.

## Поиск оперативных справочных страниц

Основным (и традиционным) источником документации являются оперативные справочные страницы, к которым вы можете обращаться, используя команду `man`. В идеальном случае вы можете просмотреть справочную страницу для любой команды, любого конфигурационного файла или любой библиотечной процедуры. На самом деле, Linux является бесплатным программным обеспечением, и некоторые страницы еще не были написаны, или были написаны очень давно. Тем не менее, эти страницы являются первым местом, куда нужно обращаться за помощью. На рисунке 1 изображена справочная страница для самой команды `man`. Используйте команду `man man` для отображения этой информации.

**Рисунок 1. Справочная страница для команды `man`**



На рисунке 1 показаны некоторые типичные элементы оперативных справочных страниц:

1. Заголовок с названием команды, за которым следует номер раздела в круглых скобках.
2. Название команды и все связанные с ней команды, описанные на этой же справочной странице.
3. Краткий обзор (synopsis) вариантов и параметров, применимых к команде.
4. Краткое описание команды.
5. Подробная информация по каждому параметру.

Возможно, вы найдете и другие разделы, например, как сообщить об ошибках, информацию об авторе и список всех связанных команд. Например, справочная страница для `man` указывает, что связанными командами (и их разделами) являются:

`apropos(1)`, `whatis(1)`, `less(1)`, `groff(1)` и `man.conf(5)`.

Справочные страницы отображаются при помощи *программы разбивки на страницы* (pager), которая обычно является командой `less` в Linux-системах. Вы можете установить это, используя переменную среды `$PAGER`, либо используя в команде `man` параметр `-P` или `--pager` вместе с названием другой программы разбиения на страницы. Программа разбиения на страницы принимает входную информацию со `stdin`, поэтому нечто похожее на работающий с файлами редактор так работать не будет.

Существует восемь общих разделов справочных страниц. Эти страницы обычно устанавливаются при установке пакета, поэтому, если у вас пакет не установлен, справочной страницы для него, вероятно, существовать не будет. Аналогично, некоторые из ваших разделов справки могут быть пусты или почти пусты. Общими разделами справочной страницы, с некоторыми примерами содержимого, являются:

1. Пользовательские команды (`env`, `ls`, `echo`, `mkdir`, `tty`).
2. Системные вызовы или функции ядра (`link`, `sethostname`, `mkdir`).
3. Библиотечные подпрограммы (`acosh`, `asctime`, `btree`, `locale`, `XML::Parser`).
4. Информация об устройстве (`isdn_audio`, `mouse`, `tty`, `zero`).
5. Описания файловых форматов (`keymaps`, `motd`, `wvdial.conf`).
6. Игры ( обратите внимание на то, что многие игры в настоящее время имеют графический интерфейс и графическую справочную информацию, размещаемую вне системы оперативных справочных страниц).
7. Разное (`arp`, `boot`, `regex`, `unix utf8`).
8. Системное администрирование (`debugfs`, `fdisk`, `fsck`, `mount`, `renice`, `rpm`).

Остальными разделами справочных страниц, которые вы можете обнаружить, являются: `9` (для документации по Linux-ядру), `n` (для новой документации), `o` (для старой документации) и `l` (для локальной документации).

Некоторые записи появляются в нескольких разделах. В наших примерах `mkdir` размещается в разделах 1 и 2, а `tty` в разделах 1 и 4.

## Команда `info`

В дополнение к стандартным оперативным справочным страницам Free Software Foundation создала большое количество `info`-файлов, которые обрабатываются программой `info`. Они обеспечивают широкие возможности навигации, включая способность мгновенно переходить к другим разделам. Дополнительную информацию можно получить при помощи команд `man info` или `info info`. Не все команды задокументированы в `info`, поэтому вы будете использовать и `man`, и `info`. Вы можете также начать с вершины древовидного списка `info`, используя `info` без параметров, как показано в листинге 1.

## Листинг 1. Команда `info`

```
File: dir,      Node: Top      This is the top of the INFO tree
```

This (the Directory node) gives a menu of major topics.  
Typing "q" exits, "?" lists all Info commands, "d" returns here,  
"h" gives a primer for first-timers,  
"mEmacs<Return>" visits the Emacs manual, etc.

In Emacs, you can click mouse button 2 on a menu item or cross reference  
to select it.

\* Menu:

Utilities

- |                         |   |
|-------------------------|---|
| * Bash: (bash).         | The GNU Bourne-Again SHell.             |
| * Enscript: (enscript). | GNU Enscript                            |
| * Gzip: (gzip).         | The gzip command for compressing files. |
| * ZSH: (zsh).           | The Z Shell Manual.                     |

Libraries

- |                         |   |
|-------------------------|---|
| * AA-lib: (aolib).      | An ASCII-art graphics library             |
| * History: (history).   | The GNU history library API               |
| * Libxmi: (libxmi).     | The GNU libxmi 2-D rasterization library. |
| * Readline: (readline). | The GNU readline library API              |

Texinfo documentation system

- |                 |                                |
|-----------------|--------------------------------|
| * Info: (info). | Documentation browsing system. |
|-----------------|--------------------------------|

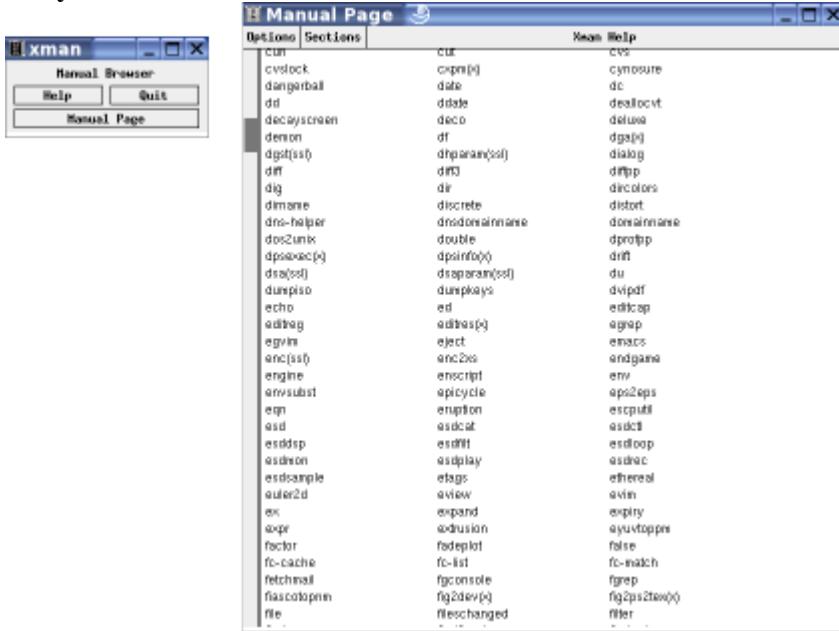
-----Info: (dir)Top, 2104 lines --Top-----  
Welcome to Info version 4.6. Type ? for help, m for menu item.

## Графические интерфейсы справочных страниц

В дополнение к стандартной команде [man](#), которая использует окно терминала и программу разбиения на страницы, ваша система может иметь также один или более графических интерфейсов к справочным страницам, таких как [xman](#) (из XFree86 Project) и [yelp](#) (браузер справочной системы Gnome).

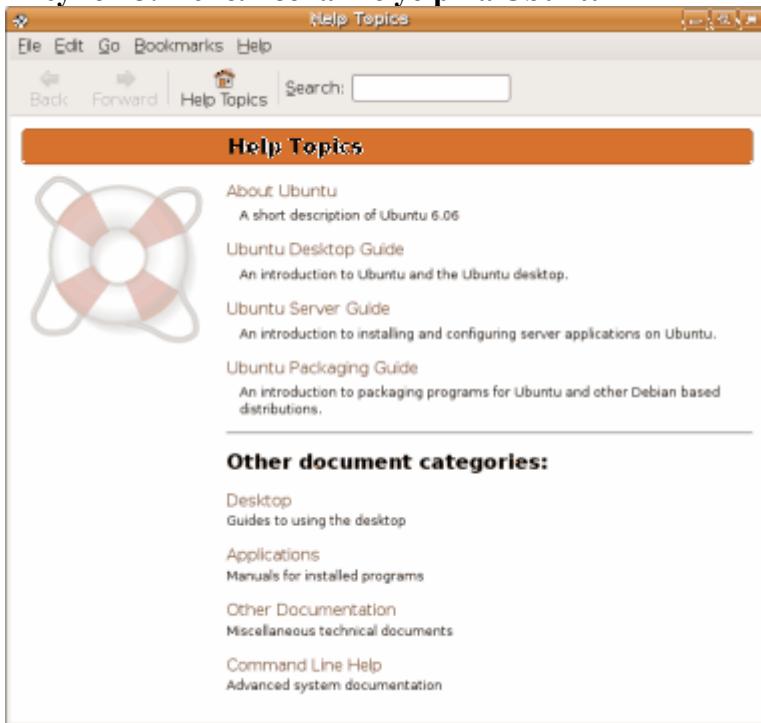
При запуске [xman](#) вы увидите небольшое окно с тремя кнопками. Нажмите кнопку Manual Page, чтобы открыть большое окно, в котором вы можете выполнять навигацию по справочным страницам или искать информацию. На рисунке 2 показан пример обоих окон.

**Рисунок 2. Использование xman**



Браузер **yelp** обычно выглядит по-разному на различных системах. На рисунке 3 показан пример работы на системе Ubuntu 6.06. Вы можете обращаться либо к справочным страницам, либо к info-страницам, используя элемент **Command Line Help** внизу экрана.

**Рисунок 3. Использование yelp на Ubuntu**



### Поиск справочных страниц

Если вы знаете, что тема появляется в определенном разделе, то можете указать этот раздел. Например, **man 4 tty** или **man 2 mkdir**. В качестве альтернативы можно использовать параметр **-a** для отображения всех доступных разделов справочной системы. Если вы укажете **-a**, будет отображаться запрос после выхода из страницы для каждого раздела. Вы можете перейти на следующую страницу, просмотреть ее или вообще выйти.

Как вы видели ранее, некоторые темы существуют в более чем одном разделе. Если вы не

хотите просматривать каждый раздел, то можете использовать параметр **-aw** команды **man** для получения списка всех доступных справочных страниц для темы. В листинге 2 приведен пример для команды **printf**. Если бы вы писали переносимый командный сценарий, то, вероятно, поинтересовались бы командой **man 1p printf** для изучения POSIX-версии команды **printf**. С другой стороны, если вы писали бы программу на C или C++, то больше заинтересовались бы командой **man 3 printf**, которая показала бы документацию по библиотечным функциям **printf**, **fprintf**, **sprintf**, **snprintf**, **vprintf**, **vfprintf** и **vsnprintf**.

## Листинг 2. Доступные справочные страницы для **printf**

```
ian@lyrebird:~> man -aw printf
/usr/share/man/man1/printf.1.gz
/usr/share/man/man1p/printf.1p.gz
/usr/share/man/man3/printf.3.gz
```

Команда **man** разбивает на страницы выводимую на ваш дисплей информацию при помощи специальной программы разбиения на страницы. В большинстве Linux-систем такой программой, вероятнее всего, будет программа **less**. Еще одним вариантом может быть более старая программа **more**.

Программа **less** имеет несколько команд, которые помогут вам искать строки в отображаемой информации. Они похожи на команды редактирования в **vi**. Используйте команду **man less** для поиска дополнительной информации по командам **/** (прямой поиск), **?** (обратный поиск) и **n** (повторить последний поиск), а также по многим другим командам.

Команда **info** пришла от создателей **emacs**, поэтому команды поиска больше похожи на **emacs**-команды. Например, **ctrl-s** ищет в прямом направлении, а **ctrl-r** ищет в обратном направлении, используя инкрементный поиск. Вы можете также перемещаться при помощи клавиш движения курсора, следуя по ссылкам (отмеченным звездочкой) при помощи клавиши **Enter**, и выйти при помощи клавиши **q**. Используйте параметр **--vi-keys** с командой **info**, если вы предпочитаете комбинации клавиш, аналогичные используемым для **man**.

## Команды поиска

Двумя важными командами, связанными с **man**, являются **whatis** и **apropos**. Команда **whatis** ищет справочные страницы для указанного вами имени и отображает информацию о нем из соответствующих справочных страниц. Команда **apropos** выполняет поиск справочных страниц по ключевому слову и выводит те из них, в которых содержится указанное вами ключевое слово. В листинге 3 продемонстрированы эти команды.

## Листинг 3. Примеры **whatis** и **apropos**

```
[ian@lyrebird ian]$ whatis man
man          (1) - format and display the on-line manual pages
man          (7) - macros to format man pages
man [manpath]      (1) - format and display the on-line manual pages
man.conf [man]     (5) - configuration data for man
[ian@lyrebird ian]$ whatis mkdir
mkdir        (1) - make directories
mkdir        (2) - create a directory
[ian@lyrebird ian]$ apropos mkdir
mkdir        (1) - make directories
mkdir        (2) - create a directory
```

```
mkdirhier          (1x) - makes a directory hierarchy
```

Кстати говоря, если вы не можете найти справочную страницу для `man.conf`, попробуйте выполнить команду `man man.config`, которая работает на некоторых системах.

Команда `apropos` может генерировать большой объем выводимой информации, поэтому, возможно, понадобится использовать более сложные регулярные выражения вместо простых ключевых слов. В качестве альтернативного метода вы можете захотеть отфильтровать выводимую информацию, используя `grep` или другой фильтр для уменьшения ее объема. В качестве практического примера вы можете использовать `e2label` для отображения или изменения метки файловой системы `ext2` или `ext3`, но для файловой системы `ReiserFS` вы должны использовать другую команду, чтобы изменить метку. Предположим, что вы выполняете команду `mount` для отображения смонтированных файловых систем, как показано в листинге 4.

#### Листинг 4. Смонтированные файловые системы ReiserFS

```
ian@lyrebird:~> mount -t reiserfs  
LABEL=SLES9 on / type reiserfs (rw,acl,user_xattr)
```

Теперь вы хотели бы узнать, какой раздел соответствует метке `SLES9`, но вы не можете вспомнить команду. Использование `apropos label` может выдать вам пару дюжин ответов, что не слишком плохо для анализа. Но подождите. Эта команда должна что-то сделать с файловой системой тома. Поэтому вы пробуете регулярные выражения, приведенные в листинге 5.

#### Листинг 5. Использование apropos с регулярными выражениями

```
ian@lyrebird:~> apropos "label.*file"  
e2label (8)           - Change the label on an ext2/ext3 filesystem  
ntfslabel (8)        - display/change the label on an ntfs file system  
ian@lyrebird:~> apropos "label.*volume"  
label.*volume: nothing appropriate.
```

Не совсем то, что вы ищете. Можно попытаться изменить порядок терминов в регулярных выражениях, либо попробовать отфильтровать информацию при помощи `grep` или `egrep`, как показано в листинге 6.

#### Листинг 6. Фильтрация выводимой командой apropos информации

```
ian@lyrebird:~> apropos label | grep -E "file|volume"  
e2label (8)           - Change the label on an ext2/ext3 filesystem  
mlabel (1)            - make an MSDOS volume label  
ntfslabel (8)         - display/change the label on an ntfs file system  
findfs (8)           - Find a filesystem by label or UUID
```

И здесь мы находим команду, которая нам нужна, - `findfs`. Используя ее так, как показано в листинге 7, мы увидим, что файловая система размещена на `/dev/hda10` этой конкретной

системы.

### Листинг 7. Поиск устройства для смонтированной метки файловой системы

```
ian@lyrebird:~> /sbin/findfs LABEL=SLES9  
/dev/hda10
```

Обращаем внимание на то, что отличные от root пользователи обычно должны указывать полный путь к команде `findfs`.

Как следует из справочной страницы по команде `man`, можно использовать также `man -k` вместо `apropos` и `man -f` вместо `whatis`. Поскольку при этом вызываются команды `apropos` или `whatis`, вероятно, нет большого смысла так делать.

### Конфигурация

Справочные страницы могут быть расположены в разных местах на вашей системе. Текущий путь поиска можно определить при помощи команды `manpath`. Если переменная окружения MANPATH установлена, для поиска справочных страниц будет использоваться она; в противном случае путь будет сформирован автоматически при помощи информации из конфигурационного файла, который мы вскоре обсудим. Если установлена переменная окружения MANPATH, команда `manpath` перед отображением пути выведет предупреждение об этом.

### Листинг 8. Отображение вашей переменной MANPATH

```
[ian@echidna ian]$ manpath  
/usr/local/share/man:/usr/share/man:/usr/X11R6/man:/usr/local/man  
  
ian@lyrebird:~> manpath  
manpath: warning: $MANPATH set, ignoring /etc/manpath.config  
/usr/local/man:/usr/share/man:/usr/X11R6/man:/opt/gnome/share/man
```

В зависимости от вашей системы конфигурационная информация для справочной системы хранится в файле `/etc/man.config` или `/etc/manpath.config`. Более старые системы используют `/etc/man.conf`. Текущий файл `man.config` содержит список каталогов (путей MANPATH), в которых будут искааться справочные страницы, например, список, показанный в листинге 9.

### Листинг 9. Записи MANPATH из /etc/man.config

```
MANPATH /usr/share/man  
MANPATH /usr/man  
MANPATH /usr/local/share/man  
MANPATH /usr/local/man  
MANPATH /usr/X11R6/man
```

В файле `manpath.config` эти записи будут указаны как `MANDATORY_MANPATH`, а не `MANPATH`.

Кроме этих записей вы найдете также записи, выдающие отображение между путями, по которым могут быть найдены исполняемые программы, и путями, в которых могут быть

размещены соответствующий справочные страницы, как показано в листинге 10.

#### Листинг 10. Записи MANPATH\_MAP из /etc/man.config

MANPATH_MAP	/bin	/usr/share/man
MANPATH_MAP	/sbin	/usr/share/man
MANPATH_MAP	/usr/bin	/usr/share/man
MANPATH_MAP	/usr/sbin	/usr/share/man
MANPATH_MAP	/usr/local/bin	/usr/local/share/man

Команда `man` использует сложный метод для поиска справочных страниц, а установка этих значений уменьшит усилия при их поиске.

Еще одна запись в конфигурационном файле определяет порядок поиска справочных страниц. Вспомните, что по умолчанию отображается первая найденная страница, поэтому этот порядок важен. Найдите внизу файла `man.config` строку `MANSECT`, либо внизу файла `manpath.config` строку `SECTION`. Проверьте конфигурационный файл на вашей системе, для того чтобы увидеть, что еще можно настроить.

Вы, возможно, заметили, что команды `apropos` и `whatis` работают быстро. Это происходит потому, что на самом деле они не выполняют поиск в отдельных справочных страницах. Вместо этого они используют базу данных, созданную командой `makewhatis`. Она обычно автоматически запускается системой ежедневно или еженедельно в виде задания `cron`.

#### Листинг 11. Выполнение makewhatis

```
[root@echidna root]# makewhatis
```

Эта команда обычно завершается без вывода какого-либо сообщения, но база данных `whatis` обновляется. Она обычно хранится в каталоге `/var/cache/man/whatis`. Обратите внимание на то, что некоторые системы SUSE не используют базу данных и, следовательно, не имеют команды `makewhatis`.

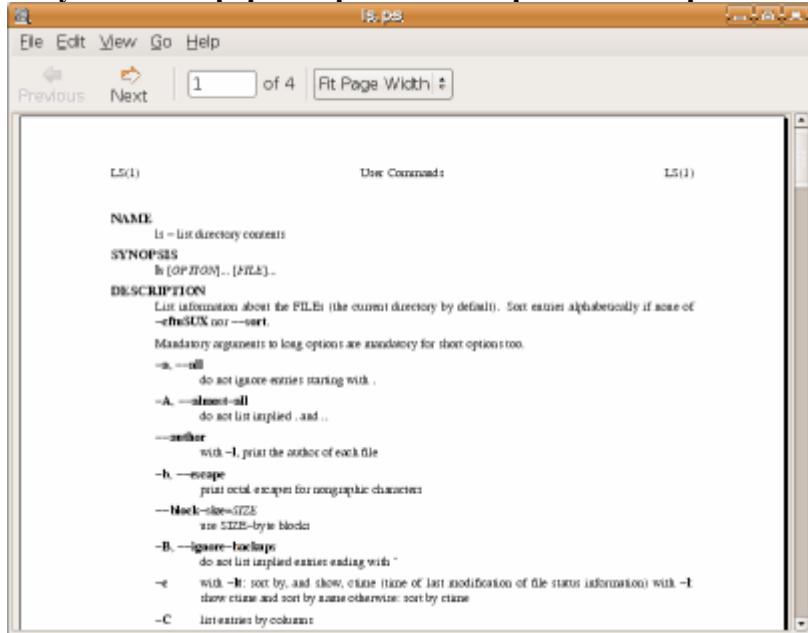
#### Вывод справочных страниц на печать

Если вы хотите распечатать страницу, укажите параметр `-t` для форматирования страницы при помощи программы `groff` или `troff`. При этом страница будет отформатирована для принтера по умолчанию, и вывод будет направлен в `stdout`. В листинге 12 показано, как отформатировать справочную страницу по команде `ls` и сохранить выводимые результаты в файл `ls.ps`. На рисунке 4 показана отформатированная выводимая информация.

#### Листинг 12. Форматирование справочной страницы по ls для вывода на печать

```
ian@pinguino:~$ man -t ls > ls.ps
```

**Рисунок 4. Отформатированная справочная страница по команде ls**



Если вам нужно отформатировать страницу для другого типа устройства, используйте параметр **-T** с типом устройства, например, dvi или ps. Дополнительная информация приведена в справочной странице по команде `man`.

#### /usr/share/doc/

Кроме справочных страниц и info-страниц, которые вы уже видели, ваша Linux-система, возможно, содержит много документации. Привычным местом для ее хранения является каталог /usr/share/doc, или /usr/doc для более старых систем. Эта дополнительная информация может быть представлена в различных форматах, например, как текст, PDF, PostScript или HTML.

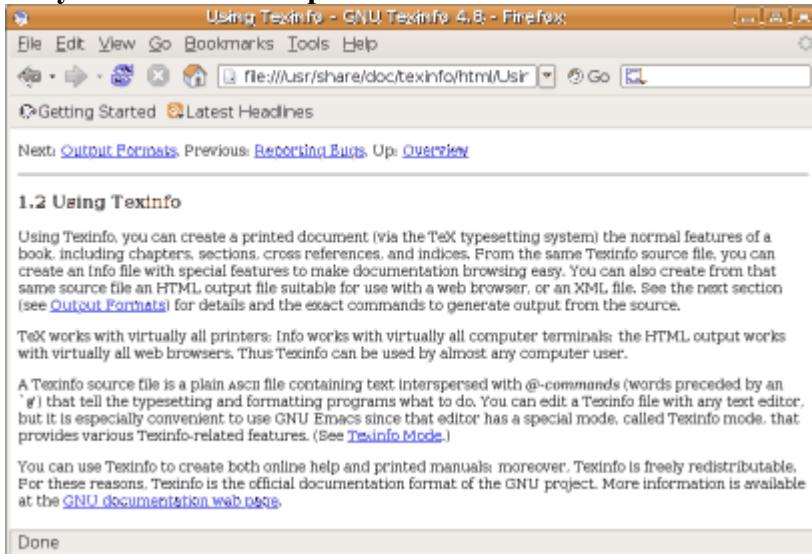
При просмотре этой документации иногда можно обнаружить ценную информацию, недоступную в справочных страницах или info-страницах, например, руководства или дополнительную техническую документацию. Как показано в листинге 13, в каталоге /usr/share/doc можетиться много файлов, то есть, у вас есть большое количество ресурсов для чтения.

#### Листинг 13. Файлы в /usr/share/doc

```
ian@pinguino:~$ find /usr/share/doc -type f | wc -l
10144
```

На рисунке 5 показан пример HTML-справки для системы Texinfo, которая используется командой `info`, которую вы видели ранее.

**Рисунок 5. HTML-справка по Texinfo из /usr/share/doc**



Иногда справочная страница будет перенаправлять вас к другому источнику документации. В качестве примера в листинге 14 приведена справочная страница для команды **pngtopnm**. Она направляет вас к локальной копии в HTML-формате /usr/share/doc/packages/netpbm/doc/pngtopnm.html, либо к интерактивной версии, если у вас нет локальной копии.

#### **Листинг 14. Указатель справочной страницы для pngtopnm**

**pngtopnm(1)**      Netpbm pointer man pages      **pngtopnm(1)**

pngtopnm is part of the Netpbm package. Netpbm documentation is kept in HTML format.

Please refer to  
<<http://netpbm.sourceforge.net/doc//pngtopnm.html>>.

If that doesn't work, also try <<http://netpbm.sourceforge.net>> and emailing Bryan Henderson, [bryanh@giraffe-data.com](mailto:bryanh@giraffe-data.com).

Local copy of the page is here:  
/usr/share/doc/packages/netpbm/doc/pngtopnm.html

#### **Другая справка по командам**

Наконец, если вы не можете найти справку по команде, попробуйте выполнить команду с параметром **--help**, **--h** или **--?**. При этом может отобразиться справочная информация по команде, либо указание о том, где найти нужную вам информацию. В листинге 15 показан пример для команды **kdesu**, которая обычно присутствует на системах с KDE.

#### **Листинг 15. Получение справки по команде kdesu**

```
ian@lyrebird:~> man kdesu
No manual entry for kdesu
ian@lyrebird:~> kdesu --help
Usage: kdesu [Qt-options] [KDE-options] command
```

Runs a program with elevated privileges.

**Generic options:**

--help	Show help about options
--help-qt	Show Qt specific options
--help-kde	Show KDE specific options
--help-all	Show all options
--author	Show author information
-v, --version	Show version information
--license	Show license information
--	End of options

**Arguments:**

command	Specifies the command to run.
---------	-------------------------------

**Options:**

-c <command>	Specifies the command to run. []
--------------	----------------------------------

В следующем разделе рассматриваются ресурсы справочной информации по Linux, доступные в режиме online.

## Интернет-документация

В данном разделе охватывается материал для темы 1.108.2 по экзамену 102 Junior Level Administration (LPIC-1). Тема имеет вес 3.

В данном разделе вы узнаете, как найти:

- Интерактивную документацию.
- Группы новостей (newsgroup).
- Списки рассылки.

## Интерактивная документация

Кроме документации, расположенной на вашей системе, существует много интерактивных источников документации и справочной информации.

### Linux Documentation Project

[Linux Documentation Project](#) - это результат работы добровольцев, которые собирают свободно распространяемую документацию по Linux. Этот проект существует для объединения различных частей документации по Linux в месте, в котором удобно искать и использовать эту информацию.

LDP состоит из следующих областей:

#### HOWTO

тематическая справочная информация, например, [Linux IPv6 HOWTO](#).

#### Guides (руководства)

более длинные и подробные книги, например, [Введение в Linux - практическое руководство](#).

#### FAQ

Frequently Asked Questions (часто задаваемые вопросы), например, [Linux Documentation Project \(LDP\) FAQ](#).

#### man pages (справочные страницы)

справочная информация по отдельным командам, которую вы использовали в предыдущем разделе данного учебника.

## **Linux Gazette**

интерактивный журнал, доступный в настоящее время на английском, французском, немецком, индонезийском, итальянском, португальском, русском и испанском языках.

Приведенные здесь примеры представляют собой многостраничные HTML-версии документации. Большинство статей предоставляются в нескольких форматах, включая одностораничный HTML, PDF, неформатированный текст и др.

LDP также предоставляет ссылки на [информацию, написанную на отличных от английского языках](#).

Сайт LDP хорошо спроектирован и имеет отличную навигацию. Если вы не уверены, какой именно раздел хотите просмотреть, воспользуйтесь полем для поиска, которое поможет искать информацию по теме.

Если вы хотите оказать помощь LDP в сборе документации по Linux, обязательно прочтите "[Руководство автора LDP](#)".

## **Дистрибуторские Web-сайты**

Web-сайты для различных дистрибутивов Linux часто предоставляют обновленную документацию, инструкции по установке, информацию по аппаратной совместимости/несовместимости и другую поддержку, например, средства поиска в базе знаний. Некоторые из этих сайтов перечислены ниже:

- [Redhat Linux](#) - крупный поставщик корпоративных Linux-продуктов, расположенный в Соединенных Штатах Америки.
- [SUSE Linux](#) - был основан в Германии и в настоящее время принадлежит компании Novell.
- [Asianux](#) - азиатский поставщик Linux, основанный компаниями Haansoft, Inc., Red Flag Software Co., Ltd. и Miracle Linux Corporation.
- [Turbolinux](#) имеет штаб-квартиру в Японии, но распространяет дистрибутивы Linux также и за пределами Азии.
- [Yellow Dog Linux](#) от Terra Soft Solutions - это дистрибутив для процессоров Apple PowerPC® и встроенных процессоров, построенных на базе процессоров PowerPC и Cell.
- [Linspire](#) - это настольная версия Linux, которую можно найти на некоторых предустановленных системах.
- [Slackware Linux Project](#) Патрика Волкердинга (Patrick Volkerding) существует с 1993 года и стремится быть самым "UNIX-подобным" дистрибутивом.
- [Debian GNU/Linux](#) существует с 1993 года как дистрибутив, созданный открыто в духе Linux и GNU.
- [Ubuntu Linux](#) - относительно новый дистрибутив Linux, основанный на Debian. Он концентрируется на удобстве использования и имеет связанные проекты: Kubuntu (версия, использующая настольную среду KDE), Edubuntu (разработанная для школ) и Xubuntu (облегченная версия, использующая настольную среду Xfce).
- [Gentoo Linux](#) - дистрибутив, который может быть автоматически оптимизирован и настроен для любого приложения или для любых потребностей. Пакеты распространяются в виде исходных кодов и компилируются под целевую среду.
- [Mandriva](#) - дистрибутив, которому свойственна простота использования. Компания была образована путем слияния нескольких пионеров программного обеспечения с открытыми исходными кодами, таких как Mandrakesoft из Франции, Conectiva из Бразилии, Edge IT из Франции и Lycoris из США.

Вы можете найти сводную информацию и ссылки на огромное число дистрибутивов Linux на

сайте [DistroWatch.com](#). Представленная в табличном виде информация по каждому дистрибутиву расскажет вам о том, какие основные пакеты включены в каждую версию, когда версия была выпущена, а также о многом другом.

### **Поставщики аппаратного и программного обеспечения**

Многие поставщики программного и аппаратного обеспечения за последние годы добавили поддержку Linux в свои продукты. На их сайтах вы сможете найти информацию о том, какая аппаратура поддерживает Linux, об инструментальных средствах разработки приложений, опубликованных исходных кодах, о загрузке Linux-драйверов для конкретной аппаратуры и о других специализированных Linux-проектах. Например:

- [IBM и Linux](#)
- [SGI и Linux](#)
- [HP и Linux](#)
- [Sun и Linux](#)
- [Набор офисных приложений StarOffice фирмы Sun](#)
- [Oracle и Linux](#)
- [BEA и Linux](#)

### **Проекты с открытым исходным кодом**

Многие проекты с открытым исходным кодом имеют домашние страницы, на которых вы найдете информацию по проектам. Некоторые проекты финансируются такой организацией, как Apache Software Foundation. Вот некоторые примеры:

- [Apache Software Foundation](#) - это родина Web-сервера Apache и многих других инструментальных программ.
- [Eclipse Foundation](#) - предоставляет независимую от поставщиков платформу разработки и интегрированную среду с открытым исходным кодом для создания программного обеспечения.
- [OpenOffice.org](#) - это мультиплатформенный и многоязычный пакет офисных приложений.
- [GNOME Foundation](#) - это родина настольной системы GNOME.
- [KDE project](#) - это родина настольной системы KDE (K Desktop Environment).

Большое количество проектов с открытыми исходными кодами размещается на [SourceForge.net](#). Они сгруппированы по категориям, например, кластеризация, базы данных, настольные, финансовые, мультимедийные приложения, системы защиты и т.д. Страницы проектов содержат ссылки на загружаемые файлы, отчеты об ошибках, форумы пользователей и ссылку на домашнюю страницу проекта (при наличии таковой), на которой вы обычно можете найти дополнительную информацию о проекте.

### **Другие ресурсы**

Другим местом, содержащим большое количество Linux-информации, является [зона IBM developerWorks Linux](#), на которой размещено данное руководство, а также много других отличных статей и руководств для Linux-разработчиков.

Многие печатные журналы также имеют интерактивные сайты, а некоторые новостные сайты существуют только в Web. Например:

- [LinuxWorld.com](#)
- [Slashdot](#)
- [freshmeat](#)
- [Linux Magazine](#) (на немецком языке)
- [Linux+](#) (на шести языках)

## Группы новостей

Группы новостей в Интернет являются, если быть более точным, формой дискуссионных списков. Они возникли из досок объявлений (bulletin board), которые были ранним средством разделения информации для совместного использования, обращение к которой осуществлялось обычно по коммутируемым каналам. Группы новостей используют протокол NNTP (*Network News Transfer Protocol*), который определен в IETF RFC 997 (февраль 1986).

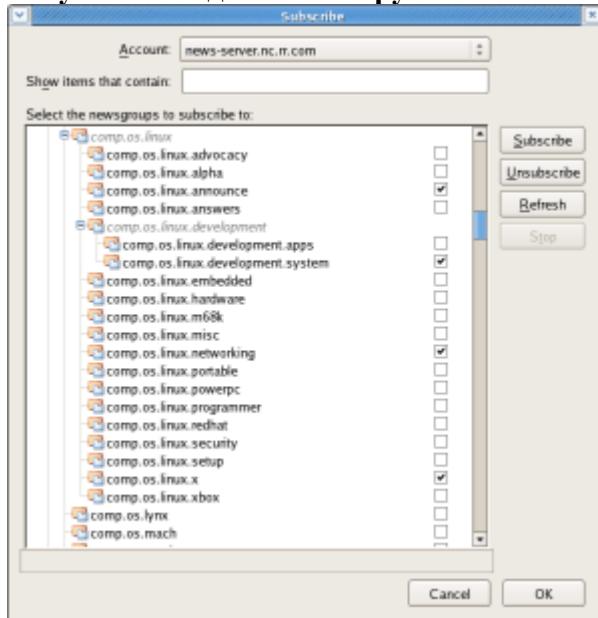
Чтобы принять участие в группе новостей, применяется программа чтения новостей, известная также под названием NNTP-клиент. Существует множество таких клиентов под Linux, включая **evolution**, **gnus**, **pan**, **slrn**, **thunderbird** и **tin**. Некоторые из них используют текстовый режим, некоторые графический интерфейс. Основным преимуществом группы новостей является то, что вы можете принять участие в дискуссии только при желании, вместо того, чтобы постоянно просматривать ваш почтовый ящик.

Usenet - это наибольший источник групп новостей. Существует несколько основных категорий, например, *comp* для компьютерной тематики, *sci* для научных дисциплин и *rec* для развлекательных тем, таких как хобби и игры. Компьютерная тематика в свою очередь разбивается на темы, которые тоже разбиваются на категории, например, группы новостей, которыми интересуются пользователи Linux, начинаются с *comp.os.linux*. Вы можете просмотреть этот [список на сайте LDP](#).

Ваш поставщик Интернет-услуг (Internet Service Provider), возможно, ведет зеркала нескольких групп новостей, хотя новые статьи могут задерживаться на длительный период, особенно для активных групп новостей. Некоторые поставщики групп новостей предлагают платные услуги, в которые входит более длительное хранение, быстрый доступ и более широкий выбор групп новостей.

На рисунке 6 показано дерево *comp.os.linux*, предоставляемое одним из ISP. В данном случае в качестве программы чтения новостей используется Mozilla Thunderbird. Вы подписываетесь на группы новостей, и ваша программа чтения новостей отображает только подписанные группы новостей. Подписанные группы новостей помечены галочкой.

Рисунок 6. Подписка на группы новостей *comp.os.linux*.\*



Обсуждения в группах новостей часто архивируются. Популярной группой новостей уже многие годы являлась Deja News. Когда ее финансирование прекратилось, архивы группы новостей были приобретены Google и представлены заново в виде [Google Groups](#).

В последнее время появились различные Web-форумы. Они обычно функционируют почти так же, как и группы новостей, но требуют использования только браузера и не требуют настройки. Примером является [форум технической поддержки Linux](#) на Web-сайте IBM developerWorks, на котором вы можете задать вопросы о данной серии учебников и по другим темам.

## Списки рассылки

Списки рассылки обеспечивают, вероятно, самый важный способ взаимодействия Linux-разработчиков. Часто проекты разрабатываются участниками, живущими очень далеко друг от друга, возможно, на разных концах земного шара. Списки рассылки преодолевают различия во временных зонах и обеспечивают, таким образом, взаимодействие каждого разработчика проекта с другими разработчиками, а также поддержку дискуссионных групп по электронной почте. Одним из наиболее известных списков рассылки для разработчиков является [Linux Kernel Mailing List](#).

Списки рассылки позволяют своим членам передавать сообщения в список, а сервер списка распространяет это сообщение всем членам группы. Конкретному члену группы не нужно знать почтовые адреса каждого члена группы, а также не нужно обслуживать списки текущих членов. Для того чтобы избежать потока (flood) сообщений от перегруженных списков, большинство списков рассылки позволяют пользователю запрашивать ежедневный *дайджест* - одно сообщение, содержащее все сообщения, опубликованные за день.

Кроме поддержки разработки списки рассылки могут обеспечивать возможность задавать вопросы и получать ответы от хорошо осведомленных разработчиков или даже от других пользователей. Например, отдельные дистрибуторы часто предоставляют списки рассылки для вновь прибывших пользователей. Вы можете просмотреть на сайте поставщика своего дистрибутива списки рассылки, которые он предоставляет.

Если у вас было время прочитать LKML FAQ, ссылка на который приведена выше, вы могли заметить, что подписчики списков рассылки не очень доброжелательно относятся к вопросам, задаваемым постоянно. Будет мудрее выполнить поиск по архивам данного списка рассылки перед тем, как задать вопрос. Есть шанс, что вы сэкономите свое время тоже. Что же касается архивов, то они часто дублируются на нескольких сайтах, поэтому используйте ближайший из них, обычно расположенный в вашей стране или на вашем континенте.

## Уведомление пользователей

В данном разделе приведен материал по теме 1.108.5 для экзамена 102 Junior Level Administration (LPIC-1). Тема имеет вес 1.

В данном разделе узнайте о том, как:

- Уведомить пользователей о текущих проблемах, связанных с функционированием системы, через сообщения, показываемые во время регистрации в системе.

## Сообщения во время регистрации

Последний короткий раздел данного учебника представляет три различных сообщения, отображаемых во время регистрации. Они восходят своими корнями к неграфическому доступу к ASCII терминалу на многопользовательских UNIX®-системах. Их значение сегодня уменьшается, поскольку многие рабочие станции являются однопользовательскими системами и пользуются такими графическими системами как GNOME или KDE, в которых эта функциональность и вовсе неэффективна.

### /etc/issue and /etc/issue.net

Первые два сообщения, /etc/issue и /etc/issue.net, отображаются на ASCII-терминале, подключенном локально (/etc/issue) или удаленно (/etc/issue.net). В листинге 16 приведены

примеры двух этих файлов, взятых из системы Fedora Core 5.

#### Листинг 16. /etc/issue и /etc/issue.net

```
[ian@attic4 ~]$ cat /etc/issue
Fedora Core release 5 (Bordeaux)
Kernel \r on an \m

[ian@attic4 ~]$ cat /etc/issue.net
Fedora Core release 5 (Bordeaux)
Kernel \r on an \m
[ian@attic4 ~]$
```

Обратите внимание на управляющие последовательности \r и \m. Они позволяют вставлять в сообщение такую информацию как дата или имя системы. Управляющие последовательности приведены в таблице 4; они такие же, которые разрешено использовать для команды mingetty.

Таблица 4. Управляющие последовательности для /etc/issue и /etc/issue.net

Последовательность	Назначение
\d	Вставляет текущий день в соответствии с localtime
\l	Вставляет строку, в которой выполняется mingetty
\m	Вставляет название архитектуры машины (эквивалент uname -m)
\n	Вставляет имя хоста сетевого узла машины (эквивалент uname -n)
\o	Вставляет доменное имя
\r	Вставляет номер редакции операционной системы (эквивалент uname -r)
\t	Вставляет текущее время в соответствии с localtime
\s	Вставляет название операционной системы
\u или \U	Вставляет текущее число вошедших в систему пользователей; \U вставляет "n" пользователей", \u вставляет только "n"
\v	Вставляет версию операционной системы (эквивалент uname -v)

То есть, вы можете увидеть, что примеры, приведенные в листинге 16, вставляют номер

редакции операционной системы и название архитектуры машины. Подключение через telnet к этой системе вызовет появление сообщения /etc/issue.net перед запросом данных о вашей учетной записи, как показано в листинге 17.

#### Листинг 17. Telnet-соединения отображают /etc/issue.net

```
Fedora Core release 5 (Bordeaux)
Kernel 2.6.17-1.2174_FC5 on an x86_64
login: ian
Password:
```

Если бы вы включили в /etc/issue.net немного большее количество управляющих последовательностей, как показано в листинге 18, то ваше сообщение при входе в систему могло бы выглядеть так, как показано в листинге 19.

#### Листинг 18. Измененный /etc/issue.net

```
[ian@attic4 ~]$ cat /etc/issue.net
Fedora Core release 5 (Bordeaux)
Kernel \r on an \m

\n
Date \d
Time \t
```

#### Листинг 19. Новое сообщение при входе в систему через telnet

```
Fedora Core release 5 (Bordeaux)
Kernel 2.6.17-1.2174_FC5 on an x86_64
localhost.localdomain
Date 22:55 on Friday, 15 September 2006
Time 22:55 on Friday, 15 September 2006
```

```
login: ian
Password:
```

Обратите внимание на то, что на этой системе последовательности \d и \t генерируют одинаковый результат. Так получается, что ни \u, ни \U не вставляют число вошедших в систему пользователей. Это, вероятно, отражает тот факт, что эти сообщения очень мало используются в настоящее время. Использование программы telnet с открытой передачей пароля очень не рекомендуется. Поскольку подключение, использующее протокол ssh передает идентификатор пользователя и, следовательно, пропускает запрос о нем, и поскольку реально ASCII-терминалы подключаются удаленно не часто, содержимое /etc/issue.net показывается редко и, возможно, поэтому не достаточно хорошо протестировано.

Вы увидите содержимое /etc/issue в том случае, если не будете использовать графическую регистрацию в системе. Даже если вы используете ее, то обычно сможете перейти в

системную консоль (в текстовый режим), используя комбинации клавиш от Ctrl-Alt-F1 до Ctrl-Alt-F6, а комбинация Ctrl-Alt-F7 возвратит вас в графический терминал.

### Ежедневное сообщение

И /etc/issue, и /etc/issue.net обеспечивают обратную связь с пользователем в форме сообщения при регистрации в системе и могли бы также использоваться для информирования пользователей, например, о предстоящем аварийном отключении. Однако это обычно делается при помощи *ежедневного сообщения* или *motd*, которое хранится в /etc/motd. Содержимое /etc/motd отображается после успешного входа в систему, но сразу перед запуском командного процессора. В листинге 20 приведен пример файла motd, а в листинге 21 показано, как он и /etc/issue.net отображается пользователю, вошедшему в систему через telnet-сессию.

### Листинг 20. Пример ежедневного сообщения (motd)

```
[ian@attic4 ~]$ cat /etc/motd  
PLEASE NOTE!  
  
All systems will shut down this weekend for emergency power testing.  
  
Save your work or lose it.
```

### Листинг 21. Пример ежедневного сообщения (motd)

```
Fedora Core release 5 (Bordeaux)  
Kernel 2.6.17-1.2174_FC5 on an x86_64  
localhost.localdomain  
Date 22:55 on Friday, 15 September 2006  
Time 22:55 on Friday, 15 September 2006
```

```
login: ian  
Password:  
Last login: Fri Sep 15 22:54:18 from 192.168.0.101  
  
PLEASE NOTE!  
  
All systems will shut down this weekend for emergency power testing.  
  
Save your work or lose it.  
[ian@attic4 ~]$
```

Опять же, сообщение motd в реальности полезно только в сессиях на ASCII-терминале. Ни KDE, ни GNOME не имеют простого и удовлетворительного способа его отображения.

Последним уведомительным сообщением, о котором вы должны знать, является команда *wall*, которая посылает предупреждение всем зарегистрировавшимся пользователям, используя текст либо из файла, либо из stdin. И опять же, они не видны пользователям, использующим стандартные настольные системы GNOME или KDE.

## Ресурсы

### Научиться

- Оригинал руководства "[LPI exam 102 prep, Topic 108: Linux documentation](#)".
- Прочтите всю [серию учебных руководств по подготовке к экзаменам в LPI](#) на developerWorks, для того чтобы познакомиться с основами Linux и подготовиться к сертификации на системного администратора.
- На странице [LPIC Program](#) размещены списки заданий, примеры вопросов и подробные цели трех уровней сертификации на системного администратора Linux в Linux Professional Institute.
- В статье "[Элементарные задания для новых Linux-разработчиков](#)" (developerWorks, февраль 2006) описано, как открыть терминальное окно или приглашение командного процессора и многое другое.
- [Linux Documentation Project](#) предоставляет множество полезных документов, в частности, свои HOWTO.
- "[LPI Linux Certification в двух словах, второе издание](#)" (O'Reilly, 2006) и "[Экзамен LPIC I, подготовка 2: Сертификационные экзамены 101 и 102 в Linux Professional Institute \(Exam Cram 2\)](#)" (Que, 2004) являются LPI-справочниками для читателей, предпочитающих книжный формат.
- Дополнительные [руководства для Linux-разработчиков](#) предоставлены в [developerWorks Linux zone](#).
- Следите за [техническими событиями и web-трансляциями на developerWorks](#).

### Получить продукты и технологии

- Загрузите [пробное программное обеспечение IBM](#) непосредственно с developerWorks.

# Учебное пособие для экзамена LPI 102, тема 111: Задачи администрирования

*Администрирование Linux для начинающих (LPIC-1) тема 111*

Ян (Ian) Шилдс (Shields), Senior Programmer, EMC

**Описание:** В этом учебном пособии Ян Шилдс продолжает готовить вас к сдаче экзамена 102 Linux Professional Institute® Администрирование Linux для начинающих (Junior Level Administration, LPIC-1 Exam 102). В этом шестом в серии из девяти пособий Ян знакомит вас с задачами администрирования. Прочтя это пособие, вы узнаете, как управлять пользователями и группами, устанавливать профили пользователей и пользовательские окружения, использовать журналы, планировать задачи, создавать резервные копии данных и поддерживать системное время.

[Больше статей из этой серии](#)

**Дата:** 14.11.2007

**Уровень сложности:** средний

## Прежде чем начать

Узнайте, чему может научить вас это учебное пособие и как извлечь из него максимальную пользу.

## Об этой серии учебных пособий

Linux Professional Institute (LPI) осуществляет сертификацию системных администраторов Linux по трем уровням: *для начинающих* (также называемый "уровень сертификации 1"), *средний уровень* (также называемый "уровень сертификации 2") и *старший уровень* (также называемый "уровень сертификации 3"). Для достижения уровня сертификации 1 вы должны сдать экзамены 101 и 102; для достижения уровня сертификации 2 — экзамены 201 и 202. Для достижения уровня сертификации 3 вы должны иметь действующий сертификат среднего уровня и сдать экзамен 301 ("core"). На старшем уровне вы также можете сдать дополнительные экзамены.

developerWorks предоставляет учебные пособия, которые помогут вам в подготовке к четырем экзаменам для сертификации начального и среднего уровня. Каждый экзамен охватывает несколько тем, и для каждой темы на developerWorks существует соответствующее пособие для самостоятельного изучения. Экзамен LPI 102 содержит девять тем, которым соответствуют учебные пособия от developerWorks:

*Таблица 1. Экзамен LPI 102: Учебные пособия и темы*

Тема экзамена LPI 102	Учебное пособие от developerWorks	Краткое содержание пособия
Тема 105	<a href="#">Подготовка к экзамену LPI 102: Ядро</a>	Установка и сопровождение ядра Linux и его модулей.
Тема 106	<a href="#">Подготовка к экзамену LPI 102: Загрузка, инициализация</a>	Загрузка системы, установка параметров ядра, выключение или перезагрузка системы.

	<u>системы, завершение</u> <u>работы, уровни</u> <u>выполнения</u>	
Тема 107	<u>Подготовка к экзамену</u> <u>LPI 102:</u> <u>Печать</u>	Управление принтерами, очереди печати и задания печати в системе Linux.
Тема 108	<u>Подготовка к экзамену</u> <u>LPI 102:</u> <u>Документация</u>	Использование и управление локальной документацией, поиск документации в Интернете и использование автоматизированных сообщений в процессе регистрации в системе для уведомления пользователей о системных событиях.
Тема 109	<u>Подготовка к экзамену</u> <u>LPI 102:</u> <u>Командные оболочки, написание скриптов, программирование и компиляция</u>	Узнайте, как настраивать окружения shell в соответствии с потребностями пользователя, создавать Bash-функции для часто используемых командных последовательностей, создавать простые новые скрипты, использовать синтаксис оболочки для создания циклов и тестирования, а также настраивать существующие скрипты.
Тема 111	Подготовка к экзамену LPI 102: Задачи администрирования	(Это учебное пособие.) Узнайте, как управлять учетными записями пользователей и групп, настраивать пользовательское и системное окружения, конфигурировать и использовать журналы, автоматизировать задачи системного администрирования, планируя запуск заданий в другое время, создавать резервные копии системы и поддерживать системное время. Подробнее см. <a href="#">цели</a> ниже.
Тема 112	Подготовка к экзамену LPI 102: Основы работы в сети	Скоро ожидается.
Тема 113	Подготовка к экзамену LPI 102: Сетевые сервисы	Скоро ожидается.
Тема 114	Подготовка к экзамену LPI 102: Безопасность	Скоро ожидается.

Чтобы сдать экзамены 101 и 102 (и достичь уровня сертификации 1), вы должны уметь:

- Работать в командной строке Linux
- Выполнять простые операции сопровождения: помогать пользователям в случае затруднений, добавлять пользователей в большие системы, осуществлять резервное копирование и восстановление, а также завершать работу и перезагружать компьютер
- Устанавливать и настраивать рабочую станцию (в том числе систему X Window), подсоединяться к локальной сети (LAN) или подключать отдельно стоящий компьютер к сети Интернет посредством модема

Для продолжения подготовки к сертификации уровня 1 см. [учебные пособия от developerWorks для экзаменов LPI 101 и LPI 102](#), а также [полный набор учебных пособий LPI](#)

[от developerWorks.](#)

Linux Professional Institute не рекомендует использование при подготовке к экзаменам любых учебных материалов или технологий, разработанных третьими лицами. За разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## Об этом учебном пособии

Добро пожаловать в учебное пособие "Задачи администрирования", шестое из девяти пособий, разработанных для подготовки к экзамену LPI 102. Из этого пособия вы узнаете, как управлять пользователями и группами, устанавливать профили пользователей и пользовательские окружения, использовать журналы, планировать задачи, создавать резервные копии данных и поддерживать системное время.

Это учебное пособие организовано в соответствии с целями LPI по этой теме. В общих чертах можно сказать, что темам с большим рейтингом на экзамене посвящено большее число вопросов.

*Таблица 2. Задачи администрирования: цели экзамена, описанные в этом учебном пособии*

Цель экзамена LPI	Рейтинг	Описание цели
1.111.1 <a href="#">Учетные записи пользователей и групп</a>	Рейтинг 4	Добавление, удаление, приостановка и изменение учетных записей пользователей. Управление информацией о пользователях и группах в соответствующих базах данных, включая теневые (shadow) базы данных. Создание и обслуживание учетных записей специального назначения и ограниченных учетных записей.
1.111.2 <a href="#">Настройка пользовательского и системного окружений</a>	Рейтинг 3	Модификация глобальных и пользовательских профилей. Установка переменных окружения и поддержка скелетных каталогов для новых учетных записей пользователей. Установка путей поиска команды.
1.111.3 <a href="#">Конфигурирование и использование системных журналов для удовлетворения потребностей администрации и обеспечения безопасности</a>	Рейтинг 3	Конфигурирование системных журналов и контроль за ними, включая тип и уровень журналируемой информации. Изучение и мониторинг журналов с целью выявления повышенной активности и обнаружения проблем. Ротация и архивирование журналов.
1.111.4 <a href="#">Автоматизация задач системного администрирования путем планирования запуска заданий в будущем</a>	Рейтинг 4	Использование команд <code>cron</code> или <code>anacron</code> для выполнения заданий через регулярные интервалы и использование команды <code>at</code> для выполнения заданий в определенное время.
1.111.5 <a href="#">Поддержка стратегии эффективного резервного копирования данных</a>	Рейтинг 3	Планирование стратегии резервного копирования и автоматическое резервное копирование файловой системы на различные носители.

1.111.6 <a href="#"><u>Поддержка системного времени</u></a>	Рейтинг 4	Поддержка системного времени и часовых поясов, синхронизация часов через NTP. Установка часов в BIOS для правильного отображения времени в UTC и конфигурирование NTP, включая корректировку временного отклонения.
--	--------------	---

### **Необходимые условия**

Чтобы извлечь максимум из этого учебного пособия, вы должны иметь базовые знания о Linux и рабочую версию системы Linux, на которой вы сможете упражняться в выполнении команд, изложенных в этом пособии.

Это пособие предполагает, что вы знакомы с предыдущими учебными пособиями этой серии LPI, так что вы, возможно, захотите сначала ознакомиться с [учебными пособиями для экзамена 101](#). В частности, вам необходимо изучить пособие "[Подготовка к экзамену LPI 101 \(тема 104\) Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard](#)", описывающее основные понятия, связанные с пользователями, группами и правами доступа к файлам.

Формат вывода программы может быть различным в зависимости от ее версии, так что результат вашей работы может выглядеть не совсем так, как это представлено в листингах и на рисунках этого пособия.

## **Учебное пособие для экзамена LPI 102, тема 111: Задачи администрирования**

*Администрирование Linux для начинающих (LPIC-1) тема III*

[Ян \(Ian\) Шилдс \(Shields\)](#), Senior Programmer, EMC

**Описание:** В этом учебном пособии Ян Шилдс продолжает готовить вас к сдаче экзамена 102 Linux Professional Institute® Администрирование Linux для начинающих (Junior Level Administration, LPIC-1 Exam 102). В этом шестом в [серии из девяти пособий](#) Ян знакомит вас с задачами администрирования. Прочтя это пособие, вы узнаете, как управлять пользователями и группами, устанавливать профили пользователей и пользовательские окружения, использовать журналы, планировать задачи, создавать резервные копии данных и поддерживать системное время.

[Больше статей из этой серии](#)

**Дата:** 14.11.2007

**Уровень сложности:** средний

### **Учетные записи пользователей и групп**

Этот раздел охватывает материал по теме 1.111.1 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 4.

Из этого раздела вы узнаете, как:

- Добавлять, изменять и удалять пользователей и группы
- Приостанавливать и изменять учетные записи пользователей
- Управлять информацией о пользователях и группах в базах данных паролей и групп

- Использовать подходящие средства для управления базами данных теневых паролей и базами данных групп
- Создавать ограниченные учетные записи и учетные записи специального назначения и управлять ими

Как вы знаете из учебного пособия "[Подготовка к экзамену LPI 101 \(тема 104\) Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard](#)", Linux является многопользовательской системой, в которой каждый пользователь принадлежит одной *основной* группе и, возможно, некоторым дополнительным группам. В Linux права на файлы тесно связаны с идентификаторами пользователей (*id*) и группами. Вспомните, что можно войти в систему в качестве какого-либо пользователя и при помощи команд `SU` или `sudo -s` стать другим пользователем и что можно воспользоваться командой `whoami` для проверки текущего действительного *id* и командой `groups`, чтобы узнать, какой группе вы принадлежите. Из этого раздела вы узнаете, как создавать и удалять пользователей и группы и управлять ими. Также вы узнаете о содержащихся в каталоге `/etc` файлах, в которых хранится информация о пользователях и группах.

### Добавление и удаление пользователей и групп

Вы добавляете пользователя в систему Linux при помощи команды `useradd` и удаляете при помощи команды `userdel`. Подобным образом, вы добавляете или удаляете группы при помощи команд `groupadd` и `groupdel`.

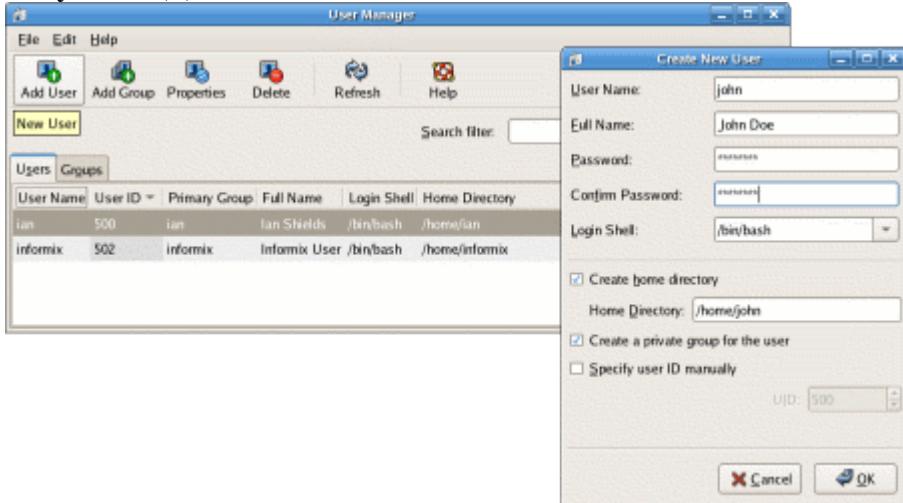
### Добавление пользователя или группы

Администрирование пользователей и групп в современных системах Linux обычно производится при помощи соответствующего графического интерфейса. Обычно доступ к нему можно получить через меню для системного администрирования. Существует большое разнообразие этих интерфейсов, так что интерфейс, присутствующий в вашей системе, может выглядеть не так, как в примере, приведенном здесь, но основные понятия и команды будут похожи.

Давайте начнем с добавления пользователя в графической системе Fedora Core 5 и затем рассмотрим приведенные выше команды. В системе Fedora Core 5 с десктопом GNOME выберите **Система > Администрирование > Пользователи и Группы** и нажмите кнопку **Добавить пользователя**.

На рисунке 1 изображены окно User Manager (Менеджер пользователей) и окно Create New User (Создать пользователя), содержащее основную информацию для нового пользователя по имени 'john'. Были введены полное имя пользователя (Full name) John Doe, и пароль (Password). Оболочка (Login Shell) `/bin/bash` предоставлена по умолчанию. В системах Fedora по умолчанию создается новая группа, имя которой совпадает с именем пользователя, в нашем случае 'john', и домашний каталог `/home/john`.

**Рисунок 1. Добавление пользователя**



В листинге 1 показан пример использования команды [id](#) для просмотра основной информации о новом пользователе. Как вы можете видеть, john имеет идентификатор пользователя 503 и соответствующую группу john с номером 503. john является членом только этой группы.

#### **Листинг 1. Просмотр информации об id пользователя**

```
[root@pinguino ~]# id john
uid=503(john) gid=503(john) groups=503(john)
```

Для выполнения той же задачи из командной строки воспользуйтесь командами [groupadd](#) и [useradd](#) для создания группы и пользователя, а затем командой [passwd](#), чтобы установить пароль для вновь созданного пользователя. Для выполнения всех этих команд необходимы привилегии пользователя root. Основные приемы использования этих команд для добавления другого пользователя, jane, показаны в листинге 2.

#### **Листинг 2. Добавление пользователя jane**

```
[root@pinguino ~]# groupadd jane
[root@pinguino ~]# useradd -c "Jane Doe" -g jane -m jane
[root@pinguino ~]# passwd jane
Changing password for user jane.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@pinguino ~]# id jane
uid=504(jane) gid=504(jane) groups=504(jane)
[root@pinguino ~]# ls -ld /home/jane
drwx----- 3 jane jane 4096 Jun 25 18:22 /home/jane
```

В этих двух примерах и id пользователя, и id группы имеют значение выше 500. В некоторых современных системах пользовательские id начинаются не с 500, а с 1000. Обычно эти значения являются признаком обычных пользователей, в то время как значения ниже 500 (или 1000, если в системе отсчет обычных пользователей начинается с 1000) зарезервированы для *системных пользователей*. Описание [Системных пользователей](#) вы

найдете ниже в этом разделе. Действующие точки отсечения установлены в /etc/login.defs как **UID\_MIN** и **GID\_MIN**.

В листинге 2, приведенном выше, команда **groupadd** имеет один параметр, **jane**, имя добавляемой группы. Имена групп должны начинаться с букв нижнего регистра или знаков подчеркивания и обычно содержат только эти символы, а также дефисы или черточки. Опции, которые можно указать этой команде, показаны в таблице 3.

*Таблица 3. Опции команды groupadd*

Опция	Назначение
-f	Выйти со статусом успешного выполнения, если группа уже существует. Удобна при написании скриптов, когда нет необходимости проверять, существует ли группа, прежде чем пытаться ее создавать.
-g	Задать id группы вручную. По умолчанию используется самое маленькое значение, не меньше, чем <b>GID_MIN</b> , причем больше, чем id любой из существующих групп. id групп обычно уникальны и не должны быть отрицательными
-o	Разрешить группу с неуникальным id.
-K	Может использоваться для отмены значений по умолчанию, хранящихся в файле /etc/login.defs.

В листинге 2, приведенном выше, команда **useradd** имеет один параметр, **jane**, имя добавляемого пользователя, а также опции -**C**, -**G** и -**M**. Наиболее часто употребляемые опции команды **useradd** показаны в таблице 4.

*Таблица 4. Опции команды useradd*

Опция	Назначение
-b --base-dir	Базовый каталог по умолчанию, в котором создаются домашние каталоги пользователей. Обычно это /home, а пользовательские каталоги — /home/\$USER.
-c --comment	Текстовая строка для описания id, содержащая, например, полное имя пользователя.
-d --home	Предоставляет определенное имя каталога для домашнего каталога.
-e --expiredate	Дата, когда учетная запись потеряет силу или будет заблокирована. Задается в формате YYYY-MM_DD.
-g --gid	Имя или номер начальной группы регистрации пользователя. Группа должна существовать, и поэтому в листинге 2 группа <b>jane</b> была создана раньше пользователя <b>jane</b> .
-G --groups	Список дополнительных групп, которым принадлежит пользователь. Группы перечисляются через запятую.
-K	Может использоваться для отмены значений по умолчанию, хранящихся в файле /etc/login.defs.
-m --create-home	Создает домашний каталог пользователя, если он не существует. Копирует скелетные файлы и другие каталоги

	из /etc/skel в домашний каталог.
-o --non-unique	Позволяет создать пользователя с неуникальным id.
-p --password	Шифрованный пароль. Если пароль не определен, по умолчанию учетная запись заблокирована. Вместо того чтобы создавать шифрованный пароль и определять его в команде <code>useradd</code> , обычно для создания пароля вы будете использовать команду <code>passwd</code> .
-s --shell	Имя login shell пользователя, если оно отличается от login shell по умолчанию.
-u --uid	Неотрицательное цифровое значение id пользователя, которое должно быть уникальным, если не определено иначе опцией -o. По умолчанию используется самое маленькое значение, не меньше, чем <code>UID_MIN</code> , причем больше, чем id любого из существующих пользователей.

### Примечания:

1. В некоторых системах, в том числе в дистрибутивах Fedora и Red Hat, имеются расширения в виде команд для создания пользователей. Например, в системах Fedora и Red Hat по умолчанию для пользователя создается новая группа, и для запрета этой функции при выполнении команды `useradd` используется опция `-n`. Следует знать, что такие различия возможны, и при возникновении сомнений обращаться к страницам руководства `man` вашей системы.
2. В системах SUSE для доступа к графическому интерфейсу администрирования пользователей и групп используется YaST или YaST2.
3. Графические интерфейсы могут использоваться для решения дополнительных задач, например, для создания файла для почты пользователя в каталоге `/var/spool/mail`.

### Удаление пользователя или группы

Удаление пользователя или группы значительно проще, чем их создание, поскольку имеет меньше опций. Фактически, команде `groupdel` для удаления группы требуется только имя группы; эта команда не имеет опций. Вы не можете удалить группу, если она является основной группой пользователя. Если для удаления пользователей и групп вы используете графический интерфейс, действия очень похожи на команды, показанные выше.

Воспользуйтесь командой `userdel`, чтобы удалить пользователя. Опция `-r` или `--remove` дает указание удалить домашний каталог пользователя и все его содержимое вместе с пользовательской почтой. Когда вы удаляете пользователя, группа, имеющая то же имя, что и пользователь, также удаляется, если переменная `USERGROUPS_ENAB` в файле `/etc/login.defs` установлена в положение yes, но это будет сделано, только если группа не является основной для другого пользователя.

В листинге 3 вы видите пример удаления группы, когда несколько пользователей разделяют одну и ту же основную группу. Здесь другой пользователь, `jane2`, был предварительно добавлен в систему с той же группой, что и `jane`.

### Листинг 3. Удаление пользователей и групп

```
root@pinguino:~# groupdel jane
groupdel: cannot remove user's primary group.
root@pinguino:~# userdel -r jane
userdel: Cannot remove group jane which is a primary group for another user.
```

```
root@pinguino:~# userdel -r jane2
root@pinguino:~# groupdel jane
```

## Примечания:

1. Для удаления пользователей и их групп команда **userdel** может быть использована с опцией **-f** или **--force**. Эта опция опасна, поэтому использовать ее следует только в крайнем случае. Прежде чем сделать это, внимательно прочтите руководство *man*.
2. Следует знать, что если вы удаляете пользователя или группу и в файловой системе есть файлы, принадлежащие этому пользователю или группе, эти файлы автоматически не удаляются или присваиваются другому пользователю или группе.

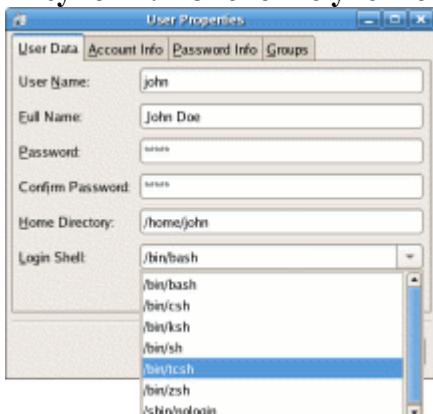
## Приостановка и изменение учетных записей

Теперь, когда вы можете создать или удалить id пользователя или группу, у вас может возникнуть потребность изменить их.

### Изменение учетных записей пользователей

Предположим, пользователь john хочет иметь в качестве оболочки по умолчанию tcsh. В графическом интерфейсе вы, как правило, найдете способ отредактировать данные о пользователе (или группе) или просмотреть свойства объекта. На рисунке 2 показан диалог Свойства пользователя (User Properties) для пользователя john, созданного нами ранее в системе Fedora Core 5.

**Рисунок 2. Изменение учетной записи пользователя**



Для изменения учетной записи пользователя из командной строки используйте команду **usermod**. Можно использовать большинство опций, которые используются с командой **useradd**, за исключением того, что для пользователя нельзя создать или наполнить содержимым новый домашний каталог. Если необходимо изменить имя пользователя, используйте опцию **-l** или **--login** в сочетании с новым именем. Возможно, вы захотите переименовать домашний каталог, чтобы он соответствовал id пользователя. У вас также может возникнуть необходимость переименовать другие элементы, такие как почтовые spool-файлы. Наконец, если login shell изменен, может возникнуть необходимость изменить некоторые связанные с ним profile-файлы. В листинге 4 показан пример операций, которые необходимо выполнить, чтобы изменить пользователя john на john2 с /bin/tcsh в качестве shell по умолчанию и переименовать домашний каталог на /home/john2.

## Листинг 4. Изменение пользователя

```
[root@pinguino ~]# usermod -l john2 -s /bin/tcsh -d /home/john2 john
[root@pinguino ~]# ls -d ~john2
```

```
ls: /home/john2: No such file or directory
[root@pinguino ~]# mv /home/john /home/john2
[root@pinguino ~]# ls -d ~john2
/home/john2
```

### Примечания:

1. Если вам необходимо изменить дополнительные группы пользователя, вы должны определить полный список дополнительных групп. Не существует команды, чтобы просто добавить или удалить единственную группу для пользователя.
2. Существуют ограничения на изменение имени или id для пользователя, который зарегистрирован в системе или который выполняет какие-либо процессы. За подробностями обратитесь к страницам руководства man.
3. Если вы меняете номер пользователя, у вас может возникнуть желание изменить владельца файлов и каталогов этого пользователя в соответствии с новым номером.

### Изменение групп

Ни для кого не является сюрпризом, что команда **groupmod** используется для изменения информации о группе. При помощи опции **groupmod** можно изменить номер, а при помощи опции **-n — имя группы**.

### Листинг 5. Переименование группы

```
[root@pinguino ~]# ls -ld ~john2
drwx----- 3 john2 john 4096 Jun 26 18:29 /home/john2
[root@pinguino ~]# groupmod -n john2 john
[root@pinguino ~]# ls -ld ~john2
drwx----- 3 john2 john2 4096 Jun 26 18:29 /home/john2
```

В листинге 5 обратите внимание, что, когда мы использовали команду **groupmod** для изменения имени группы, имя группы для домашнего каталога пользователя john2 чудесным образом изменилось. Вы удивлены? Это не удивительно, поскольку в файловой системе группы представлены их номерами, а не именами. Однако, если вы изменяете номер группы, вы должны обновить всех пользователей, для которых эта группа является основной, кроме того у вас может возникнуть желание обновить файлы и каталоги, принадлежащие этой группе, в соответствии с новым номером (так же, как было сказано выше, где речь шла об изменении номера пользователя). В листинге 6 показано, как изменить номер группы для john2 на 505, обновить учетную запись пользователя и произвести соответствующие изменения для всех файлов, входящих в файловую систему /home. Вы, вероятно, захотите изменить номера пользователей и групп, если это вообще возможно.

### Листинг 6. Переименование группы

```
[root@pinguino ~]# groupmod -g 505 john2
[root@pinguino ~]# ls -ld ~john2
drwx----- 3 john2 503 4096 Jun 26 18:29 /home/john2
[root@pinguino ~]# id john2
uid=503(john2) gid=503 groups=503
[root@pinguino ~]# usermod -g john2 john2
[root@pinguino ~]# id john2
uid=503(john2) gid=505(john2) groups=505(john2)
[root@pinguino ~]# ls -ld ~john2
```

```
drwx----- 3 john2 503 4096 Jun 26 18:29 /home/john2
[root@pinguino ~]# find /home -gid 503 -exec chgrp john2 {} \;
[root@pinguino ~]# ls -ld ~john2
drwx----- 3 john2 john2 4096 Jun 26 18:29 /home/john2
```

## Пароли пользователей и групп

Вам уже встречалась команда `passwd`, которая используется для изменения пароля пользователя. Пароль является уникальным (или должен быть таковым) для пользователя и может быть изменен пользователем. Как вы уже видели, пользователь `root` может изменить пароль любого пользователя.

Группы также могут иметь пароли, и для их установки используется команда `gpasswd`. Наличие пароля группы позволяет пользователям временно войти в группу при помощи команды `newgrp`, если им известен пароль группы. Конечно, наличие пароля, известного нескольким пользователям в некоторой степени проблематично, поэтому необходимо оценить преимущества добавления пользователя в группу при помощи команды `usermod` в сравнении с проблемой безопасности при слишком большом количестве людей, знающих пароль группы.

## Приостановка и блокирование учетных записей

Если необходимо запретить пользователю регистрацию в системе, можно *приостановить* или *заблокировать* учетную запись при помощи команды `usermod` с опцией `-L`. Для разблокирования учетной записи используется опция `-U`. В листинге 7 показано, как заблокировать учетную запись `john2`, и что произойдет, если `john2` попытается зарегистрироваться в системе. Обратите внимание, что когда учетная запись `john2` разблокируется, восстанавливается ее прежний пароль.

## Листинг 7. Блокирование учетной записи

```
[root@pinguino ~]# usermod -L john2
[root@pinguino ~]# ssh john2@pinguino
john2@pinguino's password:
Permission denied, please try again.
```

Вы, возможно, заметили ранее, что окно диалога на [рисунке 2](#) имело несколько вкладок с дополнительными свойствами пользователя. Мы кратко упомянули о возможности использования команды `passwd` для установки пароля пользователя, но и эта команда, и команды `usermod` и `chage` могут выполнять множество задач, связанных с учетными записями пользователей. Некоторые их опции показаны в таблице 5. За более подробной информацией об этих и других опциях обратитесь к соответствующим страницам руководства `man`.

Таблица 5. Команды и опции для изменения учетных записей пользователей

Опция команды	Назначение
---------------	------------

**Usermod Passwd Chage**

-L	-l	N/A	Блокирует или приостанавливает действие учетной записи.
-U	-u	N/A	Разблокирует учетную запись.

N/A	-d	N/A	Блокирует учетную запись путем отмены ее пароля.
-e	-f	-E	Устанавливает дату прекращения полномочий для учетной записи.
N/A	-n	-m	Минимальное время действия пароля в днях.
N/A	-x	-M	Максимальное время действия пароля в днях.
N/A	-w	-W	Число дней, за которое появляется предупреждение о необходимости изменить пароль.
-f	-i	-I	Число дней после того, как пароль потеряет силу, но до того, как учетная запись будет отключена.
N/A	-S	-l	Вывод краткого сообщения о статусе текущей учетной записи.

## Управление базами данных пользователей и групп

Основные репозитории, содержащие информацию о пользователях и группах, — это четыре файла в каталоге /etc:

### /etc/passwd

файл *паролей*, содержащий основную информацию о пользователях

### /etc/shadow

файл *теневых паролей*, содержащий шифрованные пароли

### /etc/group

файл *групп*, содержащий основную информацию о группах и принадлежащих этим группам пользователях

### /etc/gshadow

файл *теневых групп*, содержащий шифрованные пароли групп

Эти файлы обновляются при помощи команд, которые вы уже видели в этом учебном пособии, кроме того, после того как мы обсудим сами эти файлы, вам встретятся другие команды для работы с ними. Все эти файлы являются простыми текстовыми файлами. Вообще, вы не должны редактировать их непосредственно. Для их обновления используются специальные инструменты, так что они должным образом блокируются и поддерживаются в синхронном состоянии.

Обратите внимание, что файлы passwd и group являются *затеняемыми*. Это сделано из соображений безопасности. Сами файлы passwd и group должны быть доступными для чтения для всех, а зашифрованные пароли — недоступными для чтения для всех. Поэтому зашифрованные пароли хранятся в теневых файлах, и эти файлы доступны для чтения только пользователю root. Необходимый доступ для изменения аутентификационных данных обеспечивается при помощи *suid*-программы, которая имеет полномочия пользователя root, но может быть запущена любым пользователем. Убедитесь, что в системе установлены соответствующие права доступа. В листинге 8 показан пример.

## Листинг 8. Права доступа к базам данных пользователей и групп

```
[ian@pinguino ~]$ ls -l /etc/passwd /etc/shadow /etc/group /etc/gshadow
-rw-r--r-- 1 root root 701 Jun 26 19:04 /etc/group
```

```
-r----- 1 root root 580 Jun 26 19:04 /etc/gshadow  
-rw-r--r-- 1 root root 1939 Jun 26 19:43 /etc/passwd  
-r----- 1 root root 1324 Jun 26 19:50 /etc/shadow
```

**Примечание:** Несмотря на то, что все еще существует техническая возможность работы без теневых файлов паролей и групп, эта возможность почти никогда не используется и пользоваться ею не рекомендуется.

### Файл /etc/passwd

Файл /etc/passwd содержит одну строку для каждого пользователя системы. В листинге 9 показано несколько примеров строк.

#### Листинг 9. Записи из файла /etc/password

```
root:x:0:0:root:/bin/bash  
jane:x:504:504:Jane Doe:/home/jane:/bin/bash  
john2:x:503:505:John Doe:/home/john2:/bin/tcsh
```

Каждая строка содержит семь полей, разделенных двоеточиями (:), как показано в таблице 6.

Таблица 6. Поля файла /etc/passwd

Поле	Назначение
Имя пользователя (Username)	Имя, используемое для входа в систему. Например, john2.
Пароль (Password)	Зашифрованный пароль. Если используется зашифрованный пароль, это поле содержит единичный символ x.
id пользователя (UID)	Число, используемое для представления этого пользователя в системе. Например, 503 для пользователя john2.
id группы (GID)	Число, используемое для представления этой основной группы пользователя в системе. Например, 505 для пользователя john2.
Комментарий (GECOS)	Необязательное поле, используемое для описания пользователя. Например, "John Doe". Это поле может содержать несколько разделенных запятыми записей. Оно также используется такой программой как finger. Название поля GECOS сложилось исторически. Подробнее см. в man 5 passwd.
Домашний каталог (Home)	Абсолютный путь для домашнего каталога пользователя. Например, /home/john2.
Командная оболочка (Shell)	Программа, которая автоматически запускается при входе пользователя в систему. Обычно это интерактивный shell, такой как /bin/bash или /bin/tcsh, но это может быть и другая программа, не обязательно

интерактивный shell.

## Файл /etc/group

Файл /etc/group содержит одну строку для каждой группы системы. В листинге 10 показано несколько примеров строк.

### Листинг 10. Записи в /etc/group

```
root:x:0:root
jane:x:504:john2
john2:x:505:
```

Каждая строка содержит четыре поля, разделенных двоеточиями (:), как показано в таблице 7.

Таблица 7. Поля файла /etc/group

Поле	Назначение
Имя группы (Groupname)	Имя этой группы. Например, john2.
Пароль (Password)	Зашифрованный пароль. Если используется зашифрованный пароль группы, это поле содержит единичный символ x.
id группы (GID)	Число, используемое для представления этой группы в системе. Например, 505 для группы john2.
Члены (Members)	Разделенный запятыми список членов группы, за исключением тех членов, для которых эта группа является основной.

## Теневые файлы

Файл /etc/shadow должен быть доступен для чтения только для пользователя root. Он содержит зашифрованные пароли наряду с паролем и информацией о времени истечения действия учетной записи. Информацию о значении полей см. в man-странице ([man 5 shadow](#)). Пароли могут быть зашифрованы при помощи DES, но чаще для шифрования используется MD5. Алгоритм DES использует 7 младших битов из первых 8 символов пароля пользователя, представленных в виде 56-битного ключа, в то время как алгоритм MD5 использует весь пароль. В любом случае пароли кодированы при помощи *salt-кода*, так что из двух идентичных паролей не будут генерированы одинаковые зашифрованные значения. В листинге 11 показано, как установить одинаковые пароли для пользователей jane и john2, и затем показан результат шифрования паролей при помощи MD5 в файле /etc/shadow.

### Листинг 11. Пароли в /etc/shadow

```
[root@pinguino ~]# echo lpic1111 |passwd jane --stdin
Changing password for user jane.
passwd: all authentication tokens updated successfully.
[root@pinguino ~]# echo lpic1111 |passwd john2 --stdin
Changing password for user john2.
passwd: all authentication tokens updated successfully.
[root@pinguino ~]# grep "^\$" /etc/shadow
```

```
jane:$1$eG0/KGQY$ZJl.ltYtVw0sv.C50rqUu/:13691:0:99999:7:::  
john2:$1$grkxobie$J2muvoTpwo3dZAYYTDYNu.:13691:0:180:7:29::
```

Лидирующий **\$1\$** означает пароль MD5, а salt — это поле переменной длины до 8 символов, заканчивающееся следующим символом **\$**. Оставшаяся строка из 22 символов — это зашифрованный пароль.

## [В начало](#)

Средства администрирования пользователей и групп

Вам уже встречалось несколько команд для манипуляций с файлами учетных записей и групп и их теневыми файлами. Сейчас вы узнаете о:

- Администраторах групп
- Командах редактирования файлов паролей и групп
- Программах преобразования

### Администраторы группы

При каких-то обстоятельствах у вас может возникнуть желание, чтобы не только root, но и другие пользователи могли администрировать одну или несколько групп, добавляя или удаляя членов группы. В листинге 12 показано, как root может добавить пользователя jane в качестве администратора для группы john2 и затем jane, в свою очередь, может добавить пользователя ian в качестве пользователя.

### Листинг 12. Добавление администраторов и членов группы

```
[root@pinguino ~]# gpasswd -A jane john2  
[root@pinguino ~]# su - jane  
[jane@pinguino ~]$ gpasswd -a ian john2  
Adding user ian to group john2  
[jane@pinguino ~]$ id ian;id jane  
uid=500(ian) gid=500(ian) groups=500(ian),505(john2)  
uid=504(jane) gid=504(jane) groups=504(jane)
```

Вы можете с удивлением заметить, что, несмотря на то, что jane является администратором группы john2, она не является ее членом. Исследование структуры файла /etc/gshadow показывает, почему это произошло. Как показано в таблице 8, файл /etc/gshadow содержит четыре поля для каждой записи. Обратите внимание, что третье поле — это разделенный запятыми список администраторов группы.

*Таблица 8. Поля файла /etc/gshadow*

Поле	Назначение
Имя группы (Groupname)	Имя этой группы. Например, john2.
Пароль (Password)	Поле используется для хранения зашифрованного пароля, если у группы имеется пароль. Если группа не имеет пароля, здесь можно увидеть 'x', '!' или '!!!'.
Администраторы (Admins)	Разделенный запятыми список администраторов группы.

**Члены (Members)**      Разделенный запятыми список членов группы.

Как можно заметить, список администраторов и список членов — это два отдельных поля. Опция **-A** команды **gpasswd** позволяет пользователю root добавлять администраторов группы, в то время как опция **-M** позволяет пользователю root добавлять членов. Опция **-a** (заметьте, что используется нижний регистр) позволяет администратору добавлять члена, в то время как опция **-d** позволяет администратору удалять пользователя. Дополнительные опции позволяют удалить пароль группы. Подробнее см. в страницах руководства man.

### **Команды редактирования файлов паролей и групп**

Хотя следующих двух команд нет в списке целей LPI, необходимо знать, что при помощи команды **vipw** можно безопасно редактировать файл /etc/passwd, а при помощи команды **vigr** безопасно редактировать файл /etc/group. Эти команды заблокируют необходимые файлы на то время, пока при помощи редактора **vi** будут производиться изменения. Если вы вносите изменения в файл /etc/passwd, команда **vipw** подскажет, что необходимо проверить, не нужно ли обновить и файл /etc/shadow. Подобным образом, если вы обновляете файл /etc/group при помощи команды **vigr**, вы получите подсказку, что необходимо обновить и файл /etc/gshadow. Если необходимо удалить администраторов группы, необходимо использовать команду **vigr**, поскольку команда **gpasswd** позволяет только добавлять администраторов.

### **Программы преобразования**

Другие четыре другие связанные команды также не перечислены в целях LPI. Это команды **pwconv**, **pwunconv**, **grconv** и **grunconv**. Они используются для преобразования файлов теневых паролей и групп в нетеневые и обратно. У вас может никогда не возникнуть необходимости в этих командах, но вы должны знать об их существовании. Подробности см. в страницах руководства man.

### **Ограниченные учетные записи и учетные записи специального назначения**

В соответствии с соглашением, системные пользователи обычно имеют id меньше, чем 100, а пользователь root имеет id, равный 0. Автоматическая нумерация обычных пользователей начинается со значения **UID\_MIN**, установленного в файле /etc/login.defs, это значение обычно установлено в 500 или 1000.

Помимо учетных записей обычных пользователей и учетной записи пользователя root, обычно в системе бывает несколько учетных записей специального назначения для демонов, таких как FTP, SSH, mail, news и т.д. В листинге 13 показано несколько записей из файла /etc/passwd для этих учетных записей.

### **Листинг 13. Ограниченные учетные записи и учетные записи специального назначения**

```
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
```

Такие учетные записи часто управляют файлами, но к ним невозможно получить доступ путем обычной регистрации в системе. Поэтому обычно они имеют login shell, определенный как /sbin/nologin или /bin/false, чтобы попытки зарегистрироваться в системе терпели неудачу.

## Настройка окружения

Этот раздел охватывает материал по теме 1.111.2 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 3.

Из этого раздела вы узнаете, как настроить пользовательское окружение, включая решение следующих задач:

- Установка переменных окружения и отмена установок
- Поддержка скелетных каталогов для новых учетных записей пользователей
- Установка путей поиска команды

### Установка переменных окружения и отмена установок

При создании нового пользователя вы обычно устанавливаете множество переменных в соответствии с вашими частными потребностями. Эти переменные обычно устанавливаются в предоставляемых новым пользователям профайлах, таких как .bash\_profile и .bashrc, или в общесистемных профайлах /etc/profile и /etc/bashrc. В листинге 14 показан пример, как установить системное приглашение PS1 в /etc/profile на системе Ubuntu 7.04. Первый оператор `if` проверяет, установлена ли переменная PS1, что показывает, что это интерактивный shell, поскольку для неинтерактивного shell приглашение не требуется. Второй оператор `if` проверяет, установлена ли переменная окружения BASH. Если да, устанавливается приглашение и /etc/bash.bashrc (обратите внимание на точку). Если переменная BASH не установлена, проверяется, запущена ли она от имени root (`id=0`), и устанавливается приглашение # или \$ соответственно.

### Листинг 14. Установка переменных окружения

```
if [ "$PS1" ]; then
    if [ "$BASH" ]; then
        PS1='\u@\h:\w\$ '
        if [ -f /etc/bash.bashrc ]; then
            . /etc/bash.bashrc
        fi
    else
        if [ "`id -u`" -eq 0 ]; then
            PS1='# '
        else
            PS1='$ '
        fi
    fi
fi
```

В учебном пособии [Подготовка к экзамену LPI 102: Командные оболочки, написание скриптов, программирование и компиляция \(LPI exam 102 prep: Shells, scripting, programming, and compiling\)](#) дается подробная информация о командах, используемых для установки переменных окружения и отмены установок, а также информация о том, как и когда используются различные профайлы.

Настраивая пользовательские окружения, следует учитывать два важных момента:

1. Чтение файла /etc/profile происходит только во время регистрации в системе и не происходит при запуске каждого нового shell'a.
2. Функции и псевдонимы не наследуются новыми shell'ами. Поэтому обычно вы будете устанавливать их и ваши переменные окружения в /etc/bashrc или в собственный

профайл пользователя.

Linux Standard Base (LSB) предусматривает, что дополнительные скрипты могут быть расположены не только в системных профайлах /etc/profile и /etc/bashrc, но и в каталоге /etc/profile.d. Эти скрипты служат источником при создании интерактивного login shell. Они обеспечивают удобный способ разделения настроек для различных программ. В листинге 15 показан пример.

### Листинг 15. Файл /etc/profile.d/vim.sh из Fedora 7

```
[if [ -n "$BASH_VERSION" -o -n "$KSH_VERSION" -o -n "$ZSH_VERSION" ]; then
  [ -x //usr/bin/id ] || return
  [ `//usr/bin/id -u` -le 100 ] && return
  # for bash and zsh, only if no alias is already set
  alias vi >/dev/null 2>&1 || alias vi=vim
fi
```

Помните, что обычно вы должны **экспортировать** все переменные, установленные в профайле; иначе они не будут доступны командам, запускаемым в новом shell'e.

### Поддержка скелетных каталогов для новых учетных записей пользователей

Из раздела [Добавление и удаление пользователей и групп](#) вы узнали, как можно создать или наполнить содержимым новый домашний каталог пользователя. Источником для этого нового каталога служит поддерево, корнем которого является /etc/skel. В листинге 16 показаны файлы этого поддерева для системы Fedora 7. Обратите внимание, что большинство файлов начинается с точки, поэтому для их просмотра необходимо использовать опцию **-a**. Опция **-R** рекурсивно выводит подкаталоги, а опция **-L** — соответствующие символьные ссылки.

### Листинг 16. Файл /etc/skel из Fedora 7

```
[ian@lyrebird ~]$ ls -aRL /etc/skel
/etc/skel:
. .. .bash_logout .bash_profile .bashrc .emacs .xemacs

/etc/skel/.xemacs:
. .. init.el
```

Обратите внимание, что в добавок к файлам .bash\_logout, .bash\_profile и .bashrc, которые вы могли ожидать увидеть для Bash shell, этот пример содержит информацию о профайле для редакторов emacs и xemacs. Если вам необходима информация о функциях различных profile-файлов, обратитесь к учебному пособию [Подготовка к экзамену LPI 102: Командные оболочки, написание скриптов, программирование и компиляция \(LPI exam 102 prep: Shells, scripting, programming, and compiling\)](#).

В листинге 17 показан файл /etc/skel/.bashrc для системы Fedora 7. В разных релизах и разных дистрибутивах этот файл может быть различным, но он дает представление о том, какие пользовательские установки по умолчанию можно сделать.

## Листинг 17. Файл /etc/skel/.bashrc из Fedora 7

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions
```

Как можно видеть, источником является глобальный /etc/bashrc, затем могут быть добавлены любые специфичные для пользователя инструкции. В листинге 18 показан фрагмент файла /etc/bashrc, в котором скрипты .sh подгружаются из /etc/profile.d.

## Листинг 18. Подгрузка скриптов .sh из /etc/profile.d

```
for i in /etc/profile.d/*.sh; do
    if [ -r "$i" ]; then
        . $i
    fi
done
unset i
```

Обратите внимание, что после выполнения цикла установки для переменной *i* отменены.

### Установка путей поиска команды

Ваши профайлы по умолчанию часто содержат переменные PATH для частных функций или для продуктов, которые вы могли установить. Можно включить их в скелетные файлы /etc/skel, изменить /etc/profile, /etc/bashrc или создать файл /etc/profile.d, если он используется в вашей системе. Если вы изменяете системные файлы, убедитесь, что ваши изменения сохранятся после любых обновлений системы. В листинге 19 показано, как добавить новый каталог /opt/productxyz/bin в начало или конец существующего PATH.

## Листинг 19. Добавление каталога в путь поиска

```
PATH="$PATH${PATH:+:}/opt/productxyz/bin"
PATH="/opt/productxyz/bin${PATH:+:}$PATH"
```

Хотя строгого требования не существует, выражение \${PATH:+:} вставляет в путь разделитель (двоеточие), только если переменная PATH не установлена или равна нулю.

### Системные журналы

Этот раздел охватывает материал по теме 1.111.3 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 3.

Из этого раздела вы узнаете, как конфигурировать и использовать системные журналы, включая решение следующих задач:

- Управление типом и уровнем журналируемой информации

- Автоматическая ротация и архивирование журналов
- Просмотр журналов с целью выявления повышенной активности
- Мониторинг журналов
- Обнаружение в журналах сообщений о проблемах

## Управление типом и уровнем журналируемой информации

Функция системного журналирования в системе Linux обеспечивает системное журналирование и перехват сообщений ядра. Журналирование может осуществляться на локальной системе или пересыпаться на удаленную систему, кроме того, в конфигурационном файле /etc/syslog.conf возможна тонкая регулировка уровня журналирования. Журналирование осуществляется при помощи демона **syslogd**, который обычно получает входную информацию при помощи сокета /dev/log, как показано в листинге 20.

### Листинг 20. Сокет /dev/log

```
ian@pinguino:~$ ls -l /dev/log
srw-rw-rw- 1 root root 0 2007-07-05 15:42 /dev/log
```

В случае локального журналирования главным файлом обычно является /var/log/messages, но в большинстве инсталляций используются и многие другие файлы, которые могут быть тщательно настроены. Например, у вас может возникнуть желание выделить сообщения, порождаемые системой электронной почты.

### Конфигурационный файл syslog.conf

Файл syslog.conf является главным конфигурационным файлом для демона syslogd. Журналирование базируется на сочетании facility (категория) и priority (приоритет). Существуют следующие категории: auth (или security), authpriv, cron, daemon, ftp, kern, lpr, mail, mark, news, syslog, user, uucp, а также с local0 по local7. Ключевое слово **auth** должно использоваться вместо **security**, а ключевое слово **mark** предназначено для внутреннего использования.

Приоритеты (в порядке возрастания):

1. debug
2. info
3. notice
4. warning (или warn)
5. err (или error)
6. crit
7. alert
8. emerg (или panic)

Ключевые слова, помещенные в скобки (warn, error и panic), сейчас признаны устаревшими.

Правила журналирования определяются записями в syslog.conf. Каждое правило имеет поле селектор и поле действие, которые разделены одним или более пробелами или символами табуляции. Поле селектор устанавливает категории и приоритеты, которые используют правило, а поле действие устанавливает журналируемое действие для категорий и приоритетов. По умолчанию выбирается действие для определенного уровня и для всех более высоких уровней, хотя можно ограничить журналирование определенным уровнем. Каждый селектор состоит из категории и приоритета, разделенных точкой. Для данного действия могут быть определены несколько категорий, разделенных запятыми. Для данного

действия могут быть определены несколько пар категория/приоритет, разделенных точкой с запятой. В листинге 21 показан пример несложного syslog.conf.

### Листинг 21. Пример syslog.conf

```
# Log all kernel messages to the console.  
# Logging much else clutters up the screen.  
#kern.*                                              /dev/console  
  
# Log anything (except mail) of level info or higher.  
# Don't log private authentication messages!  
*.info;mail.none;authpriv.none;cron.none           /var/log/messages  
  
# The authpriv file has restricted access.  
authpriv.*                                            /var/log/secure  
  
# Log all the mail messages in one place.  
mail.*                                                 -/var/log/maillog  
  
# Log cron stuff  
cron.*                                                /var/log/cron  
  
# Everybody gets emergency messages  
*.emerg                                              *  
  
# Save news errors of level crit and higher in a special file.  
uucp,news.crit                                         /var/log/spooler  
  
# Save boot messages also to boot.log  
local7.*                                              /var/log/boot.log
```

### Примечания:

- Как и во многих конфигурационных файлах, строки, начинающиеся с #, и пустые строки игнорируются.
- Символ \* может использоваться для указания всех категорий или всех приоритетов.
- Специальное ключевое слово none указывает, что журналирование для этой категории не должно быть выполнено для этого действия.
- Дефис перед именем файла (как -/var/log/maillog в этом примере) указывает, что после каждой записи журнал не должен синхронизироваться. В случае аварии системы вы можете потерять информацию, но отключение синхронизации позволит повысить производительность.

В общем, действия упоминаются как "log-файлы", хотя они и не должны действительно быть файлами. В таблице 9 дается описание возможных типов log-файлов.

Таблица 9. Действия в syslog.conf

Действие	Назначение
Обычный файл	Задайте полное имя пути, начиная со слеша (/). Поставьте перед ним дефис (-), чтобы отменить синхронизацию файла после каждой записи. Это может привести к потере информации, но повысить производительность.
Именованные каналы	Размещение перед именем файла символа канала

(|) позволит использовать fifo (first in — first out, первый пришел — первый вышел) или именованный канал (named pipe) в качестве приемника для сообщений. Прежде чем запускать (или перезапускать) `syslogd`, необходимо создать fifo при помощи команды `mkfifo`. Иногда fifo используются для отладки.

Терминал и консоль

Удаленная машина

Список пользователей

Все зарегистрированные пользователи

Терминал, такой как `/dev/console`.

Чтобы сообщения пересыпались на другой хост, поместите перед именем хоста символ (@). Обратите внимание, что сообщения не пересыпаются с принимающего хоста.

Разделенный запятыми список пользователей, получающих сообщения (если пользователь зарегистрирован в системе). Сюда часто включается пользователь `root`.

Чтобы известить всех зарегистрированных пользователей при помощи команды `wall`, используйте символ звездочки (\*).

Можно поставить перед приоритетом знак !, чтобы показать, что действие не должно применяться, начиная с этого уровня и выше. Подобным образом, перед приоритетом можно поставить знак =, чтобы показать, что правило применяется только к этому уровню, или !=, чтобы показать, что правило применяется ко всем уровням, кроме этого. В листинге 22 показано несколько примеров, а страница руководства man для `syslog.conf` содержит множество других примеров.

## Листинг 22. Другие примеры `syslog.conf`

```
# Store all kernel messages in /var/log/kernel.
# Send critical and higher ones to remote host pinguino and to the console
# Finally, Send info, notice and warning messages to /var/log/kernel-info
#
kern.*                      /var/log/kernel
kern.crit                    @pinguino
kern.crit                    /dev/console
kern.info;kern.!err          /var/log/kernel-info

# Store all mail messages except info priority in /var/log/mail.
mail.*;mail.!=info           /var/log/mail
```

## Автоматическая ротация и архивирование журналов

При всем многообразии журналов вы должны иметь возможность контролировать их размер. Это делается при помощи команды `logrotate`, которая обычно выполняется демоном cron. Работа демона cron описана в этом пособии ниже в разделе [Планирование задач](#). Главная цель команды `logrotate` состоит в том, чтобы периодически создавать резервные копии журналов и начинать новые журналы. Сохраняется несколько поколений журналов, и, когда завершается срок жизни журнала последнего поколения, он может быть заархивирован. Результат может быть отправлен по почте, например, ответственному за ведение архивов.

Для определения порядка ротации и архивирования журналов используется

конфигурационный файл /etc/logrotate.conf. Для разных журналов можно задать разную периодичность, например, ежедневно, еженедельно или ежемесячно, кроме того, можно регулировать количество накапливаемых поколений, а также указать, будут ли копии архивов отправляться ответственному за ведение архивов и, если будут, когда. В листинге 23 показан пример файла /etc/logrotate.conf.

### Листинг 23. Пример файла /etc/logrotate.conf

```
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

# system-specific logs may be configured here
```

Глобальные опции размещаются в начале файла logrotate.conf. Они используются по умолчанию, если где-то в другом месте не задано ничего более определенного. В нашем примере ротация журналов происходит еженедельно и резервные копии сохраняются в течение четырех недель. Как только производится ротация журнала, на месте старого журнала автоматически создается новый. Обратите внимание, что файл logrotate.conf может содержать спецификации из других файлов. Так, в него включаются все файлы из /etc/logrotate.d.

В этом примере также содержатся специальные правила для /var/log/wtmp и /var/log/btmp, ротация которых происходит ежемесячно. Если файлы отсутствуют, сообщение об ошибке не выдается. Создается новый файл и сохраняется только одна резервная копия.

В этом примере по достижении резервной копией последнего поколения она удаляется, поскольку не определено, что следует с ней делать.

**Примечание:** В файлы /var/log/wtmp и /var/log/btmp записываются удачные и неудачные попытки регистрации в системе соответственно. В отличие от большинства журналов, эти

файлы не являются чисто текстовыми. Просмотреть их содержимое можно при помощи команд `last` или `lastb`. Подробную информацию от этих командах см. в их страницах руководства `man`.

Резервные копии журналов могут также создаваться, когда журналы достигают определенного размера, и могут быть созданы скрипты из наборов команд для выполнения до или после операции резервного копирования. В листинге 24 показан более сложный пример.

#### Листинг 24. Другой пример конфигурации logrotate

```
/var/log/messages {  
    rotate 5  
    mail logsave@pinguino  
    size 100k  
    postrotate  
        /usr/bin/killall -HUP syslogd  
    endscript  
}
```

В этом примере ротация `/var/log/messages` производится по достижении им размера 100 КБ. Накапливается пять резервных копий, и когда истекает срок жизни самой старой резервной копии, она отсылается по почте на адрес `logsave@pinguino`. Командное слово `postrotate` включает скрипт, перезапускающий демон `syslogd` после завершения ротации путем отправки сигнала HUP. Командное слово `endscript` необходимо для завершения скрипта, а также в случае, если имеется скрипт `prerotate`. Более полную информацию см. в страницах руководства `man` для `logrotate`.

#### Изучение и мониторинг журналов с целью выявления повышенной активности

Записи в журналах обычно содержат метку времени, имя хоста, на котором выполняется описываемый процесс, и имя процесса. В листинге 25 показано несколько строк из файла `/var/log/messages`, содержащих записи для `gconfd`, `ntpd`, `init` и `yum`.

#### Листинг 25. Пример записей в журнале

```
Jul  5 15:28:24 lyrebird gconfd (root-2832): Exiting  
Jul  5 15:31:06 lyrebird ntpd[2063]: synchronized to 87.98.219.90, stratum 2  
Jul  5 15:31:06 lyrebird ntpd[2063]: kernel time sync status change 0001  
Jul  5 15:31:24 lyrebird init: Trying to re-exec init  
Jul  5 15:31:24 lyrebird yum: Updated: libselinux.i386 2.0.14-2.fc7  
Jul  5 15:31:24 lyrebird yum: Updated: libsemanage.i386 2.0.3-4.fc7  
Jul  5 15:31:25 lyrebird yum: Updated: cups-langs.i386 1.2.11-2.fc7  
Jul  5 15:31:25 lyrebird yum: Updated: libXfont.i386 1.2.9-2.fc7  
Jul  5 15:31:27 lyrebird yum: Updated: NetworkManager.i386 0.6.5-7.fc7  
Jul  5 15:31:27 lyrebird yum: Updated: NetworkManager-glib.i386 0.6.5-7.fc7
```

Просматривать журналы можно при помощи программы постраничного вывода, например, `less`, искать определенные записи (например, сообщения ядра от хоста `lyrebird`) можно при помощи команды `grep`, как показано в листинге 26.

## Листинг 26. Просмотр журналов

```
[root@lyrebird ~]# less /var/log/messages
[root@lyrebird ~]# grep "lyrebird kernel" /var/log/messages | tail -n 9
Jul  5 15:26:46 lyrebird kernel: Bluetooth: HCI socket layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: L2CAP ver 2.8
Jul  5 15:26:46 lyrebird kernel: Bluetooth: L2CAP socket layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: RFCOMM socket layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: RFCOMM TTY layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: RFCOMM ver 1.8
Jul  5 15:26:46 lyrebird kernel: Bluetooth: HIDP (Human Interface Emulation) ver 1.2
Jul  5 15:26:59 lyrebird kernel: [drm] Initialized drm 1.1.0 20060810
Jul  5 15:26:59 lyrebird kernel: [drm] Initialized i915 1.6.0 20060119 on minor 0
```

## Мониторинг журналов

Время от времени может возникать необходимость мониторинга системных журналов с целью поиска событий. Например, можно попробовать поймать редко случающееся событие в тот момент, когда оно произошло. В таком случае можно использовать команду `tail` с опцией `-f` для отслеживания содержимого системного журнала. В листинге 27 показан пример.

## Листинг 27. Отслеживание обновлений в системном журнале

```
[root@lyrebird ~]# tail -n 1 -f /var/log/messages
Jul  6 15:16:26 lyrebird syslogd 1.4.2: restart.
Jul  6 15:16:26 lyrebird kernel: klogd 1.4.2, log source = /proc/kmsg started.
Jul  6 15:19:35 lyrebird yum: Updated: samba-common.i386 3.0.25b-2.fc7
Jul  6 15:19:35 lyrebird yum: Updated: procps.i386 3.2.7-14.fc7
Jul  6 15:19:36 lyrebird yum: Updated: samba-client.i386 3.0.25b-2.fc7
Jul  6 15:19:37 lyrebird yum: Updated: lib smbclient.i386 3.0.25b-2.fc7
Jul  6 15:19:46 lyrebird gconfd (ian-3267): Received signal 15, shutting down cleanly
Jul  6 15:19:46 lyrebird gconfd (ian-3267): Exiting
Jul  6 15:19:57 lyrebird yum: Updated: bluez-gnome.i386 0.8-1.fc7
```

## Обнаружение в журналах сообщений о проблемах

Выявив проблему, вы захотите записать время, имя хоста и имя процесса, породившего проблему. Если сообщение позволяет точно идентифицировать проблему для ее решения, вы это делаете. Если нет, вам может понадобиться обновить `syslog.conf`, чтобы указать, какие дополнительные сообщения для соответствующей категории должны быть записаны в системный журнал. Например, у вас может возникнуть необходимость показать информационное сообщение вместо предупредительного или даже отладить уровень сообщения. Приложение может иметь дополнительные категории, которые можно использовать.

Наконец, если вам необходимо поместить в системный журнал пометки, которые помогут узнать, какие сообщения были зажурнилированы и на какой стадии находится процесс отладки, можно воспользоваться запускаемой из терминального окна командой `logger` или скриптом `shell`, чтобы отправить сообщение, содержащее информацию о вашем выборе, демону `syslog` для журналирования в соответствии с правилами из `syslog.conf`.

## Планирование задач

Этот раздел охватывает материал по теме 1.111.4 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 4.

Из этого раздела вы узнаете, как:

- Использовать команды **cron** или **anacron** для запуска задач через равные промежутки времени
- Использовать команду **at** для запуска задач в определенное время
- Управлять задачами из cron и at
- Настроить доступ пользователя к сервисам cron и at

Из предыдущего раздела вы узнали о команде **logrotate** и увидели, что она должна запускаться через определенные промежутки времени. В следующих двух разделах, касающихся сервисов резервного копирования и сетевой службы времени, вы встретите в ту же необходимость регулярного запуска команд. Это только некоторые из множества задач администрирования, которые должны выполняться многоократно и регулярно. Из этого раздела вы узнаете о средствах, используемых для автоматизации периодического планирования задач, а также о средствах, используемых для запуска задач в какое-то определенное время.

### Запуск задач через равные промежутки времени

Запуском задач через равные промежутки времени управляет *cron*, состоящий из демона **crond** и набора таблиц, описывающих, какая работа должна быть выполнена и с какой периодичностью. Демон просыпается каждую минуту и проверяет crontab'ы, чтобы определить, что необходимо сделать. Пользователи управляют crontab'ами при помощи программы **crontab**. Демон **crond** обычно запускается процессом init в момент запуска системы.

Для простоты давайте предположим, что вы хотите регулярно запускать команду, показанную в листинге 28. Фактически эта команда только выдает сообщение о дате и времени, но она иллюстрирует приемы использования **crontab** для настройки заданий для cron, и из ее вывода узнаем, когда запускался cron. Для настройки записей в crontab необходима строка с escape-метасимволами shell, поэтому лучше сделать это при помощи простых команд и параметров, так что в этом примере команда **echo** будет запущена скриптом /home/ian/mycrontab.sh, которому не требуются параметры. Это избавит от необходимости использовать escape-символы.

### Листинг 28. Пример несложной команды

```
[ian@lyrebird ~]$ cat mycrontest.sh
#!/bin/bash
echo "It is now $(date +%T) on $(date +%A)"
[ian@lyrebird ~]$ ./mycrontest.sh
It is now 18:37:42 on Friday
```

### Создание crontab

Для создания crontab используется команда **crontab** с опцией **-e** ("edit"). Откроется редактор **vi**, если только вы не задали другой редактор в переменной окружения **EDITOR** или **VISUAL**.

Каждая запись в crontab состоит из шести полей:

1. Минута
2. Час
3. День месяца
4. Месяц года
5. День недели
6. Стока, которая должна быть запущена на исполнение при помощи `sh`

Значения для минут и часов колеблются в диапазоне 0-59 и 0-12 соответственно, для дня месяца и месяца — в диапазоне 1-31 и 1-12 соответственно. День недели может обозначаться в диапазоне 0-6, причем 0 означает воскресенье. День недели также может быть задан как sun, mon, tue и т.д. Шестое поле, в которое входит все, что располагается после пятого поля, — строка, которая передается `sh`. Символ процента (%) используется для обозначения новой строки, поэтому если вы хотите использовать % или другой специальный символ, перед ним надо поставить обратный слеш (\). Стока до первого % передается shell"у, а все строки после % передаются на стандартный ввод.

Некоторые связанные со временем поля могут определяться отдельным значением, диапазоном значений, например, 0-10 или sun-wed, или разделенным запятыми списком отдельных значений и диапазонов. В листинге 29 показана в некоторой степени искусственно созданная запись в crontab для нашего примера.

### **Листинг 29. Несложный пример crontab**

```
0,20,40 22-23 * 7 fri-sat /home/ian/mycrontest.sh
```

В этом примере наша команда исполняется каждую 0-ю, 20-ю и 40-ю минуту (каждые 20 минут) часа между 10 часами вечера и полуночью по пятницам и субботам в течение июля. Подробности о других способах определения времени см. в страницах руководства man для crontab(5).

### **Как насчет вывода?**

Вы можете заинтересоваться, что происходит с выводом команды. Большинство команд, предназначенных для использования совместно с cron, записывают вывод в журнал при помощи функции syslog, о которой вы узнали из предыдущего раздела. Однако любой вывод, направленный на стандартный вывод, будет отправлен пользователю по почте. В листинге 30 показан вывод, который вы могли бы получить от команды из нашего примера.

### **Листинг 30. Вывод cron, отправленный по почте**

```
From ian@lyrebird.raleigh.ibm.com Fri Jul 6 23:00:02 2007
Date: Fri, 6 Jul 2007 23:00:01 -0400
From: root@lyrebird.raleigh.ibm.com (Cron Daemon)
To: ian@lyrebird.raleigh.ibm.com
Subject: Cron <ian@lyrebird> /home/ian/mycrontest.sh
Content-Type: text/plain; charset=UTF-8
Auto-Submitted: auto-generated
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/home/ian>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=ian>
X-Cron-Env: <USER=ian>
```

It is now 23:00:01 on Friday

## Где мой crontab?

Crontab, созданный вами при помощи команды **crontab**, хранится в /etc/spool/cron под именем пользователя, создавшего его. Так показанный выше crontab хранится в /etc/spool/cron/ian. Так что вы не удивитесь, узнав, что команда **crontab**, подобно рассмотренной ранее команде **passwd**, является uid-программой, которая запускается с полномочиями пользователя root.

## /etc/crontab

Помимо пользовательских файлов crontab в /var/spool/cron, **cron** также проверяет /etc/crontab и файлы в каталоге /etc/cron.d. Эти системные crontab'ы имеют дополнительное поле между пятым полем для времени (день) и командой. Это дополнительное поле определяет пользователя, для которого будет запущена команда, обычно это root. /etc/crontab может выглядеть как в примере из листинга 31.

### Листинг 31. /etc/crontab

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

В этом примере реальная работа выполняется при помощи команды **run-parts**, которая запускает скрипты из /etc/cron.hourly, /etc/cron.daily и т.д.; /etc/crontab просто контролирует время выполнения повторяющихся заданий. Обратите внимание, что все команды здесь запущены от имени пользователя root. Заметьте также, что crontab может содержать присваивания значений переменным shell, которые будут установлены перед запуском команды.

## Anacron

Функция cron хорошо работает в системах, работающих постоянно. В системах, которые могут быть отключены в течение долгого времени, например, в лэптопах, для управления планированием задач, обычно выполняемых ежедневно, еженедельно или ежемесячно, может использоваться команда *anacron* ("anachronistic cron", "анахроничный cron"). Anacron не используется для выполнения задач каждый час.

Anacron хранит метки времени файлов в /var/spool/anacron, чтобы записывать время выполнения заданий. При запуске anacron проверяет, прошло ли необходимое количество дней с тех пор, как задача была выполнена в последний раз, и при необходимости запускает ее. Таблица заданий для anacron хранится в /etc/anacrontab, формат которого немного отличается от формата /etc/crontab. Как и /etc/crontab, /etc/anacrontab может содержать настройки окружения. Каждое задание имеет четыре поля.

1. период
2. задержка

### 3. идентификатор задачи

### 4. команда

Период — число дней, но он может быть также определен как @monthly для гарантии, что задача будет выполняться только один раз в месяц независимо от числа дней в месяце.

Задержка — число минут ожидания после того как задача должна быть запущена, но до того как она действительно начнет выполняться. Эта возможность может использоваться для предотвращения запуска слишком большого количества задач при первом старте системы. Идентификатор задачи может содержать любой отличный от пробела символ, за исключением слешей (/).

И /etc/crontab, и /etc/anacrontab обновляются путем непосредственного редактирования. Для обновления этих файлов или файлов в каталоге /etc/cron.d команда **crontab** не используется.

## Запуск задач в определенное время

Иногда может быть необходимо запустить задание только один раз, а не делать это регулярно. Для этого используется команда **at**. Команды, которые должны быть запущены, берутся из файла, задаваемого при помощи опции -f, или со стандартного ввода, если опция -f не используется. Опция -m посыпает пользователю письмо, даже если команда не имеет стандартного вывода. Опция -v показывает, в какое время будет запущена задача. Время также показано в выводе. В листинге 32 показаны примеры запуска использовавшегося ранее скрипта mycrontest.sh. В листинге 33 показаны выводы, которые возвращаются пользователю после запуска задания. Обратите внимание, что вывод немного более компактен по сравнению с соответствующим выводом задачи cron.

## Листинг 32. Использование команды at

```
[ian@lyrebird ~]$ at -f mycrontest.sh -v 10:25
Sat Jul  7 10:25:00 2007

job 5 at Sat Jul  7 10:25:00 2007
```

## Листинг 33. Вывод задачи, запущенной at

```
From ian@lyrebird.raleigh.ibm.com Sat Jul  7 10:25:00 2007
Date: Sat, 7 Jul 2007 10:25:00 -0400
From: Ian Shields <ian@lyrebird.raleigh.ibm.com>
Subject: Output from your job      5
To: ian@lyrebird.raleigh.ibm.com
```

It is now 10:25:00 on Saturday

Спецификации времени могут быть довольно сложными. В листинге 34 показано несколько примеров. Ознакомьтесь со страницами руководства man для **at**, или с файлом /usr/share/doc/at/timespec, или с файлом /usr/share/doc/at-3.1.10/timespec, где 3.1.10 — версия пакета at.

#### Листинг 34. Варианты задания времени в команде at

```
[ian@lyrebird ~]$ at -f mycrontest.sh 10pm tomorrow
job 14 at Sun Jul  8 22:00:00 2007
[ian@lyrebird ~]$ at -f mycrontest.sh 2:00 tuesday
job 15 at Tue Jul 10 02:00:00 2007
[ian@lyrebird ~]$ at -f mycrontest.sh 2:00 july 11
job 16 at Wed Jul 11 02:00:00 2007
[ian@lyrebird ~]$ at -f mycrontest.sh 2:00 next week
job 17 at Sat Jul 14 02:00:00 2007
```

Команда **at** также имеет опцию **-q**. Увеличение очереди увеличивает для задачи значение nice. Существует также команда **batch**, похожая на команду **at**, за исключением того, что она запускает задачи только при достаточно низкой системной нагрузке. Подробности об этих командах см. в страницах руководств man.

#### Управление запланированными задачами

##### Просмотр запланированных задач

Задачами cron и at можно управлять. Как показано в листинге 35, для просмотра crontab используется команда **crontab** с опцией **-l**, для просмотра задач, поставленных в очередь командой **at**, используется команда **atq**.

#### Листинг 35. Просмотр запланированных задач

```
[ian@lyrebird ~]$ crontab -l
0,20,40 22-23 * 7 fri-sat /home/ian/mycrontest.sh
[ian@lyrebird ~]$ atq
16      Wed Jul 11 02:00:00 2007 a ian
17      Sat Jul 14 02:00:00 2007 a ian
14      Sun Jul  8 22:00:00 2007 a ian
15      Tue Jul 10 02:00:00 2007 a ian
```

Чтобы посмотреть, какая команда запланирована для исполнения при помощи **at**, используется команда **at** с опцией **-c** и номером задания. Вы заметите, что большая часть окружения, которая была активна во время запуска команды **at**, сохранена для запланированного задания. В листинге 36 показана часть вывода для задания 15.

#### Листинг 36. Использование at -c в сочетании с номером задания

```
#!/bin/sh
# atrun uid=500 gid=500
# mail ian 0
umask 2
HOSTNAME=lyrebird.raleigh.ibm.com; export HOSTNAME
SHELL=/bin/bash; export SHELL
HISTSIZE=1000; export HISTSIZE
SSH_CLIENT=9.67.219.151\ 3210\ 22; export SSH_CLIENT
SSH_TTY=/dev/pts/5; export SSH_TTY
USER=ian; export USER
...
HOME=/home/ian; export HOME
LOGNAME=ian; export LOGNAME
...
```

```

cd /home/ian || {
    echo 'Execution directory inaccessible' >&2
    exit 1
}
${SHELL:-/bin/sh} << ` (dd if=/dev/urandom count=200 bs=1 \
2>/dev/null|LC_ALL=C tr -d -c '[:alnum:]')` 

#!/bin/bash
echo "It is now $(date +%T) on $(date +%A)"

```

Обратите внимание, что содержимое нашего скрипта было скопировано в виде встроенного документа, который будет выполнен shell'ом, установленным в переменной SHELL, или /bin/sh, если переменная SHELL не установлена. Чтобы узнать о встроенных документах, обратитесь к [Учебнику для экзамена LPI 101, Тема 103: Команды GNU и UNIX](#).

### Удаление запланированных задач

Удалить запланированные задачи можно при помощи команды **cron** с опцией **-r**, как показано в листинге 37.

#### Листинг 37. Просмотр и удаление заданий cron

```

[ian@lyrebird ~]$ crontab -l
0,20,40 22-23 * 7 fri-sat /home/ian/mycrontest.sh
[ian@lyrebird ~]$ crontab -r
[ian@lyrebird ~]$ crontab -l
no crontab for ian

```

Для удаления системных заданий cron или anacron отредактируйте /etc/crontab и /etc/anacrontab или отредактируйте или удалите файлы в каталоге /etc/cron.d.

При помощи команды **atrm** с указанием номера задания можно удалить одно или более заданий, запланированных при помощи команды **at**. Несколько заданий должны быть разделены пробелами. В листинге 38 показан пример.

#### Листинг 38. Просмотр и удаление заданий при помощи atq и atrm

```

[ian@lyrebird ~]$ atq
16      Wed Jul 11 02:00:00 2007 a ian
17      Sat Jul 14 02:00:00 2007 a ian
14      Sun Jul  8 22:00:00 2007 a ian
15      Tue Jul 10 02:00:00 2007 a ian
[ian@lyrebird ~]$ atrm 16 14 15
[ian@lyrebird ~]$ atq
17      Sat Jul 14 02:00:00 2007 a ian

```

### Настройка доступа пользователя к планированию задач

Чтобы не только root, но и все другие пользователи имели возможность воспользоваться **crontab** и функцией cron, они должны быть перечислены в файле /etc/cron.allow, если он существует. Если файл /etc/cron.allow не существует, но существует файл /etc/cron.deny, пользователи, перечисленные в нем, не могут использовать **crontab** и функцию cron. Если ни один из этих файлов не существует, использовать эту команду может только

суперпользователь. Если файл /etc/cron.deny пуст, это означает, что все пользователи могут использовать функцию cron. По умолчанию /etc/cron.deny пуст.

Для функции at подобный смысл имеют соответствующие файлы /etc/at.allow и /etc/at.deny.

## Резервное копирование данных

Этот раздел охватывает материал по теме 1.111.5 экзамена 102 Администрирование Linux для начинающих (LPIC-1). Рейтинг темы 3.

Из этого раздела вы узнаете, как:

- Планировать стратегию резервного копирования
- Создать дамп сырого устройства в файле или восстановить содержимое сырого устройства из файла
- Осуществить частичное или ручное резервное копирование
- Проверить целостность файлов резервного копирования
- Частично или полностью восстановить файловые системы из резервных копий

## Планирование стратегии резервного копирования

Наличие хорошей резервной копии — необходимая часть системного администрирования, но принятие решения о том, что, когда и как должно копироваться, может вызвать затруднения. В сфере бизнеса обычно бывают крайне важны базы данных, такие как заказы клиентов или описание имущества, и часто используются специализированные средства резервного копирования и восстановления данных, описание которых выходит за рамки этого учебного пособия. С другой стороны, некоторые файлы являются временными, и нет необходимости в их резервном копировании. В этом разделе мы фокусируем внимание на системных файлах и пользовательских данных и обсуждаем некоторые принципы, методы и средства резервного копирования таких данных.

Существует три основных метода резервного копирования:

1. *Полное* резервное копирование — обычно создание резервной копии всей файловой системы, каталога или группы связанных файлов. Создание такой копии занимает длительное время и обычно осуществляется в сочетании с одним из следующих двух методов.
2. *Дифференциальное* или *кумулятивное* резервное копирование — резервное копирование всех данных, которые изменились после создания последней полной резервной копии. Для восстановления требуется полная резервная копия плюс самая последняя дифференциальная резервная копия.
3. *Инкрементальное* резервное копирование — резервное копирование только тех данных, которая изменилась после создания последней инкрементальной копии. Для восстановления требуется полная резервная копия плюс все инкрементальные копии (по порядку), созданные после последнего полного резервного копирования.

## Что подлежит резервному копированию

Решая, что копировать, следует принять во внимание, насколько меняются данные. Это поможет определить, как часто должны создаваться их резервные копии. Подобным образом, резервные копии наиболее важных данных должны создаваться чаще, чем копии менее важных данных. Вашу операционную систему, вероятно, можно будет относительно легко восстановить, особенно если вы используете общий образ для нескольких систем, хотя важнее было бы создать копии файлов настроек для каждой системы.

Для отдела разработчиков может быть достаточно хранить резервные копии репозиториев, таких как репозитории CVS, в то время как личные песочницы (sandbox'ы) программистов

могут быть менее важными. В зависимости от того, насколько важна для вашей деятельности электронная почта, может быть достаточно иметь нечасто создаваемые резервные копии почты или может быть необходимо иметь возможность восстановить почту на самую последнюю возможную дату. У вас может возникнуть желание хранить резервные копии файлов системного стоп, но вы не станете волноваться о запланированных задачах отдельных пользователей.

Filesystem Hierarchy Standard предоставляет классификацию данных, которые могут помочь при выборе объектов и методов резервного копирования. Подробнее см. в [Учебнике для экзамена LPI 101, Тема 104: Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard](#).

Как только вы решили, что подлежит резервному копированию, необходимо решить, как часто следует делать полную копию и делать ли дифференциальные или инкрементальные резервные копии между созданием полных резервных копий. После принятия этих решений следующие советы помогут в выборе соответствующих инструментов.

### Автоматизация резервного копирования

Из предыдущего раздела вы узнали, как планировать задачи, а также о функции cron, которая идеально поможет автоматизировать планирование резервного копирования. Однако резервные копии часто записываются на сменные носители, в основном, на пленку, поэтому, вероятно, будет необходимым вмешательство оператора. Чтобы гарантировать, что процесс резервного копирования происходит автоматически и обладает воспроизводимостью, насколько это возможно, необходимо создать и использовать соответствующие скрипты.

### Создание дампов и восстановление содержимого сырых устройств

Один из способов создания полной резервной копии файловой системы — создать образ разделов, на которых она расположена. *Сырое устройство*, например, /dev/hda1 или /dev/sda2, может быть открыто и прочитано как последовательный файл. Точно так же оно может быть записано с резервной копии как последовательный файл. Это не требует со стороны средства резервного копирования знаний относительно расположения файловой системы, но необходимо, чтобы восстановление было сделано в такое место, которое имеет по крайней мере такой размер, как оригинал. Некоторые средства для управления сырыми устройствами *готовы к работе с файловой системой*, что означает, что они понимают одну или более файловых систем Linux. Эти утилиты могут создать дамп сырого устройства, но не могут создать дамп неиспользованной части раздела. Они могут требовать или не требовать для восстановления наличия раздела такого же или большего размера. Команда **dd** — пример утилиты первого типа, а команда **dump** — пример утилиты второго типа, который характерен для файловых систем типа ext2 и ext3.

#### Команда dd

Самая простая форма использования команды **dd** — копирование входного файла в выходной файл, где любой файл может быть сырым устройством. Для резервного копирования сырого устройства, такого как /dev/hda1 или /dev/sda2, входной файл будет сырым устройством. В идеале для уверенности, что в ходе резервного копирования данные не будут изменены, файловая система не должна быть смонтирована на устройстве или смонтирована только для чтения. В листинге 39 показан пример.

#### Листинг 39. Резервное копирование разделов при помощи dd

```
[root@lyrebird ~]# dd if=/dev/sda3 of=backup-1
2040255+0 records in
2040255+0 records out
```

```
1044610560 bytes (1.0 GB) copied, 49.3103 s, 21.2 MB/s
```

Параметры `if` и `of` определяют входной и выходной файлы соответственно. В этом примере входной файл — сырое устройство `dev/sda3`, а выходной файл — файл `backup-1` в домашнем каталоге пользователя `root`. Чтобы создать дамп файла для записи его на пленку или на дискету, следует указать что-то типа `of=/dev/fd0` или `of=/dev/st0`.

Обратите внимание, что было скопировано 1,044,610,560 байт данных, и выходной файл имеет очень большой размер, несмотря на то, что фактически используется только около 3% этого конкретного раздела. Вы, наверно, захотите сжать данные, если только вы не используете при копировании на ленту аппаратное сжатие. В листинге 40 показан способ достичь этого, а также вывод команд `ls` и `df`, которые показывают размеры файлов и процент использования файловой системы на `/dev/sda3`.

#### Листинг 40. Резервное копирование с сжатием при помощи dd

```
[root@lyrebird ~]# dd if=/dev/sda3 | gzip > backup-2
2040255+0 records in
2040255+0 records out
1044610560 bytes (1.0 GB) copied, 117.723 s, 8.9 MB/s
[root@lyrebird ~]# ls -l backup-[12]
-rw-r--r-- 1 root root 1044610560 2007-07-08 15:17 backup-1
-rw-r--r-- 1 root root 266932272 2007-07-08 15:56 backup-2
[root@lyrebird ~]# df -h /dev/sda3
Filesystem           Size   Used  Avail Use% Mounted on
/dev/sda3            972M   28M  944M   3% /grubfile
```

Сжатие при помощи `gzip` уменьшило размер файла примерно до 20% от его полного размера. Однако неиспользованные блоки могут содержать какие-то данные, поэтому даже сжатая резервная копия может быть значительно больше общего размера содержащихся на разделе данных.

Если разделить размер на количество записей, обработанных `dd`, вы увидите, что `dd` записывает данные блоками размером 512 байт. При копировании на сырое выводное устройство, например, ленту, это может серьезно понизить производительность, поэтому `dd` может читать или записывать данные гораздо более крупными блоками. Укажите опцию `obs` для изменения размера вывода или опцию `ibs` для определения размера выводного блока. Также можно определить только `bs`, чтобы установить одинаковый размер блока для ввода и вывода.

Если для хранения резервной копии необходимо сделать запись на несколько лент или сменных накопителей, копию следует разбить на более мелкие части, например, при помощи утилиты `split`.

Если необходимо пропустить блоки, например, ярлыки дисков или лент, это можно сделать при помощи `dd`. Примеры см. в страницах руководства `man`.

Помимо простого копирования данных, команда `dd` может преобразовывать данные, например, между ASCII и EBCDIC, между порядками "от старшего к младшему" (big-endian) и "от младшего к старшему" (little-endian) или между записями данных переменной длины и записями данных фиксированной длины. Очевидно, эти преобразования могут быть полезны при копировании реальных файлов, а не сырых устройств. Подробности также см. в страницах руководства `man`.

## Команда dump

Команда **dump** может использоваться для полного, дифференциального или инкрементального резервного копирования на системах ext2 или ext3. В листинге 41 показан пример.

### Листинг 41. Резервное копирование с сжатием при помощи dump

```
[root@lyrebird ~]# dump -0 -f backup-4 -j -u /dev/sda3
DUMP: Date of this level 0 dump: Sun Jul  8 16:47:47 2007
DUMP: Dumping /dev/sda3 (/grubfile) to backup-4
DUMP: Label: GRUB
DUMP: Writing 10 Kilobyte records
DUMP: Compressing output at compression level 2 (bzlib)
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 12285 blocks.
DUMP: Volume 1 started with block 1 at: Sun Jul  8 16:47:48 2007
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing backup-4
DUMP: Volume 1 completed at: Sun Jul  8 16:47:57 2007
DUMP: Volume 1 took 0:00:09
DUMP: Volume 1 transfer rate: 819 kB/s
DUMP: Volume 1 12260kB uncompressed, 7377kB compressed, 1.662:1
DUMP: 12260 blocks (11.97MB) on 1 volume(s)
DUMP: finished in 9 seconds, throughput 1362 kB/sec
DUMP: Date of this level 0 dump: Sun Jul  8 16:47:47 2007
DUMP: Date this dump completed: Sun Jul  8 16:47:57 2007
DUMP: Average transfer rate: 819 kB/s
DUMP: Wrote 12260kB uncompressed, 7377kB compressed, 1.662:1
DUMP: DUMP IS DONE
[root@lyrebird ~]# ls -l backup-[2-4]
-rw-r--r-- 1 root root 266932272 2007-07-08 15:56 backup-2
-rw-r--r-- 1 root root 266932272 2007-07-08 15:44 backup-3
-rw-r--r-- 1 root root    7554939 2007-07-08 16:47 backup-4
```

В этом примере **-0** определяет *уровень дампа*, выражаящийся целым числом, исторически сложилось, что используется значение от 0 до 9, где 0 обозначает полный дамп. Опция **-f** определяет выходной файл, который может быть сырьим устройством. Укажите **-**, чтобы направить вывод на стандартный вывод. Опция **-j** определяет уровень сжатия по умолчанию, равный 2, с использованием сжатия bzlib. Если вы предпочитаете сжатие zlib, используйте опцию **-z**. Опция **-U** указывает, что запись информации о дампе, обычно это /etc/dumpdates, должна быть обновлена. Все параметры, стоящие после опций, — файл или список файлов, причем файл также может быть сырьим устройством, как в этом примере. Обратите внимание, насколько резервная копия меньше в случае, если программа резервного копирования осведомлена о структуре файловой системы и может не сохранять неиспользуемые блоки устройства.

Если выводом является такое устройство как лента, когда его объем будет полностью использован, команда **dump** запросит другой том. Также можно предусмотреть несколько имен файлов, разделенных запятыми. Например, если вам нужно, чтобы автоматически был создан дамп, которому требуется две ленты, вы можете вставить ленты в /dev/st0 и /dev/st1, запланировать команду **dump**, указав обе ленты в качестве вывода, и отправиться домой спать.

Если определить уровень дампа выше 0, будет создан инкрементальный дамп из всех новых файлов и файлов, изменившихся с момента создания последнего дампа уровня меньшего, чем данный. Поэтому дамп уровня 1 будет дифференциальным, даже если одновременно был получен дамп уровня 2 или выше. В листинге 42 показан результат обновления метки времени существующего файла на /dev/sda3 и создания нового файла, а затем сделан дамп уровня 2. После этого создан другой новый файл и сделан дамп уровня 1. Также показана информация из /etc/dumpdates. Для краткости часть вывода второго дампа опущена.

#### Листинг 42. Резервное копирование с сжатием при помощи dump

```
[root@lyrebird ~]# dump -2 -f backup-5 -j -u /dev/sda3
DUMP: Date of this level 2 dump: Sun Jul  8 16:55:46 2007
DUMP: Date of last level 0 dump: Sun Jul  8 16:47:47 2007
DUMP: Dumping /dev/sda3 (/grubfile) to backup-5
DUMP: Label: GRUB
DUMP: Writing 10 Kilobyte records
DUMP: Compressing output at compression level 2 (bzlib)
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 91 blocks.
DUMP: Volume 1 started with block 1 at: Sun Jul  8 16:55:47 2007
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing backup-5
DUMP: Volume 1 completed at: Sun Jul  8 16:55:47 2007
DUMP: 90 blocks (0.09MB) on 1 volume(s)
DUMP: finished in less than a second
DUMP: Date of this level 2 dump: Sun Jul  8 16:55:46 2007
DUMP: Date this dump completed: Sun Jul  8 16:55:47 2007
DUMP: Average transfer rate: 0 kB/s
DUMP: Wrote 90kB uncompressed, 15kB compressed, 6.000:1
DUMP: DUMP IS DONE
[root@lyrebird ~]# echo "This data is even newer" >/grubfile/newerfile
[root@lyrebird ~]# dump -1 -f backup-6 -j -u -A backup-6-toc /dev/sda3
DUMP: Date of this level 1 dump: Sun Jul  8 17:08:18 2007
DUMP: Date of last level 0 dump: Sun Jul  8 16:47:47 2007
DUMP: Dumping /dev/sda3 (/grubfile) to backup-6
...
DUMP: Wrote 100kB uncompressed, 16kB compressed, 6.250:1
DUMP: Archiving dump to backup-6-toc
DUMP: DUMP IS DONE
[root@lyrebird ~]# ls -l backup-[4-6]
-rw-r--r-- 1 root root 7554939 2007-07-08 16:47 backup-4
-rw-r--r-- 1 root root   16198 2007-07-08 16:55 backup-5
-rw-r--r-- 1 root root   16560 2007-07-08 17:08 backup-6
[root@lyrebird ~]# cat /etc/dumpdates
/dev/sda3 0 Sun Jul  8 16:47:47 2007 -0400
/dev/sda3 2 Sun Jul  8 16:55:46 2007 -0400
/dev/sda3 1 Sun Jul  8 17:08:18 2007 -0400
```

Обратите внимание, что backup-6 на самом деле больше, чем backup 5. Дамп уровня 1 иллюстрирует использование опции **-A** для создания таблицы содержимого, которая может использоваться, чтобы определить, находится ли файл в архиве, без необходимости действительно монтировать архив. Это особенно полезно при использовании лент или других сменных архивных накопителей. Вы снова увидите эти примеры позже в этом разделе, когда мы будем обсуждать восстановление данных.

Команда **dump** может создавать файлы или подкаталоги дампа, но не может обновить /etc/dumpdates, и поддерживается только уровень дампа 0, то есть полный дамп.

Листинг 43 иллюстрирует процесс формирования и записи на дискету дампа каталога /usr/include/bits и его содержимого при помощи команды **dump**. В этом случае дамп не помещается на одну дискету, поэтому требуется новый том. Запрос и ответ выделены жирным шрифтом.

#### Листинг 43. Резервное копирование каталога в несколько томов при помощи команды **dump**

```
[root@lyrebird ~]# dump -0 -f /dev/fd0 /usr/include/bits
DUMP: Date of this level 0 dump: Mon Jul  9 16:03:23 2007
DUMP: Dumping /dev/sdb9 (/ (dir usr/include/bits)) to /dev/fd0
DUMP: Label: /
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 2790 blocks.
DUMP: Volume 1 started with block 1 at: Mon Jul  9 16:03:30 2007
DUMP: dumping (Pass III) [directories]
DUMP: End of tape detected
DUMP: Closing /dev/fd0
DUMP: Volume 1 completed at: Mon Jul  9 16:04:49 2007
DUMP: Volume 1 1470 blocks (1.44MB)
DUMP: Volume 1 took 0:01:19
DUMP: Volume 1 transfer rate: 18 kB/s
DUMP: Change Volumes: Mount volume #2
DUMP: Is the new volume mounted and ready to go?: ("yes" or "no") y
DUMP: Volume 2 started with block 1441 at: Mon Jul  9 16:05:10 2007
DUMP: Volume 2 begins with blocks from inode 2
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /dev/fd0
DUMP: Volume 2 completed at: Mon Jul  9 16:06:28 2007
DUMP: Volume 2 1410 blocks (1.38MB)
DUMP: Volume 2 took 0:01:18
DUMP: Volume 2 transfer rate: 18 kB/s
DUMP: 2850 blocks (2.78MB) on 2 volume(s)
DUMP: finished in 109 seconds, throughput 26 kBytes/sec
DUMP: Date of this level 0 dump: Mon Jul  9 16:03:23 2007
DUMP: Date this dump completed: Mon Jul  9 16:06:28 2007
DUMP: Average transfer rate: 18 kB/s
DUMP: DUMP IS DONE
```

Если резервная копия записывается на ленту, следует помнить, что обычно лента перематывается после каждого использования. Устройство с именем типа /dev/st0 или /dev/st1 перематывается автоматически. Соответствующие неперематываемые эквивалентные устройства — /dev/nst0 и /dev/nst1. В любом случае всегда можно воспользоваться командой **mt** для выполнения таких операций с магнитной лентой как проматывание файлов и записей, перемотка и запись отметок конца файла (EOF marks). Дополнительную информацию см. в страницах руководств **man** для **mt** и **st**.

Разумный выбор уровней дампов позволит минимизировать количество архивов, необходимых для восстановления к любому определенному уровню. Пример стратегии на основе головоломки Ханойская башня см. в страницах руководства **man** для **dump**.

Эти команды, как и команда **dd**, имеют большое количество опций, не описанных в этом

кратком введении. Подробнее см. в страницах руководства man.

## Частичное и ручное резервное копирование

До сих пор вы знакомились с инструментами, которые хорошо работают резервном копировании файловых систем целиком. Иногда бывает необходима резервная копия не всей файловой системы, а отдельных файлов или подкаталогов. Например, может понадобиться создавать резервную копию большей части файловой системы еженедельно, а резервную копию файлов электронной почты — ежедневно. Для этих целей обычно используются другие две команды, `cpio` и `tar`. Обе они могут записывать архивы в файлы или на устройства, например, на ленты и дискеты, и обе могут восстанавливать данные из таких архивов. Сейчас из этих двух команд чаще используется `tar`, возможно потому, что она лучше работает с полными каталогами, и GNU-версия tar поддерживает сжатие и с помощью gzip, и с помощью bzip.

### Использование cpio

Команда `cpio` работает для создания архива в режиме *copy-out*, для восстановления архива — в режиме *copy-in*, для копирования набора файлов из одного места в другое — в режиме *copy-pass*. В режиме copy-out используются опции `-o` или `--create`, в режиме copy-in — опции `-i` или `--extract` и в режиме copy-pass — опции `-p` или `--pass-through`. Вводом является список файлов, получаемый на стандартный ввод. Вывод происходит на стандартный вывод, или на устройство, или в определенный файл при помощи опции `-f` или `--file`.

В листинге 44 показано, как сгенерировать список файлов при помощи команды `find`. Обратите внимание, что команда `find` с опцией `-print0` используется для создания строк, оканчивающихся на ноль (null-terminate), для имен файлов и команда `cpio` с опцией `--null` — для чтения этого формата. Это позволяет правильно оперировать именами файлов, имеющими пробелы или символы перевода строки.

### Листинг 44. Резервное копирование домашнего каталога при помощи cpio

```
[root@lyrebird ~]# find ~ian -depth -print0 | cpio --null -o >backup-cpio-1  
18855 blocks
```

Чтобы видеть список файлов по мере их архивирования, добавьте к `cpio` опцию `-v`.

Как и при использовании других команд, способных архивировать файлы, можно задать размер блока. Подробнее об этих и других опциях см. в страницах руководства man.

### Использование tar

Команда `tar` (название происходит от *Tape ARchive*) создает архивный файл, или *tarfile*, или *tarball* из набора входных файлов или каталогов; также она восстанавливает файлы из таких архивов. Если в качестве ввода для `tar` используется каталог, все файлы и подкаталоги включаются автоматически, что делает `tar` очень удобным для архивирования поддеревьев структуры каталогов.

Как и для других команд, которые мы обсуждали, вывод может быть направлен в файл, на устройство, такое как лента или дискета, или на стандартный вывод. Местоположение вывода определяется при помощи опции `-f`. Другие наиболее часто используемые опции — это `-C` для создания архива, `-X` для разархивирования, `-V` для подробного вывода, содержащего список обрабатываемых файлов, `-Z` для сжатия с использованием gzip и `-j` для сжатия с использованием bzip2. Большинство опций команды `tar` имеет короткую форму, при которой используется один дефис, и длинную форму, при которой используется пара

дефисов. Описание длинных форм и других опций см. в страницах руководства man.

В листинге 45 показано, как создать резервную копию системных заданий cron при помощи **tar**.

#### Листинг 45. Резервное копирование системных заданий cron при помощи tar

```
[root@lyrebird ~]# tar -czvf backup-tar-1 /etc/*crontab /etc/cron.d
tar: Removing leading `/' from member names
/etc/anacrontab
/etc/crontab
/etc/cron.d/
/etc/cron.d/sa-update
/etc/cron.d/smolt
```

В первой строке вывода указано, что **tar** удалит лидирующий слеш (\) из имен членов. Это позволяет восстановить файлы в какое-то другое место для проверки, прежде чем заменить системные файлы. Это хорошая идея, которая позволит при создании архивов избежать перемешивания абсолютных и относительных имен файлов, поскольку при восстановлении из архива все имена будут относительными.

Команда **tar** при помощи опции **-r** или **--append** может добавить в архив дополнительные файлы. Это может привести к тому, что в архиве будет несколько копий файла. В таком случае в ходе операции восстановления будет восстановлен только *последний файл*. Для выбора одного из нескольких файлов используется опция **--occurrence**. Если архив находится не на ленте, а на обычной файловой системе, для обновления архива используется опция **-u** или **--update**. Это работает подобно дополнению архива, за исключением того, что метки времени для файлов в архиве сравниваются с метками в файловой системе и добавляются только те файлы, которые изменились после создания заархивированной версии. Как было упомянуто, это не работает с архивами на лентах.

Как и другие изучаемые здесь команды, команда **tar** имеет множество опций, не описанных в этом кратком введении. Подробнее см. в страницах руководств man или info.

#### Целостность файла резервной копии

Целостность файла резервной копии чрезвычайно важна. Если резервная копия испорчена, нет смысла хранить ее. Хорошая стратегия резервного копирования также подразумевает проверку резервных копий.

Первый шаг к обеспечению целостности резервной копии — убедиться, что данные, для которых делается резервная копия, собраны правильно. Для того чтобы данные, для которых создается резервная копия, не изменились в процессе копирования, обычно бывает достаточно, чтобы система не была смонтирована или была смонтирована только для чтения. Если вам необходимо создать резервную копию файловых систем, каталогов или файлов, которые меняются в то время как создается резервная копия, следует убедиться, что не было сделано изменений в ходе резервного копирования. Если изменения были, необходимо избрать стратегию для их сбора, или повторив резервное копирование, или, возможно, заменив такие файлы внутри резервной копии. Разумеется, это затрагивает и процедуру восстановления.

Допустим, вы получили хорошие резервные копии, их необходимо периодически проверять. Один из способов состоит в том, чтобы восстановить резервную копию на запасной том и убедиться, что результат совпадает с тем, что было скопировано. Это самое простое, что следует сделать, прежде чем позволить обновить файловые системы данными резервной

копии. Если резервная копия сохранена на медиа-носитель, такой как CD или DVD, можно использовать команду `diff` как часть процедуры резервного копирования, чтобы убедиться в качестве резервной копии. Помните, что даже качественные резервные копии при хранении могут портиться, поэтому, даже если они проверялись во время резервного копирования, их следует периодически проверять. Хранение дайджестов используемых программ, таких как `md5sum` или `shalsum` — также хороший способ проверки целостности файла с резервной копией.

## Восстановление файловых систем из резервных копий

Резервное копирование файлов дает возможность восстановить их при необходимости. Иногда может возникнуть желание восстановить всю файловую систему, но гораздо чаще необходимо восстанавливать только определенные файлы или, возможно, набор каталогов. Почти всегда, прежде чем действительно сделать восстановленные файлы реальными, данные восстанавливаются в какое-то временное место и производится проверка, действительно ли восстановлено то, что нужно, и совместимы ли восстановленные данные с текущим состоянием системы.

Родственная проблема — необходимость убедиться, что нужные элементы находятся в определенной резервной копии, поскольку часто возникает необходимость получить доступ к версии файла, который был изменен, или, возможно, удален "когда-то на прошлой или позапрошлой неделе". Имея в виду все вышесказанное, давайте рассмотрим опции восстановления.

## Восстановление dd-архива

Вспомните, что команда `dd` не распознает файловую систему, поэтому, чтобы узнать, что находится в дампе раздела, необходимо восстановить его. В листинге 46 показано, как раздел, из которого ранее в листинге 39 был создан дамп, восстановить на раздел `/dev/sdc7`, специально созданный на сменном USB-устройстве только с этой целью.

### Листинг 46. Восстановление раздела при помощи dd

```
[root@lyrebird ~]# dd if=backup-1 of=/dev/sdc7
2040255+0 records in
2040255+0 records out
1044610560 bytes (1.0 GB) copied, 44.0084 s, 23.7 MB/s
```

Вспомните, что после того как была получена эта резервная копия, мы добавили к файловой системе на `/dev/sda3` некоторые файлы. Вы увидите, что это действительно так, если подмонтируете недавно восстановленный раздел и сравните его с оригиналом, как показано в листинге. Обратите внимание, что файл, метка времени которого была обновлена при помощи `touch`, здесь не показан, как следовало бы ожидать.

### Листинг 47. Сравнение восстановленного раздела с текущим state

```
[root@lyrebird ~]# mount /dev/sdc7 /mnt/temp-dd/
[root@lyrebird ~]# diff -rq /grubfile/ /mnt/temp-dd/
Only in /grubfile/: newerfile
Only in /grubfile/: newfile
```

## Восстановление dump-архива при помощи restore

Вспомните, что в последний раз мы использовали `dump` для дифференциального резервного копирования и что мы создали таблицу содержимого. В листинге 48 показано, как, используя сам архив (`backup-5`) или таблицу содержимого (`backup-6-toc`), воспользоваться командой `restore` для проверки файлов из архива, созданного при помощи `dump`.

### Листинг 48. Проверка содержимого архивов

```
[root@lyrebird ~]# restore -t -f backup-5
Dump tape is compressed.
Dump date: Sun Jul  8 16:55:46 2007
Dumped from: Sun Jul  8 16:47:47 2007
Level 2 dump of /grubfile on lyrebird.raleigh.ibm.com:/dev/sda3
Label: GRUB
      2      .
100481    ./ibshome
100482    ./ibshome/index.html
      16     ./newfile
[root@lyrebird ~]# restore -t -A backup-6-toc
Dump date: Sun Jul  8 17:08:18 2007
Dumped from: Sun Jul  8 16:47:47 2007
Level 1 dump of /grubfile on lyrebird.raleigh.ibm.com:/dev/sda3
Label: GRUB
Starting inode numbers by volume:
  Volume 1: 2
      2      .
100481    ./ibshome
100482    ./ibshome/index.html
      16     ./newfile
      17     ./newerfile
```

Команда `restore` также может при помощи опции `-C` сравнить содержимое архива с содержимым файловой системы. В листинге 49 мы обновили `newerfile` и затем сравнили резервную копию с файловой системой.

### Листинг 49. Сравнение архива с файловой системой при помощи restore

```
[root@lyrebird ~]# echo "something different" >/grubfile/newerfile
[root@lyrebird ~]# restore -C -f backup-6
Dump tape is compressed.
Dump date: Sun Jul  8 17:08:18 2007
Dumped from: Sun Jul  8 16:47:47 2007
Level 1 dump of /grubfile on lyrebird.raleigh.ibm.com:/dev/sda3
Label: GRUB
filesys = /grubfile
./newerfile: size has changed.
Some files were modified!  1 compare errors
```

Восстановление при помощи команды `restore` может производиться интерактивно или автоматически. В листинге 50 показано, как восстановить `newerfile` в домашний каталог пользователя `root` (так что при необходимости его можно проверить, прежде чем заменить обновленный файл), а затем заменить обновленный файл резервной копией. Этот пример иллюстрирует интерактивное восстановление.

### Листинг 50. Восстановление файла при помощи restore

```
[root@lyrebird ~]# restore -i -f backup-6
Dump tape is compressed.
restore > ?
Available commands are:
    ls [arg] - list directory
    cd arg - change directory
    pwd - print current directory
    add [arg] - add `arg' to list of files to be extracted
    delete [arg] - delete `arg' from list of files to be extracted
    extract - extract requested files
    setmodes - set modes of requested directories
    quit - immediately exit program
    what - list dump header information
    verbose - toggle verbose flag (useful with ``ls'')
    prompt - toggle the prompt display
    help or `?' - print this list
If no `arg' is supplied, the current directory is used
restore > ls new*
newerfile
newfile
restore > add newerfile
restore > extract
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume # (none if no more volumes): 1
set owner/mode for '.'? [yn] y
restore > q
[root@lyrebird ~]# mv -f newerfile /grubfile
```

### Восстановление архива cpio

Команда **cpio** в режиме copy-in (опция **-i** или **--extract**) может вывести список содержимого архива или восстановить выбранные файлы. Использование опции **--absolute-filenames** при перечислении файлов уменьшит количество ненужных сообщений, которые в противном случае выдаст **cpio**, поскольку эта опция отбросит все лидирующие символы / от каждого имени пути, имеющего / в начале. В листинге 51 показан частичный вывод листинга нашего предыдущего архива.

### Листинг 51. Восстановление выбранных файлов при помощи cpio

```
[root@lyrebird ~]# cpio -id --list --absolute-filenames <backup-cpio-1
/home/ian/.gstreamer-0.10/registry.i686.xml
/home/ian/.gstreamer-0.10
/home/ian/.Trash/gnome-terminal.desktop
/home/ian/.Trash
/home/ian/.bash_profile
```

В листинге 52 показано, как восстановить все файлы, содержащие в имени пути или имени файла слово "samp". Вывод пропущен через команду **uniq**, чтобы уменьшить количество сообщений вида "Removing leading '/' ...". Для создания каталога следует использовать опцию **-d**; в противном случае все файлы будут созданы в текущем каталоге. Кроме того, **cpio** не

заменит на файловой системе никакие более новые файлы на архивные копии, если только не использовалась опция **-u** или **--unconditional**.

### Листинг 52. Восстановление избранных файлов при помощи cpio

```
[root@lyrebird ~]# cpio -ivd "*samp*" < backup-cpio-1 2>&1 |uniq  
cpio: Removing leading '/' from member names  
home/ian/crontab.samp  
cpio: Removing leading '/' from member names  
home/ian/sample.file  
cpio: Removing leading '/' from member names  
18855 blocks
```

### Восстановление архива tar

Команда **tar** также может сравнивать архивы с текущей файловой системой и восстанавливать файлы из архивов. Для выполнения сравнения используются опции **-d**, **--compare**, или **--diff**. Вывод покажет файлы, содержимое которых отличается, а также файлы, у которых отличаются метки времени. В листинге 53 показан расширенный вывод (использована опция **-v**), полученный в результате сравнения ранее созданного файла и файлов в /etc после того как с целью изменить метку времени был затронут файл /etc/crontab. Опция **--directory /** дает команде **tar** указание выполнить сравнение, начиная не с текущего, а с корневого каталога.

### Листинг 53. Сравнение архивов и файлов при помощи tar

```
[root@lyrebird ~]# touch /etc/crontab  
[root@lyrebird ~]# tar --diff -vf backup-tar-1 --directory /  
etc/anacrontab  
etc/crontab  
etc/crontab: Mod time differs  
etc/cron.d/  
etc/cron.d/sa-update  
etc/cron.d/smolt
```

В листинге 54 показано, как извлечь из текущего каталога только /etc/crontab и /etc/anacrontab.

### Листинг 54. Извлечение файлов из архива при помощи tar

```
[root@lyrebird ~]# tar -xzvf backup-tar-1 "*tab"  
etc/anacrontab  
etc/crontab
```

Обратите внимание, что **tar**, в отличие от **cpio**, автоматически создает иерархию каталогов. В следующем разделе этого учебного пособия показано, как управлять системным временем.

### Системное время

Этот раздел охватывает материал по теме 1.111.6 экзамена 102 Администрирование Linux для

начинающих (LPIC-1). Рейтинг темы 4.

Из этого раздела вы узнаете, как:

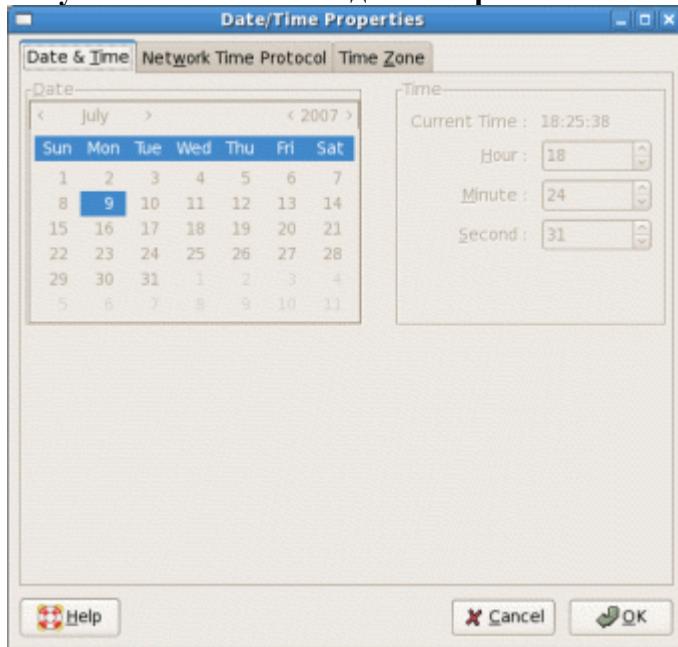
- Установить системную дату и время
- Установить часы BIOS в соответствии со временем UTC
- Настроить часовой пояс
- Настроить сервис Network Time Protocol (NTP), включая корректировку временного отклонения

### Установка системной даты и времени

В системе Linux системное время является крайне важным. Ранее вы видели, что cron и anacron выполняют действия в определенное время, поэтому для корректной работы им необходимо точное время. Большинство средств для резервного копирования и восстановления, которые обсуждались в предыдущем разделе, наряду со средствами разработки, такими как make, также зависят от надежного измерения времени. Большинство компьютеров, собранных примерно с 1980 года, включают некий набор механизмов часов и большинство созданных, начиная с 1984 года, имеет стабильный механизм часов, который поддерживает ход часов, даже если компьютер выключен.

Если вы проводили установку системы Linux в графическом режиме, вы, вероятно, настроили время и выбрали нужный часовой пояс. Можно выбрать, использовать ли для настройки часов Network Time Protocol (NTP), а также выбрать или не выбирать возможность поддержки системных часов с использованием Coordinated Universal Time (UTC). Если позже войти в настройку часов при помощи графических средств в Fedora, Red Hat или другой подобной системе, появится диалоговое окно, подобное изображенному на рисунке 3.

Рисунок 3. Обновление даты и времени



Сюрприз! В этом диалоге фактически невозможно настроить сами часы. Из этого раздела вы больше узнаете о различиях между локальными часами и NTP, а также о том, как настроить системное время.

Независимо от того, живете ли вы в Нью Йорке, Бухаресте, Находке, Улан-Баторе, Бангкоке или Канберре, в Linux большая часть вычислений, связанных со временем, привязана к Coordinated Universal Time (UTC). Если вы запускаете только систему Linux, принято настраивать аппаратные часы в соответствии с UTC, но если вы загружаете также и другую

операционную систему, такую как Windows, может понадобиться настроить аппаратные часы на местное время. Поскольку рассматривается Linux, это не имеет значения, за исключением случаев, когда внутри Linux оказываются два различных метода хранения записи о часовых поясах и, если они не совпадают, можно уладить несоответствие, например, при помощи нескольких дополнительных меток времени в файловых системах FAT. В листинге 55 показано, как использовать команду **date** для просмотра текущей даты и времени. Даже если аппаратные часы поддерживают время UTC, всегда отображается локальное время.

### Листинг 55. Отображение текущей даты и времени

```
[root@lyrebird ~]# date;date -u  
Mon Jul  9 22:40:01 EDT 2007
```

Команда **date** поддерживает большое разнообразие возможных форматов вывода, некоторые из которых вы уже видели в [листинге 28](#). Если вы хотите больше узнать о различных форматах данных, обратитесь к странице *man* для команды **date**.

Если необходимо настроить дату, сделать это можно, передав дату и время в качестве аргумента. Требуемый формат сложился автоматически и в некоторой степени необычен даже для американцев и действительно необычен для остальной части мира. Необходимо указать как минимум месяц, день, час и минуту в формате MMDhhmm, можно также добавить год в виде двух- или четырехзначного числа (CCYY или YY) и при желании точку и за ней двухзначное число для секунд. В листинге 56 показан пример, в котором системная дата изменена чуть больше чем на минуту.

### Листинг 56. Настройка системной даты и времени

```
[root@lyrebird ~]# date; date 0709221407;date  
Mon Jul  9 23:12:37 EDT 2007  
Mon Jul  9 22:14:00 EDT 2007  
Mon Jul  9 22:14:00 EDT 2007
```

## Настройка часов BIOS на временную зону UTC

Система Linux, как и большинство других современных операционных систем, фактически имеет двое часов. Первые часы — аппаратные, иногда называемые Real Time Clock, сокращенно (RTC), или часы BIOS, обычно они связаны с колеблющимся кварцевым кристаллом, имеющим точность хода до нескольких секунд в день. Точность зависит от различных колебаний, например, окружающей температуры. Вторые часы — внутренние программные часы, которые идут непрерывно, в том числе и при перерывах в работе системы. Они подвержены отклонениям, связанным с большой системной нагрузкой и задержкой прерываний. Однако система обычно считывает показания аппаратных часов при загрузке и потом использует системные часы. Команда **date**, о которой вы только что узнали, устанавливает не аппаратные, а системные часы.

Если используется NTP, можно установить аппаратные часы в ходе первой инсталляции системы и больше никогда не беспокоиться о них. Если нет, эта часть учебного пособия покажет, как просмотреть и установить время на аппаратных часах.

Для просмотра текущих показаний аппаратных часов можно воспользоваться командой **hwclock**. В листинге 57 показаны текущие показания обоих часов, и системных, и аппаратных.

### Листинг 57. Показания системных и аппаратных часов

```
[root@lyrebird ~]# date;hwclock  
Mon Jul  9 22:16:11 EDT 2007  
Mon 09 Jul 2007 11:14:49 PM EDT -0.071616 seconds
```

Обратите внимание, что значения различаются. Можно синхронизировать аппаратные часы с системными при помощи команды `hwclock` с опцией `-w` или `--systohc` и синхронизировать системные часы с аппаратными при помощи команды `hwclock` с опцией `-s` или `--hctosys`, как показано в листинге 58.

### Листинг 58. Настройка соответствия системных часов в аппаратным

```
[root@lyrebird ~]# date;hwclock;hwclock -s;date  
Mon Jul  9 22:20:23 EDT 2007  
Mon 09 Jul 2007 11:19:01 PM EDT -0.414881 seconds  
Mon Jul  9 23:19:02 EDT 2007
```

Можно указать опции `--utc` или `--localtime`, чтобы системные часы поддерживали UTC или местное время. Если значение не указано, оно берется из третьей строки файла `/etc/adjtime`.

Ядро Linux имеет режим, при котором каждые 11 минут системное время копируется в аппаратные часы. По умолчанию эта функция отключена, но она включается NTP. Запуск какой-либо команды, которая устанавливает время устаревшим способом, например, `hwclock --hctosys`, отключает ее, поэтому, если используется NTP, хорошей идеей будет просто позволить NTP выполнить эту работу. Чтобы узнать, как проверить, обновляются ли часы каждые 11 минут или нет, обратитесь к странице `man` команды `adjtimex`. Может возникнуть необходимость установить пакет `adjtimex`, если он не установлен по умолчанию.

Команда `hwclock` сохраняет изменения, сделанные в аппаратных часах, для того чтобы периодически компенсировать погрешность часов. Необходимые данные хранятся в `/etc/adjtime`, который является ASCII-файлом. Если NTP не используется, для компенсации отклонений часов можно использовать команду `adjtimex`. В противном случае аппаратные часы будут регулироваться приблизительно каждые 11 минут при помощи NTP. Кроме того, эта команда показывает, используют аппаратные часы местное время или время UTC, первое значение в `/etc/adjtime` показывает величину отклонения аппаратных часов за день (в секундах). В листинге 59 показаны два примера.

### Листинг 59. /etc/adjtime показывает отклонение часов и какое время они показывают, локальное или UTC

```
[root@lyrebird ~]# cat /etc/adjtime  
0.000990 1184019960 0.000000  
1184019960  
LOCAL  
root@pinguino:~# cat /etc/adjtime  
-0.003247 1182889954 0.000000  
1182889954  
LOCAL
```

Обратите внимание, что в обеих этих системах на аппаратных часах используется местное время, но отклонения часов отличаются — 0.000990 на lyrebird и -0.003247 на pinguino.

## Настройка часового пояса

Часовой пояс — критерий того, насколько местное время отличается от UTC. Информация о доступных часовых поясах, которые можно настроить, хранится в /usr/share/zoneinfo. По традиции /etc/localtime является линком на один из файлов часового пояса из этого дерева каталогов, например, /usr/share/zoneinfo/Eire или /usr/share/zoneinfo/Australia/Hobart. В современных системах с большой долей вероятности это будет копия соответствующего файла часового пояса, поскольку когда в процессе загрузки нужна информация о местном часовом поясе, файловая система /usr/share не может быть смонтирована.

Подобным образом другой файл, /etc/timezone, традиционно является линком /etc/default/init и используется для установки переменной окружения для часового пояса TZ и нескольких связанных с местоположением переменных окружения. Файл в системе может существовать, а может нет. Если он существует, он может содержать просто имя текущего часового пояса. Информацию о часовом поясе можно также найти в /etc/sysconfig/clock. В листинге 60 показаны эти файлы, взятые из систем Ubuntu 7.04 и Fedora 7.

### Листинг 60. Информация о часовых поясах из /etc

```
root@pinguino:~# cat /etc/timezone
America/New_York

[root@lyrebird ~]# cat /etc/sysconfig/clock
# The ZONE parameter is only evaluated by system-config-date.
# The timezone of the system is defined by the contents of /etc/localtime.
ZONE="America/New York"
UTC=false
ARC=false
```

В некоторых системах, например, Debian и Ubuntu, для установки часового пояса есть команда **tzconfig**. В других, например, в Fedora, для установки часового пояса и проверки, используют ли часы UTC или нет, используется команда **system-config-date**. Листинг 61 иллюстрирует использование команды **tzconfig** для просмотра текущего часового пояса.

### Листинг 61. Просмотр настроек часового пояса при помощи tzconfig

```
root@pinguino:~# tzconfig
Your current time zone is set to America/New_York
Do you want to change that? [n]:
Your time zone will not be changed
```

## Настройка Network Time Protocol

*Network Time Protocol (NTP)* — протокол для синхронизации часов компьютера по сети. Обычно проводится синхронизация с UTC.

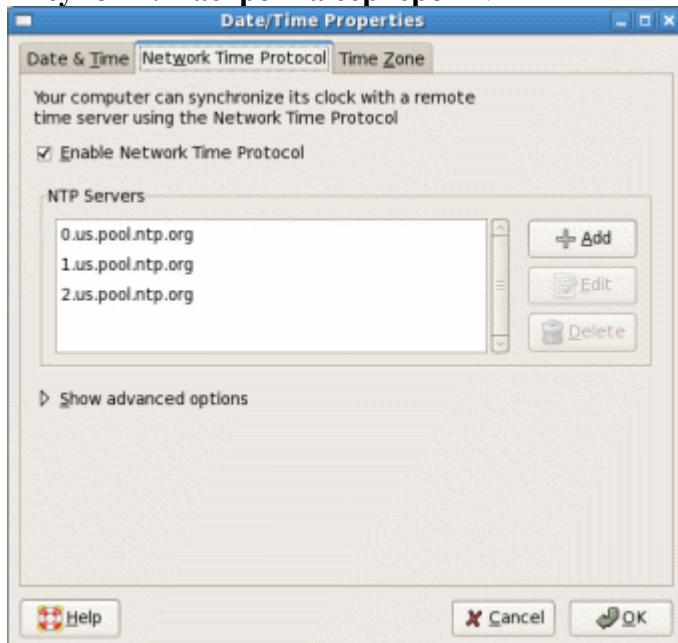
NTP версии 3 — Internet draft standard, описанный в RFC 1305. Текущая версия NTP версии 4, находящаяся в стадии разработки, еще не описана в RFC. RFC 4330 описывает Simple NTP (SNTP) версии 4.

Синхронизация времени достигается путем отправки сообщения *серверам времени*. Время возвращается отрегулированным при помощи смещения на половину времени задержки при прохождении туда и обратно. Поэтому точность времени зависит от задержки сети и от того, насколько задержка одинакова для обоих направлений. Чем короче путь до сервера времени, тем, вероятно, более точным будет время. Более подробную информацию, чем может предоставить это упрощенное описание, см. в разделе [Ресурсы](#).

В Интернете существует огромное количество компьютеров, поэтому серверы времени организованы в *страты*. Относительно маленький номер серверов страты 1 поддерживает очень точное время от источника, например, от атомных часов. Большой номер серверов страты 2 получает их время от сервисов страты 1 и делает его доступным для еще большего номера серверов страты 3 и так далее. Чтобы облегчить нагрузку на серверы времени, большое количество волонтеров отдает сервисы времени через pool.ntp.org (ссылку см. в разделе [Ресурсы](#)). Циклические (round robin) DNS-серверы достигают баланса нагрузки на NTP, распределяя запросы к NTP-серверу между группой доступных серверов.

Если используется графический интерфейс, можно настроить серверы времени NTP, используя диалог, подобный приведенному на рисунке 4. Тот факт, что эта система имеет возможность автоматически обновить время при помощи NTP является причиной того, что в диалоге, изображенном на рисунке 3, нельзя изменить дату и время.

**Рисунок 4. Настройка серверов NTP**



Информация о конфигурации NTP хранится в файле /etc/ntp.conf, так что можно отредактировать файл, сохранить его и затем перезапустить демон ntpd. В листинге 62 показан пример файла /etc/ntp.conf, использующего серверы времени из рисунка 4.

#### Листинг 62. Файл /etc/ntp.conf

```
[root@lyrebird ~]# cat /etc/ntp.conf
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default kod nomodify notrap nopeer noquery

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
```

```

restrict 127.0.0.1

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).

#broadcast 192.168.1.255 key 42      # broadcast server
#broadcastclient      # broadcast client
#broadcast 224.0.1.1 key 42          # multicast server
#multicastclient 224.0.1.1          # multicast client
#manyserver 239.255.254.254        # manycast server
#manyclient 239.255.254.254 key 42 # manycast client

# Undisciplined Local Clock. This is a fake driver intended for backup
# and when no outside source of synchronized time is available.
#server 127.127.1.0 # local clock
#fudge 127.127.1.0 stratum 10

# Drift file. Put this in a directory which the daemon can write to.
# No symbolic links allowed, either, since the daemon updates the file
# by creating a temporary in the same directory and then rename()'ing
# it to the file.
driftfile /var/lib/ntp/drift

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8
server 0.us.pool.ntp.org
restrict 0.us.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery
server 1.us.pool.ntp.org
restrict 1.us.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery
server 2.us.pool.ntp.org
restrict 2.us.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery

```

Если используются серверы времени pool.ntp.org, они могут размещаться в мире где угодно. Обычно вы будете получать лучшее время, накладывая на серверы ограничения, как в этом примере, где используется us.pool.ntp.org, в результате чего были выбраны только серверы, находящиеся в США. Ссылки на дополнительную информацию о проекте pool.ntp.org см. в разделе [Ресурсы](#).

## Команды NTP

Для настройки системного времени с сервера времени NTP можно использовать команду [ntpdate](#), как показано в листинге 63.

### Листинг 63. Настройка системного времени с сервера времени NTP при помощи ntpdate

```
[root@lyrebird ~]# ntpdate 0.us.pool.ntp.org
```

```
10 Jul 10:27:39 ntpdate[15308]: adjust time server 66.199.242.154 offset -0.007271 sec
```

Поскольку серверы работают в циклическом режиме, в следующий раз при запуске этой команды вы, возможно, увидите другой сервер. В листинге 64 показаны первые несколько DNS-ответов для 0.us.pool.ntp.org несколькими моментами позже запуска описанной выше команды **ntpdate**.

#### Листинг 64. Циклический NTP-сервер pool

```
[root@lyrebird ~]# dig 0.pool.ntp.org +noall +answer | head -n 5
0.pool.ntp.org. 1062 IN A 217.116.227.3
0.pool.ntp.org. 1062 IN A 24.215.0.24
0.pool.ntp.org. 1062 IN A 62.66.254.154
0.pool.ntp.org. 1062 IN A 76.168.30.201
0.pool.ntp.org. 1062 IN A 81.169.139.140
```

Команда **ntpdate** сейчас признана устаревшей, поскольку то же самое действие можно выполнить при помощи команды **ntpq** с опцией **-q**, как показано в листинге 65.

#### Листинг 65. Настройка системного времени при помощи ntpd -q

```
[root@lyrebird ~]# ntpd -q
ntpd: time slew -0.014406s
```

Обратите внимание, что команда **ntp** использует информацию о сервере времени из конфигурационного файла /etc/ntp.conf. Подробнее о команде **ntp** и ее опциях см. в страницах руководства man. Кроме того, следует осознавать, что если запущен демон ntpd, **ntp -q** завершит его исполнение, оставив в /var/log/messages сообщение о неудаче.

Другая связанная команда — **ntpq**, которая позволяет запросить демон NTP. Подробности см. в страницах руководства man.

Это учебное пособие подошло к концу. Мы рассмотрели материалы, касающиеся системного администрирования. Не забудьте [высказать мнение об этом учебном пособии](#).

### Ресурсы

#### Научиться

- [LPI exam 102 prep, Topic 111: Administrative tasks](#) — оригинал этого учебного пособия на сайте developerWorks (EN).
- Для изучения основ операционной системы Linux и подготовки к сертификации по системному администрированию ознакомьтесь со всей [серий учебных пособий для подготовки к экзаменам LPI](#) на сайте developerWorks.
- В [LPIC Program](#) вы найдете списки заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux (EN).
- Посмотрите [домашнюю страницу Partimage](#), чтобы узнать о Partimage, средстве для резервного копирования и восстановления раздела, готовом к работе с файловой системой (EN).

- "[/etc: Host-specific system configuration](#)" описывает требования Linux Standard Base (LSB) для /etc (EN).
- [Проект Network Time Protocol](#) описывает референтную реализацию протокола NTP и соответствующую документацию (EN).
- [Проект Network Time Synchronization](#) поддерживает широкий набор документации и справочной информации, в том числе презентации, касающиеся протоколов синхронизации времени (EN).
- [Проект pool.ntp.org](#) — большой виртуальный кластер серверов времени, которые стремятся обеспечить миллионам клиентов надежное и простое использование сервиса NTP, избегая чрезмерной нагрузки на наиболее популярные серверы времени (EN).
- Из статьи "[Basic tasks for new Linux developers](#)" (developerWorks, март 2005) узнайте, как открыть окно терминала или получить приглашение shell, а также о многом другом (EN).
- [Linux Documentation Project](#) содержит ряд полезных документов, главным образом HOWTO (EN).
- [LPI Linux Certification in a Nutshell, Second Edition](#) (O'Reilly, 2006) и [LPIC 1 Exam Cram 2: Linux Professional Institute Certification Exams 101 and 102 \(Exam Cram 2\)](#) (Que, 2004) предназначены для читателей, отдающих предпочтение книжному формату (EN).
- Найдите другие [учебные пособия для Linux-разработчиков](#) в разделе [Linux](#) сайта developerWorks.
- Регулярно посещайте раздел [технических мероприятий и Web-трансляций](#) developerWorks (EN).

## Получить продукты и технологии

- Постройте ваш следующий проект разработки в Linux при помощи [пробного программного обеспечения IBM](#), доступного непосредственно с developerWorks (EN).

# Экзамен LPI 201: Ядро Linux

*Администрирование, средний уровень (LPIC-2) тема 201*

[Дэвид Мерц](#), автор, Gnosis Software, Inc.

**Описание:** В этом учебном пособии Дэвид Мерц начинает готовить вас к сдаче экзамена 201 Linux Professional Institute® Администрирование, средний уровень (LPIC-2). Из этого первого из восьми пособий вы узнаете, как разбираться в текстах ядра Linux™, научитесь компилировать и настраивать ядро.

[Больше статей из этой серии](#)

**Дата:** 20.09.2005 (Опубликовано: 29.08.2005)

**Уровень сложности:** средний

## Прежде чем начать

Узнайте, чему может научить вас это учебное пособие и как извлечь из него максимум.

## Об этой серии учебных пособий

[Linux Professional Institute](#) (LPI) осуществляет сертификацию системных администраторов Linux по двум уровням: для начинающих и среднего уровня. Для достижения каждого уровня сертификации вы должны сдать два экзамена LPI.

Каждый экзамен охватывает несколько тем, каждая тема имеет свой рейтинг. Рейтинг показывает относительную важность каждой темы. Грубо говоря, чем выше рейтинг темы, тем больше она может содержать вопросов. Темы экзамена LPI 201 и их рейтинги:

### Тема 201

Ядро Linux (рейтинг 5). В фокусе этого учебного пособия.

### Тема 202

Запуск системы (рейтинг 5).

### Тема 203

Файловая система (рейтинг 10).

### Тема 204

Оборудование (рейтинг 8).

### Тема 209

Совместное использование файлов и служб (рейтинг 8).

### Тема 211

Поддержка системы (рейтинг 4).

### Тема 213

Настройка работ и автоматическое выполнение заданий (рейтинг 3).

### Тема 214

Устранение неполадок (рейтинг 6).

Linux Professional Institute не одобряет использование при подготовке к экзаменам любых учебных материалов или технологий, разработанных третьими лицами. За разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## Об этом учебном пособии

Добро пожаловать в учебное пособие "Ядро Linux", первое из восьми пособий, разработанных для подготовки к экзамену LPI 102. Из этого пособия вы узнаете, как компилировать и настраивать ядро Linux.

Это учебное пособие организовано в соответствии с рабочей программой LPI по этой теме:

### **2.201.1 Компоненты ядра (рейтинг 1)**

Вы узнаете, как использовать компоненты ядра, необходимые для специфического оборудования, драйверов оборудования, системных ресурсов и для удовлетворения тем или иным требованиям. Вы узнаете, как создавать различные типы образов ядра, определять, является ли ядро стабильным и разрабатывать ядро и патчи, а также о том, как использовать модули ядра.

### **2.201.2 Компиляция ядра (рейтинг 1)**

Вы узнаете, как правильно скомпилировать ядро, позволяющее при необходимости включать или отключать специальные компоненты. Вы научитесь осуществлять сборку и пересборку ядра Linux, осуществлять обновление и находить изменения в новом ядре, создавать образ `initrd` и устанавливать новое ядро.

### **2.201.3 Прикладывание патчей к ядру (рейтинг 2)**

Вы узнаете, как правильно делать для ядра патчи различных назначений, например, для обновления ядра, исправления ошибок и добавления поддержки нового оборудования. Также вы узнаете, как правильно удалять патчи из существующих ядер.

### **2.201.4 Настройка kernel (рейтинг 1)**

Вы узнаете, как настроить ядро в соответствии со специфическими системными требованиями путем прикладывания патчей, компиляции и редактирования конфигурационных файлов согласно требованиям. Вы узнаете, как определить, необходимо ли компилировать ядро или можно приложить патч, а также о том, как создавать и конфигурировать модули ядра.

Строго говоря, это учебное пособие -- одно из нескольких в этой серии, касающееся непосредственно Linux. Средства для работы с сетью, сопровождения системы, управления файлами и каталогами и так далее важны для работы с Linux и являются частью почти любого дистрибутива Linux. Но базовое ядро -- часть программного обеспечения, которая является связующим звеном между соперничающими друг с другом за аппаратные ресурсы программами и обеспечивает доступ к этим ресурсам -- это то программное обеспечение, которое разработано Линусом Торвальдсом и которое правильно называть "Linux как таковой".

Одно из лучших свойств ядра Linux заключается в том, что оно является свободно распространяемым. Множество выдающихся людей внесло свой вклад в улучшение ядра Linux, и вы, как системный администратор, имеете доступ к исходным кодам ядра. Это позволяет вам конфигурировать и настраивать ядро в соответствии с вашими потребностями.

#### **Необходимые условия**

Чтобы извлечь максимум из этого учебного пособия, вы должны иметь базовые знания о Linux и рабочую версию системы Linux, где вы сможете упражняться в выполнении команд, приведенных в этом пособии.

## **Экзамен LPI 201: Ядро Linux**

*Администрирование, средний уровень (LPIC-2) тема 201*

Дэвид Мерц, автор, Gnosis Software, Inc.

**Описание:** В этом учебном пособии Дэвид Мерц начинает готовить вас к сдаче экзамена 201 Linux Professional Institute® Администрирование, средний уровень (LPIC-2). Из этого первого из восьми пособий вы узнаете, как разбираться в текстах ядра Linux™, научитесь компилировать и настраивать ядро.

**Дата:** 20.09.2005 (Опубликовано: 29.08.2005)

**Уровень сложности:** средний

## Компоненты ядра

В этом разделе мы рассматриваем материал по теме 2.201.1 экзамена 201

Администрирование, средний уровень (LPIC-2). Рейтинг темы 1.

### Из чего состоит ядро?

В ядро Linux входит базовое ядро как таковое плюс некоторое количество модулей ядра. В большинстве случаев базовое ядро и большая коллекция модулей ядра, компилируемые одновременно и устанавливаемые или распространяемые вместе, основаны на коде, созданном Линусом Торвальдсом или измененном производителями дистрибутивов Linux. Базовое ядро всегда загружается в ходе загрузки системы и остается загруженным во время работы постоянно. Модули ядра первоначально могут быть загружены, а могут нет (хотя как правило часть из них загружена) и могут подгружаться или выгружаться во время работы.

Модульная структура ядра позволяет подключать дополнительные модули, скомпилированные позднее или отдельно от базового ядра. Дополнительные модули могут создаваться, когда вы добавляете оборудование к уже работающей системе Linux, а иногда могут поставляться третьими лицами. Модули ядра иногда распространяются в виде бинарных файлов, в результате чего ваши способности, как системного администратора, настраивать модули ядра оказываются не востребованы. В любом случае, загруженный модуль становится частью работающего ядра до тех пор, пока он не будет выгружен. Вопреки некоторым представлениям, модуль ядра не просто является интерфейсом прикладного программирования (API) для общения с базовым ядром, а становится частью работающего ядра.

### Соглашения о наименовании ядра

Ядра Linux следуют соглашениям о наименовании/нумерации, что позволяет быстро получить важную информацию о загруженном ядре. В соглашении определены обозначения для major номера, minor номера, редакция и в некоторых случаях включена строка, описывающая производителя/настройки. Эти соглашения применяются для нескольких типов файлов, в том числе к архивам исходников ядра, патчам и, возможно, нескольким базовым ядрам (если вы запускаете то одно, то другое ядро).

Как и обычная разделенная точками последовательность, ядро Linux следует соглашению по разделению стабильной и экспериментальной веток. Для стабильных веток используется четный minor номер, тогда как для экспериментальных веток -- нечетный minor номер. Редакция -- просто последовательная нумерация, отражающая исправления ошибок и перенос нововведений в старые версии ядра. Кроме того, номер часто характеризует производителя или специальные возможности. Например:

- `linux-2.4.37-foo.tar.gz`: обозначает архив исходников для стабильной ветки ядра 2.4 от компании "Foo Industries"
- `/boot/bzImage-2.7.5-smp`: обозначает собранное экспериментальное базовое ядро 2.7 с возможностью поддержки SMP
- `patch-2.6.21.bz2`: обозначает патч для обновления более ранней стабильной версии 2.6 до редакции 21

## Файлы ядра

Базовое ядро Linux может быть двух версий: *zImage*, ограниченной 508 Кбайт, и *bzImage* для более крупных ядер (приблизительно до 2.5 Мбайт). Как правило, современные дистрибутивы Linux используют ядро формата *bzImage*, что позволяет включать множество

компонент. Вы можете предположить, что так как "z" в zImage означает сжатие с помощью gzip, то "bz" в bzImage может означать сжатие с помощью bzip2. Однако, "b" просто обозначает "big", а для сжатия по-прежнему используется gzip. В обоих случаях в каталоге /boot/ базовое ядро часто переименовывается в vmlinuz. Как правило файл /vmlinuz является символьной ссылкой на файл с полным именем ядра, включающим номер версии, например, /boot/vmlinuz-2.6.10-5-386.

В каталоге /boot/ есть несколько файлов, связанных с базовым ядром, которые вам уже знакомы (иногда они могут располагаться вместо этого в корневом каталоге файловой системы). **System.map** -- это таблица, отображающая адреса символов ядра. **initrd.img** иногда используется базовым ядром для создания упрощенной файловой системы на ram-диске, подключаемом на этапе загрузки для монтирования основной файловой системы.

## Модули ядра

Модули ядра содержат дополнительный код ядра, который может быть загружен после базового ядра. Модули обычно предоставляют одну или несколько функций:

- **Драйверы устройств (Device drivers)**: поддержка специфических типов оборудования
- **Драйверы файловой системы (File system drivers)**: предоставляют необязательную возможность чтения и/или записи специфической файловой системы
- **Системные вызовы (System calls)**: большинство поддерживается базовым ядром, но модули могут добавлять или изменять системные службы
- **Сетевые драйверы (Network drivers)**: реализуют соответствующие сетевые протоколы
- **Загрузчики исполняемых файлов (Executable loaders)**: анализируют и загружают исполняемые дополнительные форматы

В этом разделе мы рассматриваем материал по теме 2.201.2 экзамена 201 Администрирование, средний уровень (LPIC-2). Рейтинг темы 1.

## Получение исходников ядра

Первое, что нужно сделать, чтобы скомпилировать новое ядро, -- получить его исходные коды. Основное хранилище исходников ядра -- Linux Kernel Archives ([kernel.org](http://kernel.org); см. [Ресурсы](#)). Производитель вашего дистрибутива мог включить в него обновленные исходники ядра, чтобы показать внесенные изменения. Например, вы можете получить и распаковать версию ядра с командами, подобными следующим:

### Листинг 1. Получение и распаковка ядра

```
% cd /tmp/src/  
% wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.12.tar.bz2  
% cd /usr/src/  
% tar jxvf /tmp/src/linux-2.6.12.tar.bz2
```

Для распаковывания исходников ядра в каталог /usr/src/ вам понадобятся права root'a. Однако, вы можете распаковать и собрать ядро в своем домашнем каталоге. Поиските на [kernel.org](http://kernel.org) другие форматы архивов и протоколов загрузки.

## Проверка исходников ядра

Если вы благополучно получили и распаковали архив с исходниками ядра, в вашей системе должен появиться каталог /usr/src/linux-2.6.12 (или, если вы распаковывали архив в другом месте, каталог с похожим названием). Важно, что тот каталог должен содержать файл

README с текущей информацией. В этом каталоге содержатся подкаталоги с исходными файлами, в основном это файлы .c и .h. Главные действия по объединению этих файлов в работающее ядро прописаны в файле **Makefile**, который использует утилита **make**.

## Конфигурирование сборки

После получения и распаковывания исходников ядра у вас может появиться желание сконфигурировать ядро. Команда **make** имеет три опции для настройки опций ядра: **config**, **menuconfig** и **xconfig**. Вообще, можно отредактировать файл **.config**, но делать это нежелательно (вы откажетесь от дополнительной информации и можете легко создать нерабочую конфигурацию).

Командой **make config** воспользоваться также непросто, как и отредактировать файл **.config** вручную; вам придется настраивать каждую опцию (их сотни) в определенном порядке, без возможности возврата к предыдущему действию. Команда **make menuconfig** предоставляет curses интерфейс, где вы можете выбрать только те опции, которые необходимо изменить. Команда **make xconfig** предоставляет симпатичный графический интерфейс (особенно красивый в Linux 2.6+).

Для значительной части опций ядра возможны три варианта выбора: (1) включить компоненту в базовое ядро (include the capability in the base kernel); (2) включить как модуль ядра (include it as a kernel module); (3) совсем не включать компоненту (omit the capability entirely). Вообще, в создании многочисленных модулей ядра нет ничего страшного (кроме незначительного увеличения времени компиляции), поскольку пока они не нужны, они и не загружены. Если дисковое пространство ограничено, можно не включать никаких возможностей.

## Запуск процесса сборки

Теперь, чтобы собрать базовое ядро с выбранными опциями, следует выполнить следующие действия:

- **make dep**: необходимо только для ядра 2.4, для 2.6 не требуется.
- **make clean**: очистить предыдущие объектные файлы, это особенно полезно, если вы собираете данное ядро не первый раз.
- **make bzImage**: создать базовое ядро. В особых случаях для небольших образов ядра можно использовать **make zImage**. Вы также можете воспользоваться командой **make zlilo**, чтобы установить ядро прямо в загрузчик lilo, или командой **make zdisk**, чтобы создать загрузочную дискету. Вообще, лучше создавать образ ядра в каталоге типа **/usr/src/linux/arch/i386/boot/vmlinuz**, используя команду **make bzImage**, и затем копировать его оттуда вручную.
- **make modules**: создать все сконфигурированные загружаемые модули ядра.
- **sudo make modules\_install**: установить все собранные модули в каталог **/lib/modules/2.6.12/**, название подкаталога совпадает с номером версии ядра.

## Создание стартового ram-диска

Если вы создали важный загрузочный драйвер, стартовый ram-диск позволит загрузить его в процессе начальной загрузки. Это касается главным образом тех драйверов файловой системы, которые были собраны в виде модулей ядра. По существу, стартовый ram-диск -- некий магический корневой псевдо-раздел, который живет в памяти и позже выполняет **chroot** на реальный раздел диска (например, если ваш корневой раздел расположен на RAID). Более подробное описание вы найдете в следующих учебных пособиях этой серии.

Создание стартового ram-диска осуществляется при помощи команды **mkinitrd**. Чтобы узнать, какие опции имеет команда **mkinitrd**, включенная в ваш дистрибутив Linux, обратитесь к странице man этой команды. Самое простое -- запустить команду, подобную

следующей:

## Листинг 2. Создание ram-диска

```
% mkinitrd /boot/initrd-2.6.12 2.6.12
```

## Инсталляция собранного ядра Linux

Успешно собрав базовое ядро и связанные с ним модули (это займет какое-то время -- на медленных машинах до нескольких часов), вы должны скопировать образы ядра (`vmlinuz` или `bzImage`) в свой каталог `/boot/`.

После того как вы скопировали необходимые файлы в `/boot/` и установили модули ядра при помощи `make modules_install`, необходимо сконфигурировать загрузчик, обычно это `lilo` или `grub`, для доступа к соответствующему ядру (ядрам). Информацию о конфигурировании `lilo` и `grub` вы найдете в следующем учебном пособии этой серии.

## Дополнительная информация

На сайте [kernel.org](http://kernel.org) есть много полезных ссылок, по которым можно получить дополнительную информацию о компонентах ядра и требованиях для сборки. Чрезвычайно полезная и подробная информация содержится в руководстве *Kernel Rebuild Guide* Квана Лоу (Kwan Lowe). Ссылки на эти ресурсы вы найдете в разделе [Ресурсы](#).

## Приложение патчей к ядру

В этом разделе мы рассматриваем материал по теме 2.201.3 экзамена 201 Администрирование, средний уровень (LPIC-2). Рейтинг темы 2.

## Получение патчей

Исходники ядра Linux распространяются в виде дерева основных исходников в сочетании с множеством небольших патчей. Обычно это позволяет получить самое свежее ядро через максимально быстрые каналы. Это соглашение позволяет прикладывать специальные патчи, полученные не с [kernel.org](http://kernel.org), а из других источников.

Если вы хотите применить несколько уровней изменений, вам необходимо получить всю серию патчей последовательно (по возрастанию). Например, предположим, что к моменту чтения этого пособия доступно ядро 2.6.14 и вы загрузили ядро 2.6.12. Вы должны сделать следующее:

## Листинг 3. Последовательное получение патчей

```
% wget http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.13.bz2  
% wget http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.14.bz2
```

## Распаковывание и применение патчей

Чтобы применить патч, необходимо сначала распаковать архив при помощи `bzip2` или `gzip`, в зависимости от формата сжатия архива, а затем приложить патч. Например:

## Листинг 4. Распаковывание и применение патчей

```
% bzip2 -d patch2.6.13.bz2  
% bzip2 -d patch2.6.14.bz2
```

```
% cd /usr/src/linux-2.6.12  
% patch -p1 < /path/to/patch2.6.13  
% patch -p1 < /path/to/patch2.6.14
```

Применив патчи, продолжите компиляцию, как описано в предыдущем разделе. Команда **make clean** удалит дополнительные объектные файлы, которые, возможно, не соответствуют новым изменениям.

## Настройка ядра

В этом разделе мы рассматриваем материал по теме 2.201.4 экзамена 201 Администрирование, средний уровень (LPIC-2). Рейтинг темы 1.

### О настройке

Настройка ядра описана в разделе этого пособия, рассказывающем о сборке ядра (точнее, в опциях **make [x|menu]config**). Когда базовое ядро и его модули собраны, вы можете включить или отменить возможности ядра в порядке подключения дополнительных возможностей, запуска различных профилей и подключения памяти.

В этом разделе рассматриваются способы изменения поведения ядра в ходе работы системы.

### Поиск информации о загруженном ядре

Linux (и другие UNIX-подобные операционные системы) использует специальные, как правило совместимые способы хранения информации о загруженном ядре (или других запущенных процессах). Специальный каталог **/proc/** содержит псевдо-файлы и подкаталоги, содержащие массу информации о загруженной системе.

В ходе работы системы Linux каждый процесс создает подкаталог со своим номером, каждый из которых содержит несколько статусных файлов. Здесь хранятся сводные данные о командах пользовательских уровней и системных средствах, но основная часть данных расположена в каталоге файловой системы **/proc/**.

Специфические данные, касающиеся статуса самого ядра, находятся в каталоге **/proc/sys/kernel**.

### Подробнее о текущих процессах

Хотя статус процессов, особенно пользовательских, не имеет отношения к ядру *как таковому*, важно иметь о них представление, если вы намерены заниматься отладкой основного ядра. Простейший способ получить сводку процессов -- выполнить команду **ps** (также существуют графические средства). Зная ID процесса, вы можете исследовать запущенный процесс. Например:

#### Листинг 5. Исследование запущенного процесса

```
% ps  
 PID TTY          TIME CMD  
16961 pts/2    00:00:00 bash  
17239 pts/2    00:00:00 ps  
% ls /proc/16961  
binfmt  cwd@  exe@  maps  mounts  stat   status  
cmdline  environ  fd/  mem    root@  statm
```

В этом пособии не будет исследоваться вся информация, содержащаяся в псевдо-файлах процессов. Для примера приведен фрагмент файла **status**:

### Листинг 6. Фрагмент псевдо-файла status

```
$ head -12 /proc/17268/status
Name: bash
State: S (sleeping)
Tgid: 17268
Pid: 17268
PPid: 17266
TracerPid: 0
Uid: 0 0 0 0
Gid: 0 0 0 0
FDSize: 256
Groups: 0
VmSize: 2640 kB
VmLck: 0 kB
```

### Процессы ядра

Каталог `/proc/` наряду с данными о пользовательских процессах содержит полезную информацию о загруженном ядре. Особенно важен каталог `/proc/sys/kernel/`:

### Листинг 7. Каталог `/proc/sys/kernel/`

```
% ls /proc/sys/kernel/
acct          domainname  msgmni      printk       shmall    threads-max
cad_pid       hostname   osrelease   random/     shmmmax   version
cap-bound     hotplug    ostype      real-root-dev shmmnmi
core_pattern  modprobe   overflowgid rtsig-max  swsusp
core_uses_pid msgmax    overflowuid rtsig-nr   sysrq
ctrl-alt-del msgmnb    panic       sem        tainted
```

Содержимое этих псевдо-файлов отображает информацию о загруженном ядре. Например:

### Листинг 8. Просмотр псевдо-файла `ostype`

```
% cat /proc/sys/kernel/ostype
Linux
% cat /proc/sys/kernel/threads-max
4095
```

### Уже загруженные модули ядра

Как и другая информация о запущенной системе Linux, данные о загруженном ядре хранятся в каталоге файловой системы `/proc/`, точнее в каталоге `/proc/modules`. Однако, как правило, доступ к этим данным можно получить при помощи утилиты `lsmod` (которая просто показывает заголовки необработанного содержимого файла `/proc/modules`); команда `cat /proc/modules` выведет такую же информацию. Рассмотрим пример:

### Листинг 9. Содержимое файла `/proc/modules`

```
% lsmod
```

Module	Size	Used by	Not tainted
lp	8096	0	
parport_pc	25096	1	
parport	34176	1 [lp parport_pc]	
sg	34636	0 (autoclean) (unused)	
st	29488	0 (autoclean) (unused)	
sr_mod	16920	0 (autoclean) (unused)	
sd_mod	13100	0 (autoclean) (unused)	
scsi_mod	103284	4 (autoclean) [sg st sr_mod sd_mod]	
ide-cd	33856	0 (autoclean)	
cdrom	31648	0 (autoclean) [sr_mod ide-cd]	
nfsd	74256	8 (autoclean)	
af_packet	14952	1 (autoclean)	
ip_vs	83192	0 (autoclean)	
floppy	55132	0	
8139too	17160	1 (autoclean)	
mii	3832	0 (autoclean) [8139too]	
supermount	15296	2 (autoclean)	
usb-uhci	24652	0 (unused)	
usbcore	72992	1 [usb-uhci]	
rtc	8060	0 (autoclean)	
ext3	59916	2	
jbd	38972	2 [ext3]	

## Загрузка дополнительных модулей ядра

Для загрузки модулей ядра есть два инструмента. Команда `modprobe` -- немного более высокого уровня. Она регулирует загрузку взаимозависимых модулей, то есть при загрузке модулей ядра загружаются и те, модули, от которых они зависят. Однако, `modprobe` на самом деле только надстройка над `insmod`.

Например, предположим, вы хотите загрузить в ядро возможность поддержки Reiser file system (надеюсь, она еще не встроена в ядро). Вы можете использовать опцию `modprobe -nv`, чтобы просто посмотреть, что сделала бы команда, но на самом деле ничего не загружать:

### Листинг 10. Проверка зависимостей при помощи modprobe

```
% modprobe -nv reiserfs
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/fs/reiserfs/reiserfs.o.gz
```

В нашем случае зависимостей нет. В других случаях они могут быть (и их придется урегулировать вручную, если использовать команду `modprobe` без опции `-n`). Пример:

### Листинг 11. Еще один вывод команды modprobe

```
% modprobe -nv snd-emux-synth
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/drivers/sound/
    soundcore.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/
    snd.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/synth/
    snd-util-mem.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/seq/
    snd-seq-device.o.gz
```

```
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/
    snd-timer.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/seq/
    snd-seq.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/seq/
    snd-seq-midi-event.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/
    snd-rawmidi.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/seq/
    snd-seq-virmidi.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/core/seq/
    snd-seq-midi-emul.o.gz
/sbin/insmod /lib/modules/2.4.21-0.13mdk/kernel/sound/synth/emux/
    snd-emux-synth.o.gz
```

Предположим, вы хотите загрузить модуль ядра сейчас. Вы можете воспользоваться командой **modprobe**, которая попутно загрузит и все зависимости, но чтобы увидеть все в подробностях, используйте команду **insmod**.

На основании вышеизложенного вы могли сделать предположение, что надо запустить, например, **insmod snd-emux-synth**. Но если сделать это без предварительной загрузки зависимостей, вы получите сообщение о "неразрешенных символах" ("unresolved symbols"). Давайте попробуем вместо этого использовать Reiser file system:

### Листинг 12. Загрузка модуля ядра

```
% insmod reiserfs
Using /lib/modules/2.4.21-0.13mdk/kernel/fs/reiserfs/reiserfs.o.gz
```

Довольно успешно, ваше ядро теперь поддерживает новую файловую систему. Вы можете монтировать разделы, читать их, производить в них запись и так далее. Другие возможности системы могут быть подключены подобным образом.

### Удаление загруженных модулей ядра

Выгрузка модулей, как и их загрузка, может быть выполнена при помощи высокогоуровневой команды **modprobe** или низкоуровневой **rmmod**. Инструмент высокого уровня осуществляет выгрузку в обратном порядке. Команда **rmmod** просто удаляет отдельный модуль ядра, но может не справиться с этой задачей, если модуль используется (обычно из-за зависимостей). Например:

### Листинг 13. Попытка выгрузки модулей, имеющих зависимости

```
% modprobe snd-emux-synth
% rmmod soundcore
soundcore: Device or resource busy
% modprobe -rv snd-emux-synth
# delete snd-emux-synth
# delete snd-seq-midi-emul
# delete snd-seq-virmidi
# delete snd-rawmidi
# delete snd-seq-midi-event
# delete snd-seq
# delete snd-timer
# delete snd-seq-device
```

```
# delete snd-util-mem  
# delete snd  
# delete soundcore
```

Однако, если ничто не препятствует удалению модуля, команда `rmmod` выгрузит его из памяти. Например:

#### Листинг 14. Выгрузка модулей, не имеющих зависимостей

```
% rmmod -v reiserfs  
Checking reiserfs for persistent data
```

### Автоматическая загрузка модулей ядра

При желании вы можете заставить модули ядра загружаться автоматически, воспользовавшись загрузчиком модулей ядра, входящим в последние версии Linux, или демоном `kerneld` в более старых версиях. Если вы используете эти приемы, в случае, если ядро обнаружит, что к нему обращаются через неподдерживаемые специфические системные вызовы, оно попытается загрузить соответствующий модуль ядра.

Однако, за исключением систем с недостаточным количеством памяти, обычно нет причин отказываться от загрузки необходимых модулей ядра в процессе загрузки системы (подробнее об этом рассказывается в следующем учебном пособии). Загрузчик модулей ядра включен в некоторые дистрибутивы.

### Автоматическая выгрузка модулей ядра

Наряду с автоматической загрузкой возможна и автоматическая выгрузка модулей, в основном в системах с недостаточным количеством памяти, например, встроенных системах Linux. Однако, следует знать, что модули ядра могут быть загружены с опцией `insmod --autoclean`, которая помечает их как незагруженные, если они не используются в данное время.

Раньше демон `kerneld` периодически вызывал команду `rmmod --all` для удаления неиспользуемых модулей ядра. В особых случаях (если не используется демон `kerneld`, не входящий в свежие системы Linux), можно добавить в `crontab` команду `rmmod --all`, запускаемую примерно раз в минуту. Но обычно это бывает лишним, поскольку модули ядра как правило используют намного меньше памяти, чем типичные пользовательские процессы.

## Ресурсы

### Научиться

- [Оригинал этого учебного пособия](#) на developerWorks.
- В [Программе LPIC](#) вы найдете список заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.
- Прочтите [Kernel Rebuild Guide](#) Квана Лоу (Kwan Lowe), чтобы больше узнать о сборке ядра.
- Ссылки на другие ресурсы для разработчиков Linux вы найдете в [developerWorks Linux zone](#).

## **Получить продукты и технологии**

- Получите исходники Linux с [kernel.org](http://kernel.org), Linux Kernel Archives.
- [Закажите SEK для Linux](#), набор из двух DVD, содержащих trial-версии последнего программного обеспечения IBM для Linux от DB2®, Lotus®, Rational®, Tivoli® и WebSphere®.
- Постройте ваш следующий проект разработки для Linux с использованием [IBM trial software](#), загрузив его непосредственно с developerWorks.

## **Обсудить**

- [Примите участие в обсуждении материала на форуме](#).
- [KernelNewbies.org](#) содержит множество средств для тех, кто плохо знаком с ядром: FAQ, канал IRC, список рассылки и wiki.
- [KernelTrap](#) -- веб-сообщество, которое уделяет большое внимание распространению новостей, связанных с разработкой ядра.
- В [Kernel Traffic](#) вы можете найти бюллетень, содержащий некоторые дискуссии из списка рассылки по ядру Linux.
- Читайте [блоги developerWorks](#) и вливайтесь в сообщество developerWorks.
- Участвуйте в русскоязычных [форумах ОС Linux](#).

# Учебник для экзамена LPI 201: Запуск системы

*Администрирование, средний уровень (LPIC-2) тема 202*

Дэвид Мерц, автор, Gnosis Software, Inc.

**Описание:** Это второй из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите этапы, которые система Linux™ проходит в процессе запуска, и их настройку.

[Больше статей из этой серии](#)

**Дата:** 31.08.2005

**Уровень сложности:** средний

## Перед тем как начать

Узнайте, чему эти обучающие программы могут научить вас, и как извлечь из них больше пользы.

## Об этой серии учебных пособий

Linux Professional Institute (LPI) производит сертификацию системных администраторов Linux двух уровней: для начинающих и среднего уровня. Чтобы получить сертификат каждого уровня, вы должны сдать два экзамена LPI.

Каждый из экзаменов состоит из нескольких тем, и каждая тема имеет свой рейтинг. Рейтинги указывают относительную важность каждой темы. Грубо говоря, вы вправе ожидать больше вопросов на экзамене на темы с более высоким рейтингом. Темы и их рейтинги для экзамена LPI 201:

### Тема 201

Ядро Linux (рейтинг 5).

### Тема 202

Запуск системы (рейтинг 5). Тема данной главы.

### Тема 203

Файловая система (рейтинг 10).

### Тема 204

Оборудование (рейтинг 8).

### Тема 209

Совместное использование файлов и служб (рейтинг 8).

### Тема 211

Поддержка системы (рейтинг 4).

### Тема 213

Настройка работ и автоматическое выполнение заданий (рейтинг 3).

### Тема 214

Устранение неполадок (рейтинг 6).

Linux Professional Institute® не приветствует использование для подготовки к экзаменам материалов и технологий от третьих лиц. Для более подробной информации обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## **Об этом руководстве**

Добро пожаловать в "Запуск системы", второй из восьми учебников, разработанных для подготовки к экзамену LPI 201. С его помощью вы изучите этапы, которые система Linux проходит в процессе инициализации и как изменить и настроить их поведение для ваших конкретных нужд.

Это руководство организовано по следующим разделам, которые LPI относит к данной теме:

### **2.201.1 Запуск системы и процессы загрузки (рейтинг 2)**

Вы научитесь редактировать соответствующие скрипты запуска системы, чтобы настраивать стандартные уровни запуска и сам процесс загрузки. Также рассказывается о работе с уровнями запуска и о создании специального образа `initrd`.

### **2.201.2 Восстановление файловой системы (рейтинг 3)**

Вы научитесь должным образом управлять системой Linux в процессе загрузки и в режиме восстановления, пользуясь утилитой `init` и опцией `init= kernel`.

Эта тема находится, строго говоря, на границе Linux. [Предыдущее пособие \(тема 201\)](#) посвящено ядру – сердцу Linux. Данное пособие касается работы вспомогательных инструментов и скриптов, которые нужны для работы ядра и подготовки системы для правильной работы. Такие скрипты и инструменты, связанные с инициализацией, не являются частью ядра, а поддерживаются создателями дистрибутивов Linux или отдельными системными администраторами. Впрочем, любая Linux система, даже самая простая, требует ряда начальных действий. Здесь мы произведем обзор таких шагов.

В следующих пособиях мы рассмотрим разнообразные инструменты для организации сети, обслуживания системы, управление файлами и данными и так далее, которые важны для рабочей установки Linux и являются частью почти каждого дистрибутива Linux.

## **Требования**

Чтобы работа с этим учебником была максимально плодотворна, вы уже должны быть достаточно хорошо знакомы с ОС Linux и иметь компьютер с ОС Linux, чтобы на практике иметь возможность самостоятельно проверять работу команд, описанных в этом руководстве.

# **Учебник для экзамена LPI 201: Запуск системы**

*Администрирование, средний уровень (LPIC-2) тема 202*

[Дэвид Мерц](#), автор, Gnosis Software, Inc.

**Описание:** Это второй из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите этапы, которые система Linux™ проходит в процессе запуска, и их настройку.

[Больше статей из этой серии](#)

**Дата:** 31.08.2005

**Уровень сложности:** средний

## **Запуск системы и процессы загрузки**

### **Что происходит, когда включается компьютер с ОС Linux?**

Давайте разделим процесс загрузки ОС Linux на девять этапов, которые имеют место

практически для любой конфигурации ОС Linux:

1. Первый этап загрузки -- это считывание BOIS'ом компьютера или другими программно-аппаратными средствами MBR жесткого диска или другого загрузочного устройства (например, компакт-диска, гибкого диска или сетевого загрузочного устройства, etc.).
2. Начинается работа загрузчика. Linux на архитектуре x86 обычно использует LILO или GRUB. Некоторые старые системы могут использовать loadlin чтобы загрузиться через вспомогательный DOS-раздел. В системах Power PC® это может быть BootX или yaboot. Вообще, загрузчик -- это простая программа, которая, тем не менее, знает, где искать ядро Linux, может выбрать, какую загружать из нескольких версий ядра или даже выбрать другую операционную систему на той же машине.
3. Загрузка ядра Linux.
4. Монтируется корневая файловая система. В некоторых случаях, временно монтируется начальная корневая файловая система из содержимого, например, RAM-диска, инициализируемого загрузчиком, чтобы дать возможность загружаться специальным драйверам и модулям, которые могут понадобиться для работы настоящей корневой файловой системы.

Теперь у нас есть корневая файловая система, и мы можем начать собственно инициализацию.

5. Запускается процесс `init`, прародитель всех остальных процессов в ОС Linux.
6. Считывается содержание файла `/etc/inittab`, чтобы определиться с дальнейшим ходом загрузки. Особено важно что прописано в файле `/etc/inittab` в строке, определяющей уровень запуска системы (и, следовательно, последующие этапы загрузки).

Действительно, все происходящее после этого момента полностью определяется содержимым файла `/etc/inittab`. Фактически, скрипты и другие инструменты, которые работают, подчиняются соответствующим настройкам, но, в принципе, вы могли бы полностью изменить `/etc/inittab`, чтобы управлять работой различных инструментов по вашему желанию.

Одна из установок в файле `/etc/inittab` особенно важна. Это строка, похожая на:

**`id:5:initdefault:`**

Обычно она находится ближе к началу файла и устанавливает уровень запуска системы. Уровень запуска определяет, какие действия будут предприняты в оставшихся предписаниях файла `/etc/inittab`.

Что происходит, когда сценарий `/etc/inittab` отработан? И особенно, какие именно файлы и директории принимают участие в процессе?

7. Инициализация, независимая от уровня запуска. Существуют ряд действий, которые будут выполняться независимо от установленного уровня запуска. Эти шаги обозначены в `/etc/inittab` строками, похожими на:

```
# System initialization.  
si::sysinit:/etc/rc.d/rc.sysinit
```

В некоторых системах Linux (в основном в системах на основе Debian), вы скорее увидите строчки, более похожие на следующие:

```
si::sysinit:/etc/init.d/rcS
```

В последнем случае файл `/etc/init.d/rcS` -- это просто скрипт, который по очереди запускает скрипты `/etc/rcS.d/[Ss]??*`. С другой стороны, если в вашей системе

используется `/etc/rc.d/rc.sysinit`, для выполнения инициализации достаточно одного длинного скрипта, содержащегося в этом файле.

8. Инициализация, зависимая от уровня запуска. Фактически, вы можете определить столько действий, связанных с уровнем запуска, сколько захотите, и при этом каждое действие может относиться к одному или нескольким уровням запуска. Как правило, `/etc/inittab` будет содержать строки типа:

```
l0:0:wait:/etc/rc.d/rc 0
# ...
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

В свою очередь, скрипт `/etc/rc.d/rc` будет управлять всеми файлами, названными `/etc/rc$1.d/[KkSs]??*`. В следующем примере можно увидеть, что в данной системе, стартующей с уровнем запуска 5, будут выполняться (по порядку):

```
/etc/rc5.d/K15postgresql
/etc/rc5.d/S01switchprofile
/etc/rc5.d/S05harddrake
...
/etc/rc5.d/S55sshd
...
/etc/rc5.d/S99linuxconf
/etc/rc5.d/S99local
```

Файлы, начинающиеся с "K" или "k" являются *убивающими (kill) скриптами*, они завершают процессы или упорядочивают их действия (последствия). Файлы, которые начинаются с "S" или "s" -- это *запускающие (startup) скрипты*, они начинают новые процессы или подготавливают систему к работе с этим уровнем запуска. Большинство из них являются скриптами shell, и большая часть их будет ссылками (часто на `/etc/init.d/`).

В то время когда система Linux стартует с определенным уровнем запуска, вы хотите зарегистрироваться в системе как пользователь. Чтобы авторизация прошла успешно, используется программа `getty`. Множество разновидностей программ на основе `getty` используется создателями дистрибутивов, типа `agetty`, `mgetty`, и `mingetty`. Но все они делают примерно то же.

9. Войдите в систему в приглашении. Уже знакомый нам `/etc/inittab` обычно запускает `getty` на одном или нескольких виртуальных экранах и делает это для нескольких уровней запуска. Уровни определены в строках типа:

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Цифра в начале показывает, в каком виртуальном терминале будет работать программа `getty`; следующие несколько цифр -- это те уровни запуска, при которых это случится (например, запуск `mingetty` при каждом из уровней 2, 3, 4 и 5).

Следующие шаги инициируют запуск дополнительных служб, вход в графическое

окружение, восстановление настроек пользовательского интерфейса или других более персонифицированных деталей, которые находятся вне рамок этого учебного пособия.

## Понятие уровня запуска

Понятие уровня запуска несколько произвольно, по крайней мере, оно не прописано в ядро Linux. Действительные уровни запуска сопоставлены набору номеров, чтобы можно было установить (или изменить имеющийся) по умолчанию уровень запуска выбором номера от 0 до 6. В соответствии с соглашением, следующий смысл присваивается каждому номеру уровня запуска:

### Листинг 1. Уровни запуска

```
# Default runlevel. The runlevels used by Mandrake Linux are:  
#   0 - Halt (Do NOT set initdefault to this)  
#   1 - Single user mode  
#   2 - Multiuser, without NFS (The same as 3, if you don't have networking)  
#   3 - Full multiuser mode  
#   4 - Unused  
#   5 - X11  
#   6 - Reboot (Do NOT set initdefault to this)
```

Это соглашение, как можно видеть, используется в дистрибутиве Mandrake Linux, но большинство дистрибутивов используют то же самое соглашение. Может так оказаться, что текстовые или встроенные дистрибутивы не используют некоторые из уровней запуска, но все равно ими будут зарезервированы эти же номера.

## Конфигурационные строки в /etc/inittab

Вы видели множество строчек из файла /etc/inittab в примерах, но что же конкретно они означают? Каждая строка имеет формат:

**id:runlevels:action:process**

Поле **id** это короткое сокращение, обозначающее конфигурационную строчку. (1 - 4 буквы в свежих версиях **init**; 1 - 2 в более старых). Поле **runlevels** уже обсуждалось. Следующее поле **action** обозначает действие, предпринимаемое строкой. Некоторые действия могут быть "специальными," такие как:

**ca::ctrlaltdel:/sbin/shutdown -t3 -r now**

Эта строка устанавливает действие для последовательности клавиш Ctrl-Alt-Delete (независимо от уровня запуска). Но большинство действий просто запускает соответствующие процессы. Частичный список действий включает:

- **respawn**: Процесс будет перезапущен всякий раз, когда завершится (как в случае с **getty**).
- **wait**: Процесс будет начат однажды, когда будет введен указанный уровень запуска, и **init** будет ждать его завершения.
- **once**: Процесс будет выполнен однажды, когда будет введен указанный уровень запуска.
- **boot**: Процесс будет выполнен во время загрузки системы (но после **sysinit**). Уровень запуска не имеет значения.

## Что такое загрузчик?

Несколько лет назад для загрузки Linux на x86 системах в основном использовалась программа, названная LILO. Название LILO -- сокращение от "LInux LOader." Сейчас более популярна программа, названная GRUB (GRand Unified Bootloader). На системах, отличных от x86, используются другие загрузчики, но все они сконфигурированы аналогичным для LILO и GRUB способом.

Хотя существуют различия в их конфигурационном синтаксисе, и LILO и GRUB выполняют в значительной степени одну и ту же задачу. По существу, каждый из них предоставляет выбор операционной системы (включая, возможно, несколько ядер Linux) и загружает ядро выбранной ОС в память компьютера. Обе программы позволяют вам передавать аргументы ядру Linux по ходу, и обе могут быть сконфигурированы с возможностью загрузки на том же компьютере ОС, отличных от Linux.

Либо LILO, либо GRUB (или какой-то другой загрузчик) находится на MBR (Master Boot Record) первичного жесткого диска, который автоматически загружается системным BIOS. LILO имеет ограничения на загрузку специального raw сектора жесткого диска. Загрузчик GRUB более изощрен и распознает разные файловые системы, например, такие как ext2/3, ReiserFS, VFAT или UFS. Это означает, что GRUB не нужно перезаписывать MBR каждый раз, как только изменился конфигурационный файл (как это делает LILO).

## Настройка загрузчика LILO

Настройка загрузчика LILO производится при помощи содержимого файла /etc/lilo.conf. Для более детального изучения параметров настройки LILO, прочитайте страницы помощи man для lilo.conf. Общий характер поведения определяют несколько начальных параметров.

Например, вы наверняка увидите **boot=/dev/hda** или нечто подобное. Эта команда устанавливает загрузчик на MBR первичного жесткого диска IDE. Вы можете также установить LILO внутрь конкретного раздела, обычно это нужно, когда вы используете другой основной загрузчик. Например, **boot=/dev/sda3** устанавливает LILO на третий раздел первого SCSI диска. Другие параметры определяют внешний вид и время ожидания LILO.

Запомните, что после того, как вы внесете исправления в файл /etc/lilo.conf, вам необходимо запустить LILO для фактической установки нового загрузочного сектора, который используется во время загрузки. Можно легко забыть установить новые параметры, но загрузчик сам по себе не сможет прочесть новую конфигурацию, за исключением того случая, когда записаны фактические адреса секторов (которые LILO распознает в процессе работы).

Если используется LILO, особенное значение имеют строки типа **image=** и, может быть, **other=**, если имеется выбор между ОС Linux и другими операционными системами. Пример /etc/lilo.conf может содержать:

## Листинг 2. Пример конфигурации LILO

```
image=/boot/bzImage-2.7.4
label="experimental"
image=/boot/vmlinuz
label="linux"
initrd=/boot/initrd.img
append="devfs=mount acpi=off quiet"
vga=788
read-only
other=/dev/hda3
```

```
label=dos
```

Такая конфигурация позволяет вам выбирать либо ядро версии 2.7.4, которое находится в стадии разработки, либо стабильное ядро (далее объявлено, что использовать как стартовый RAM-диск (initrd) в процессе загрузки). Вы можете также выбрать DOS, которая находится на третьем разделе первичного IDE диска.

## Настройка загрузчика GRUB

Бесспорным преимуществом GRUB является то, что его не надо переустанавливать всякий раз, после того как вы изменили параметры загрузки. Конечно, в первый раз GRUB все же нужно установить, обычно это делается командой типа **grub-install /dev/hda**. Как правило, в процессе инсталляции дистрибутив делает это для вас сам, так что вы, может статься, так ни разу сами этого и не сделаете.

Теперь, так как GRUB знает, как читать разные файловые системы, вы можете легко внести изменения в файл /boot/grub/menu.lst, чтобы изменить параметры следующей загрузки. Взгляните на пример конфигурации GRUB:

### Листинг 3. Пример конфигурации GRUB

```
timeout 5
color black/yellow yellow/black
default 0
password secretword

title linux
kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 quiet
vga=788 acpi=off
initrd (hd0,1)/boot/initrd.img

title experimental
kernel (hd0,1)/boot/bzImage-2.7.4 root=/dev/hda2 quiet

title dos
root (hd0,4)
makeactive
chainloader +1
```

## Изменения параметров во время работы загрузчика (LILO)

И LILO и GRUB позволяют передать специальные параметры выбранному вами ядру. Если вы используете LILO, вы можете передать параметры приглашению boot добавлением их к выбранному вами ядру. Например, для обычных параметров загрузки вы можете ввести:

**LILO: linux ether=9,0x300,0xd0000 root=/dev/ha2 vga=791 acpi=on**

Эта строчка передает специальные параметры модулю Ethernet, указывает корневой раздел, выбирает режим видео, и т.д. Конечно, не все это удобно, так как вы должны знать точно подходящие значения этих параметров и уметь правильно ввести их.

Особенное значение имеет параметр, который изменяет уровень запуска системы загрузчиком. Например, в целях восстановления системы, вы хотите загрузить систему в однопользовательском режиме. Осуществляется это следующим образом:

**LILO: experimental single**

или:

## Lilo: linux 1

Другой специальный параметр -- аргумент `init=`, который позволяет вам использовать программы, отличные от `init` в качестве первичного процесса. Параметры для режима аварийной ситуации могут быть следующими: `init=/bin/sh`, что, по крайней мере, позволит вам иметь в своем распоряжении командную строку (Linux shell), если `init` совсем вышел из строя.

## Изменения параметров во время работы загрузчика (GRUB)

С загрузчиком GRUB вы имеете еще большую гибкость. Фактически, GRUB представляет из себя оболочку командной строки и предоставляет пользователю базовую функциональность `shell`. GRUB дает возможность не только изменить базовую конфигурацию загрузчика, но даже читать файловые системы. Для настройки параметров загрузки, нажмите "`e`" в командной строке GRUB, после этого добавьте параметры (например, номер уровня запуска или ключевое слово "single" как в LILO). Все другие аргументы в приглашении загрузки, которые вы могли бы ввести, используя LILO, могут быть использованы в командной строке GRUB.

Для понимания своих возможностей вы можете открыть командную строку GRUB.

Например, предположите, что вам кажется, что ваш файл `/etc/inittab` плохо сконфигурирован, и вы хотите исследовать это перед загрузкой. Вы могли бы ввести:

```
grub> cat (hd0,2)/etc/inittab
```

Это позволило бы заранее просмотреть ваш инициализационный файл, без запуска операционной системы. Если бы там обнаружилась ошибка, то можно было бы загрузиться в однопользовательский режим и исправить ее.

## Настройка, осуществляемая после загрузчика

Как только вы осознаете шаги в загрузке Linux после загрузки ядра (другими словами, процесс `init` и все, что он вызывает), вы также осознаете, как их отредактировать. В основном, вся настройка осуществляется редактированием файла `/etc/inittab` и различных скриптов в каталоге `/etc/rc?.d/`.

Например, недавно мне понадобилось настроить видео BIOS на ноутбуке с Linux, базирующимся на Debian, использующем разработки третьих фирм. Если он не был запущен до того, как запустятся X11, мой драйвер XOrg не установил бы правильные режимы видео. Как только я выяснил, в чем была проблема, решение было столь же просто, как создание скрипта `/etc/rcS.d/S56-resolution.sh`. Другими словами, я запускал дополнительный скрипт при каждой загрузке системы.

Замечу, что я удостоверился, что этот скрипт исполняется раньше, чем `/etc/rcS.d/S70xorg-common` вследствие простого соглашения, что скрипты запускаются в алфавитном порядке (если бы я хотел, чтобы мой скрипт выполнялся позже, я, возможно, назвал бы его `S98-resolution.sh` вместо `/etc/rcS.d/S56-resolution.sh`). Может быть, я поместил бы этот скрипт только в каталог `/etc/rc5.d/`, чтобы он запускался, когда выполняются X11, но я могу вручную запустить `startx` из-под другого уровня запуска.

Все настройки в процессе инициализации открыты для редактирования, прямо в файловой системе; почти все можно исправить с помощью текстовых скриптов.

**Описание:** Это второй из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите этапы, которые система Linux™ проходит в процессе запуска, и их настройку.

## **Восстановление файловой системы**

### **О восстановлении**

Самое замечательное свойство Linux, которое рассматривается в перспективе обслуживания системы, это то, что все является файлом. Конечно, время от времени возникает вопрос , в каком файле что живет. Но зато, как правило, восстановление Linux означает применение основных утилит файловой системы, таких как [cp](#), [mv](#), [rm](#) и текстовый редактор типа [vi](#). Для автоматизации этих действий полезны такие инструменты как grep, awk и bash; или на более высоком уровне, perl или python. Но в данном учебном пособии мы не ставим целью изучать обращение с файлами.

Предположим, что вы знаете, как редактировать файлы и управлять ими, вот только в порушенной системе файлы, которых коснулось повреждение, возможно останутся вообще непригодными для использования.

### **Исправление испорченной файловой системы с помощью fsck**

Ваш лучший друг в восстановлении поврежденной файловой системы [fsck](#). [Следующий раздел \(тема 203\)](#) содержит больше информации, таким образом, здесь мы лишь представим этот инструмент в общих чертах.

Команда [fsck](#) является фактически только началом команды для большого количества других инструментов [fsck.\\*](#) -- [fsck.ext2](#), [fsck.ext3](#), или [fsck.reiser](#). Вы можете определить тип явно, используя опцию [-t](#), но [fsck](#) предпримет усилие понять самостоятельно. Прочтите страницу помощи man для [fsck](#) или [fsck.\\*](#) для получения более подробной информации. Основное, что вам нужно знать, что при использовании аргумента [-a](#) программа будет пытаться исправить все найденные ошибки.

Вы можете проверить неподмонтированную файловую систему, упоминая местонахождение устройства, на котором она находится. Например, введите [fsck /dev/hda8](#), чтобы проверить неиспользуемый раздел. Вы можете также проверить корневую файловую систему, набрав [fsck /home](#), но как правило, делают это, только если файловая система уже смонтирована как "только для чтения", а не для "чтения-записи".

### **Монтирование и отмонтирование с помощью mount and umount**

Одно из основных преимуществ систем Linux состоит в гибкости пользовательского контроля, при монтировании и отмонтировании файловых систем. В отличие от Windows и некоторых других операционных систем, местоположения разделов не автоматически закреплены ядром Linux, а присоединены к иерархии корневой файловой системы командой [mount](#). Кроме того, различные типы файловых систем (даже на различных устройствах) могут быть смонтированы в рамках той же самой иерархии. Вы можете отмонтировать конкретный раздел командой [umount](#), назначать любую точку монтирования (например, [/home](#)) или адрес устройства (например, [/dev/hda7](#)).

Когда производится восстановление файловой системы, возможность управлять точками монтирования позволяет вам проводить анализ состояния разделов, используя [fsck](#) или другие инструменты, без риска дальнейшего повреждения уже поврежденной файловой системы. Вы можете также в обычном порядке монтировать файловую систему, используя различные параметры; самые важные из них монтируют файловую систему для использования только в режиме чтения с помощью одного из синонимов [-r](#) или [-o ro](#).

В качестве примера, вы могли бы хотеть заменить местоположение каталога одного пользователя на каталог другого, или из-за повреждения раздела, или просто хотите расширить дисковое пространство, или переместиться на более быстрый диск. Такое изменение можно выполнить, используя:

```
# umount /home # old /dev/hda7 home dir
```

```
# mount -t xfs /dev/sda1 /home # new SCSI disk using XFS
# mount -t ext3 /dev/sda2 /tmp # also put the /tmp on SCSI
```

### Монтирование при загрузке с помощью /etc/fstab

Для восстановления, модернизации системы, и специальных целей полезно иметь возможность монтировать и отмонтировать файловые системы по желанию. Но для повседневной работы, вам будет удобно, чтобы необходимый конкретный набор подмонтирований осуществлялся автоматически при каждой загрузке системы. Вы управляете точками монтирования, прописывая нужные строки конфигурации в файл /etc/fstab. Типичная конфигурация могла бы выглядеть так:

#### Листинг 4. Пример конфигурации в /etc/fstab

```
/dev/hda7 / ext3 defaults 1 1
none /dev/pts devpts mode=0620 0 0
/dev/hda9 /home ext3 defaults 1 2
none /mnt/cdrom supermount
dev=/dev/hdc,fs=auto,ro,--,iocharset=iso8859-1,codepage=850,umask=0 0 0
none /mnt/floppy supermount
dev=/dev/fd0,fs=auto,--,iocharset=iso8859-1,sync,codepage=850,umask=0 0 0
none /proc proc defaults 0 0
/dev/hda8 swap swap defaults 0 0
```

Более полную информацию по этому вопросу вы найдете в [следующем разделе \(тема 203\)](#).

## Ресурсы

### Научиться

- [Оригинал данной главы](#) на developerWorks.
- В [LPIC Program](#) (Программе LPIC) вы найдете список заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.
- В "[Boot loader showdown: Getting to know LILO and GRUB](#)" (developerWorks, August 2005) обсуждается, как работают загрузчики и могут помочь вам решить, какой из двух самых популярных загрузчиков (LILO или GRUB) больше подойдет для вас.
- Найдите другие ресурсы для разработчиков Linux на [developerWorks Linux zone](#).

### Получить продукты и технологии

- [Закажите SEK для Linux](#), набор из двух DVD, содержащих trial-версии последнего программного обеспечения IBM для Linux от DB2®, Lotus®, Rational®, Tivoli® и WebSphere®.
- Постройте ваш следующий проект разработки для Linux с использованием [IBM trial software](#), загрузив его непосредственно с developerWorks.

# Учебник для экзамена LPI 201: Оборудование

*Администрирование, средний уровень (LPIC-2) тема 204*

Дэвид Мерц, автор, Gnosis Software, Inc.

Бред Хантинг, Mathematician, Университет Колорадо

**Описание:** Это четвертое из восьми пособий, где David Mertz и Brad Huntting продолжают подготовку к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Вы научитесь добавлять и настраивать оборудование в Linux™ системах, включая RAID-массивы, PCMCIA-карты, другие запоминающие устройства, мониторы, видео-карты и прочие компоненты.

[Больше статей из этой серии](#)

**Дата:** 02.09.2005

**Уровень сложности:** средний

## Перед тем как начать

Узнайте, чему эти обучающие программы могут научить вас, и как извлечь из них больше пользы.

## Об этой серии учебных пособий

Linux Professional Institute (LPI) производит сертификацию системных администраторов Linux двух уровней: для начинающих и среднего уровня. Чтобы получить сертификат каждого уровня, вы должны сдать два экзамена LPI.

Каждый из экзаменов состоит из нескольких тем, и каждая тема имеет свой рейтинг. Рейтинги указывают относительную важность каждой темы. Грубо говоря, вы вправе ожидать больше вопросов на экзамене на темы с более высоким рейтингом. Темы и их рейтинги для экзамена LPI 201:

### Тема 201

Ядро Linux (рейтинг 5).

### Тема 202

Запуск системы (рейтинг 5).

### Тема 203

Файловая система (рейтинг 10).

### Тема 204

Оборудование (рейтинг 8).

### Тема 209

Совместное использование файлов и служб (рейтинг 8).

### Тема 211

Поддержка системы (рейтинг 4). Тема данной главы.

### Тема 213

Настройка работ и автоматическое выполнение заданий (рейтинг 3).

### Тема 214

Устранение неполадок (рейтинг 6).

Linux Professional Institute® не приветствует использование для подготовки к экзаменам материалов и технологий от третьих лиц. Для более подробной информации обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## **Об этом руководстве**

Добро пожаловать в "Оборудование", четвертое из восьми руководств для подготовки к 201 экзамену LPI. В этом руководстве вы изучите процесс добавления и конфигурирования оборудования в Linux системе, включая RAID-массивы, PCMCIA карты, различные устройства хранения, дисплеев, видео-контроллеров и других компонент.

Это руководство организовано по следующим разделам, которые LPI относит к данной теме:

### **2.204.1 Конфигурирование RAID (рейтинг 2)**

Вы получите возможность конфигурировать и создавать программные RAID-ы. Этот раздел описывает использование программных средств mkraid и конфигурирование RAID 0, 1 и 5.

### **2.204.2 Добавление нового оборудования (рейтинг 3)**

Вы научитесь конфигурировать внутренние и внешние устройства системы, включая новые жесткие диски, терминальные устройства, источники бесперебойного питания, мультиплексоры последовательных портов и LCD панели.

### **2.204.3 Программное обеспечение и конфигурирование ядра (рейтинг 2)**

Вы научитесь задавать конфигурацию опций ядра для поддержки различных устройств, включая устройства UDMA66 и IDE устройства записи CD. Этот раздел включает использование LVM (Logical Volume Manager) для работы с жесткими дисками и их разделами, а также программные средства для задания установок для жестких дисков.

### **2.204.4 Конфигурирование устройств PCMCIA (рейтинг 1)**

Вы научитесь конфигурировать установку Linux для включения поддержки PCMCIA. Сюда входит конфигурирование PCMCIA устройств, таких как Ethernet адаптеры для того, чтобы происходила их автоматическая настройка при установке в компьютер.

Несмотря на то, что для работы с устройствами применяются приложения пользовательского уровня, в большой степени базовая поддержка устройств осуществляется собственно ядром Linux, модулями ядра, или же и тем и другим совместно. Единственным существенным исключением является тесная связь ядра Linux и оборудования в случае графических карт и компьютерных дисплеев. Для обслуживания обычного текстового экрана консоли вполне достаточно ядра (даже для поддержки некоторых графических возможностей через framebuffer), но обычно полная функциональность графической подсистемы контролируется XFree86 или более часто X.Org, драйверами X11. Практически все дистрибутивы включают X11, связанные с ними window manager'ы и окружения рабочего стола; но для серверов, где не требуется поддержки функциональности рабочего стола, использование X11 может оказаться излишним.

## **Требования**

Чтобы работа с этим учебником была максимально плодотворна, вы уже должны быть достаточно хорошо знакомы с ОС Linux и иметь компьютер с ОС Linux, чтобы на практике иметь возможность самостоятельно проверять работу команд, описанных в этом руководстве.

Кроме того дополнительную информацию о добавлении оборудования можно найти в двух других руководствах: "[LPI exam 201 prep \(topic 201\): Ядро Linux](#)" и "[LPI exam 201 prep \(topic 203\): Файловая система](#)." LPI экзамен по теме "Оборудование" предполагает хорошее знание ядра Linux, а также настройку используемых файловых систем, так что, пожалуйста, обращайтесь к этим учебникам во время подготовки к экзамену.

# Учебник для экзамена LPI 201: Оборудование

Администрирование, средний уровень (LPIC-2) тема 204

Дэвид Мерц, автор, Gnosis Software, Inc.

Бред Хантинг, Mathematician, Университет Колорадо

**Описание:** Это четвертое из восьми пособий, где David Mertz и Brad Huntting продолжают подготовку к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Вы научитесь добавлять и настраивать оборудование в Linux™ системах, включая RAID-массивы, PCMCIA-карты, другие запоминающие устройства, мониторы, видео-карты и прочие компоненты.

**Дата:** 02.09.2005

**Уровень сложности:** средний

## Конфигурирование RAID

### Что такое RAID?

RAID (Redundant Array of Inexpensive Disks) представляет из себя механизм объединения нескольких разделов или отдельных жестких дисков в большие или более надежные виртуальные диски. Изначально было определено большое число различных типов (уровней) RAID, но прижилось только три: RAID-0 (объединение дисков), RAID-1 (зеркалирование), and RAID-5 (объединение с сохранением контрольных сумм). RAID-4 также изредка используется; он достаточно близок к RAID-5, но контрольные суммы размещаются на специально выделенном устройстве, а не распределяются по дискам.

В этом руководстве обсуждается "*new-style*" RAID для Linux (он входит в ядра версии 2.4 и 2.6, для более ранних версий существуют backport'ы). "*Old-style*" RAID, изначально использовавшийся в версиях 2.0 и 2.2, содержит ошибки, и его использование не рекомендуется. По своей сути "*new-style*" представляет из себя 0.90 RAID layer, разработанный Ingo Molnar и др.

### Использование RAID массива

Работу с RAID массивами можно разделить на две части. Простейшей задачей можно считать монтирование RAID. Как только виртуальное устройство RAID сконфигурировано, для команды `mount` оно выглядит как обычное блочное устройство. RAID массив после его создания носит название вида `/dev/mdN` и может быть смонтирован следующим образом:

```
% mount /dev/md0 /home
```

Также вы можете включить монтирование виртуального RAID раздела в `/etc/fstab` (как правило, это наилучшее решение). Драйвер устройства считывает суперблоки сырых разделов диска для сборки сконфигурированного раздела RAID.

Более сложной задачей (или более многоступенчатой) является создание RAID устройства из соответствующих сырых разделов. Вы можете создать раздел RAID при помощи программы `mkraid` в сочетании с конфигурационным файлом `/etc/raidtab`.

Так же вы можете воспользоваться новой программой `mdadm`, при помощи которой, вы можете манипулировать RAID-устройствами без необходимости править конфигурационные файлы. В большинстве дистрибутивов `mdadm` вытесняет `raidtools` (который включает `mkraid`), но в этом руководстве обсуждается именно `mkraid`, для того, чтобы соответствовать требованиям, предъявляемым экзаменом LPI. Используемые подходы в

обоих случаях сходны, но вам будет необходимо справиться со справочным руководством man по **mdadm** для изучения опций командной строки.

### Формат /etc/raidtab

В файле /etc/raidtab используются следующие поля для описания компонентов RAID. Этот список не является исчерпывающим.

- **raiddev**: Раздел виртуального диска, предоставляемого RAID (/dev/md?). Это устройство, с которым могут работать **mkfs** и fsck, оно может быть подмонтировано как обычный дисковый раздел.
- **raid-disk**: Раздел используемый при создании RAID. Он должен иметь тип раздела 0xFD, установленный при помощи fdisk или подобной программы.
- **spare-disk**: Эти диски (как правило, это один диск) обычно остаются не задействованными. В случае если один из дисков, входящих в raid, выходит из строя, spare диск начинает выступать в качестве его замены.

### Конфигурирование RAID-0

RAID-0 или "disk striping" дает большую производительность при операциях ввода/вывода ценой уменьшения общей надежности (выход из строя одного диска из raid-массива может привести к утрате всего устройства RAID). В качестве примера ниже приведен /etc/raidtab для создания устройства RAID-0:

```
raiddev /dev/md0
    raid-level      0
    nr-raid-disks  2
    nr-spare-disks 0
    chunk-size     32
    persistent-superblock 1
    device          /dev/sda2
    raid-disk       0
    device          /dev/sdb2
    raid-disk       1
```

Здесь определено виртуальное устройство RAID-0, имеющее название /dev/md0. Первые 32 KB устройства /dev/md0 выделяются на /dev/sda2, следующие 32 KB на /dev/sdb2, третий на /dev/sda2 и т.д.

Для создания устройства выполните следующую команду:

```
% sudo mkraid /dev/md0
```

При использовании **mdadm** вместо файла /etc/raidtab используются опции.

### Конфигурирование RAID-1

RAID-1 или "disk mirroring" просто дублирует данные на обоих блочных устройствах. RAID-1 великолепно справляется с задачами защиты от аппаратных сбоев, но заметно снижает производительность. RAID-1 в целом стоит дороже, так как половина вашего дискового пространства резервируется. Например:

```
raiddev /dev/md0
    raid-level      1
    nr-raid-disks  2
    nr-spare-disks 1
    persistent-superblock 1
    device          /dev/sdb6
```

```
raid-disk      0
device        /dev/sdc5
raid-disk      1
device        /dev/sdd5
spare-disk    0
```

Данные, записываемые на /dev/md0, будут сохранены и на /dev/sdb6 и на /dev/sdc5.

Устройство /dev/sdd5 сконфигурировано как *hot spare*. В случае сбоя на устройстве /dev/sdb6 или /dev/sdc5, данные будут перенесены на /dev/sdd5, и оно будет переведено во включенное состояние для замены сбояного устройства.

## Конфигурирование RAID-5

RAID-5 требует, по крайней мере, трех устройств и использует коррекцию ошибок для получения преимуществ, предоставляемых распределенными дисками, вместе с устойчивостью к сбою одного из устройств. Положительным моментом является необходимость использования только одного дополнительного устройства для обеспечения надежности. Отрицательным моментом является большая сложность RAID-5; при возникновении сбоя в одном из устройств, он переходит в режим *degraded mode*, который существенно снижает пропускную способность операций ввода/вывода, пока не будет завершена процедура подключения резервного spare-диска и перекачивание на него данных.

```
raiddev /dev/md0
  raid-level      5
  nr-raid-disks   7
  nr-spare-disks 0
  persistent-superblock 1
  parity-algorithm left-symmetric
  chunk-size      32
  device          /dev/sda3
  raid-disk       0
  device          /dev/sdb1
  raid-disk       1
  device          /dev/sdc1
  raid-disk       2
  device          /dev/sdd1
  raid-disk       3
  device          /dev/sde1
  raid-disk       4
  device          /dev/sdf1
  raid-disk       5
  device          /dev/sdg1
  raid-disk       6
```

## Использование mke2fs или mke3fs

Если вы форматируете виртуальные устройства RAID-5 при помощи e2fs или e3fs, вы должны обращать внимание на опцию *stride*. Опция -R *stride=nn* позволяет mke2fs размещать данные файловой системы ext2 так, что они лучше воспринимаются RAID-устройством.

Если chunk size установлен в 32 KB, это означает, что эти 32 KB последовательных данных будут размещаться на одном диске. Если файловая система ext2 имеет размер блока в 4 KB, то на восемь блоков файловой системы будет приходиться один array chunk. Мы можем указать эту информацию файловой системе, запустив команду:

```
% mke2fs -b 4096 -R stride=8 /dev/md0
```

Производительность RAID-5 существенно увеличивается при создании файловой системы с правильной информацией о stride.

### **Поддержка в ядре, обслуживание сбоев**

Включение в ядре опции *persistent-superblock* дает возможность ядру стартовать RAID автоматически при загрузке системы. New-style RAID использует persistent superblock и поддерживается в ядрах 2.4 и 2.6. Для устаревших ядер версий 2.0 и 2.2 доступны соответствующие патчи.

При выходе из строя устройства происходит следующее:

- **RAID-0:** Все данные теряются;
- **RAID-1/RAID-5:** Сбойное устройство отключается, а spare-диск (если он имеется) включается, и данные переносятся на него.

Документ "The Software-RAID HOWTO" из Linux HOWTO project описывает переключение между устройствами при сбоях или обновлении устройств, включая hot-swap диски, и описывает случаи, когда необходима перезагрузка. Обычно, SCSI (или Firewire) устройства поддерживают горячую замену, а IDE устройства -- нет.

**Описание:** Это четвертое из восьми пособий, где David Mertz и Brad Huntting продолжают подготовку к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Вы научитесь добавлять и настраивать оборудование в Linux™ системах, включая RAID-массивы, PCMCIA-карты, другие запоминающие устройства, мониторы, видео-карты и прочие компоненты.

## **Установка нового оборудования**

### **Оборудование**

Linux, особенно последние версии, обладает удивительной стабильностью работы и широким спектром совместимости с различными устройствами. В общем и целом, есть два уровня поддержки аппаратуры, которыми следует озабочиться. Первый уровень обеспечивает базовую поддержку на уровне системы, обычно это означает загрузку модуля ядра, соответствующего вашему устройству.

Второй уровень имеет отношение к некоторым устройствам, в той или иной степени требующим поддержки со стороны подсистемы X11R6: обычно это или XFree86, или X.Org (в прежние времена, использовались и коммерческие подсистемы X11, но в данном руководстве они не обсуждаются).

Поддержка основных категорий hot-swappable устройств, например, работающих через PCMCIA или USB интерфейсы, обсуждается далее в соответствующих разделах.

### **X11**

К сведению: изначально X.Org является преемником проекта XFree86 (технически, это просто другая его ветвь). Не смотря на то, что XFree86 официально не свернут, практически все производители переключились на X.Org по лицензионным причинам. К счастью, за исключением небольших изменений в названиях конфигурационных файлов, основная часть кода в обеих ветвях одинакова, некоторые новые возможности с большей вероятностью будут поддерживаться только в X.Org.

X11R6 представляет собой (сетевое) графическое представление приложений на

пользовательской рабочей станции. Вопреки интуитивному пониманию, "Х сервер" -- это физическая машина, с которой непосредственно взаимодействует пользователь через клавиатуру, мышь, видео карту, дисплей и т.д. "Х клиент" -- это машина, которая предоставляет процессорное время, дисковое пространство и другие ресурсы, не имеющие непосредственного отношения к уровню представления для запуска приложения. Во многих и даже в большинстве Linux систем, Х сервер и Х клиент сосуществуют на одной и той же машине, и эффективный локальный канал обмена информацией используется для взаимодействия с пользовательскими устройствами ввода/вывода.

Х сервер, такой как X.Org, необходим для обеспечения поддержки устройств ввода/вывода, при помощи которых пользователь будет взаимодействовать с приложением. В подавляющем большинстве случаев сложности встречаются при настройке видео-карт и мониторов. К счастью, эти сложности остаются в прошлом с выходом последних версий X.Org/XFree86, где успешно реализована процедура автоматического детектирования оборудования. Технически Х сервер также необходим для поддержки устройств ввода -- клавиатуры и мыши -- но обычно их подключение происходит безболезненно, поскольку они имеют стандартные интерфейсы. Все остальное: доступ к дисковым устройствам, принтерам, таким специальным устройствам, как сканеры и др. обслуживается клиентскими Х приложениями и, в конечном счете, ядром Linux.

### **Поддержка устройств ядром системы**

Практически все, что вам следует знать о поддержке устройств ядром Linux, ограничивается поиском, сборкой и загрузкой правильных модулей ядра. Все это исчерпывающе изложено в руководстве по теме 201, на большинство вопросов читатели смогут найти ответы там.

Для работы с модулями ядра системный администратор должен иметь представление о таких командах, как lsmod, insmod и modprobe и, в меньшей степени, о rmmod. Команды lsmod, insmod и rmmod -- это команды низкого уровня для получения списка загруженных модулей, их загрузки и выгрузки в работающее ядро Linux. Команда modprobe проводит эти действия на более высоком уровне, проверяя взаимные зависимости и осуществляя необходимые вызовы insmod и rmmod в зависимости от необходимости.

### **Осмотр оборудования**

Отдельные утилиты могут оказаться полезными для получения информации об имеющемся оборудовании. Команда lspci выдает детальную информацию о найденных PCI устройствах (во многих случаях включая даже те, которые работают через PCMCIA или USB шины). Соответственно, setpci может конфигурировать устройства на PCIшине. Команда lspnp выводит список BIOS device node и ресурсов для plug-and-play устройств. Команда lsusb подобным образом просматривает USB (для модификации конфигураций используется setpnp).

### **Настройка сервера X11 (первая часть)**

Исходно X.Org (или XFree86) поставляется с набором видео драйверов и драйверов для других периферийных устройств, вам просто нужно подобрать правильные. В конечном счете, вся конфигурация Х сервера располагается в достаточно детализированном, а местами несколько напоминающем шифровку, файле /etc/X11/xorg.conf (или xfree86.conf). Ряд стандартных утилит может быть использован для упрощения процесса конфигурирования, но текстовый редактор сработает в любом случае. Некоторые фронтэнды включаются непосредственно в X.Org, так для графического конфигурирования включен xorgcfg (допускаю, что вам он покажется не слишком работоспособным) и xorgconfig для конфигурирования в текстовом режиме. Многие дистрибутивы Linux снабжены более дружественными фронтендами.

Команда SuperProbe может оказаться полезной для определения модели вашей видео-карты.

Вы можете так же обратиться к базе данных /usr/X11R6/lib/X11/Cards для получения детальной информации о поддерживаемых видео-картах.

## Настройка сервера X11 (вторая часть)

Внутри конфигурационного файла /etc/X11/xorg.conf вы должны создать серию блоков "Section" ... "EndSection", каждая из которых определяет ряд деталей и опций, относящихся к конкретному устройству. Имена этих разделов следующие:

* Files:	Пути поиска файлов
* ServerFlags:	Флаги сервера
* Module:	Загрузка динамических модулей
* InputDevice:	Описание устройств ввода
* Device:	Описание графического устройства
* VideoAdaptor:	Xv описание видео-адаптера
* Monitor:	Описание монитора
* Modes:	Описание видео мод
* Screen:	Конфигурация экрана
* ServerLayout:	Общая конфигурация
* DRI:	Конфигурация относящаяся к DRI
* Vendor:	Специфичная для данного производителя конфигурация

## Настройка сервера X11 (часть третья)

Среди всех секций, **Screen** выступает в качестве мастер-конфигурации системы отображения. Например, вы можете определить несколько разделов **Monitor**, но выбран будет только один, указанный в:

```
Section "Screen"
    Identifier      "Default Screen"
    Device          "My Video Card"
    Monitor         "Current Monitor"
    DefaultDepth   24
    SubSection     "Display"
        Depth       24
        Modes        "1280x1024" "1024x768" "800x600"
    EndSubSection
    # more subsections and directives
EndSection
```

Раздел, носящий название **ServerLayout**, в действительности является главной ("master") конфигурацией -- он ссылается и к используемой секции **Screen**, и к различным секциям **InputDevice**. В случае, если у вас возникли проблемы, скорее всего, они решаются правильным выбором **Device** или **Monitor**. К счастью, DPMS-мониторы нынче, как правило, избавляют нас от болезненной настройки опций **Modeline** (в прежние суровые времена, вам было необходимо отыскать крайне специфичные для вашего монитора частотно/временные характеристики, нынче же, DPMS делает эту работу за вас).

**Описание:** Это четвертое из восьми пособий, где David Mertz и Brad Huntting продолжают подготовку к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Вы научитесь добавлять и настраивать оборудование в Linux™ системах, включая RAID-массивы, PCMCIA-карты, другие запоминающие устройства, мониторы, видео-карты и прочие компоненты.

## Конфигурирование PCMCIA устройств

### PCMCIA

PCMCIA устройства иногда также называются устройствами PC-Card. Это тонкие, размером с кредитную карту, модули изначально проектировавшиеся с расчетом на hot-swap и легкость транспортировки. Наибольшее распространение получили они в ноутбуках. Однако некоторые десктопные и серверные конфигурации также имеют PCMCIA интерфейсы, зачастую установленные на внешних креплениях и подсоединенные через различные шины (специальные PCI или ISA карты, USB-трансляторы и т.д.). Различное оборудование может быть реализовано в размерах PCMCIA карт, включая Wireless и Ethernet адаптеры, микродрайвы, flash-диски, модемы, SCSI-адAPTERЫ и многое другое специфическое оборудование.

Технически, PCMCIA интерфейс представляет из себя переходник к расположенным ниже уровнем ISA или PCI шинам. В большинстве случаев, передача происходит прозрачным образом -- одни и те же модули ядра или программы, которые взаимодействуют с ISA или PCI устройством, используются для обслуживания протокола обмена предоставляемого PCMCIA. Единственное настоящее различие PCMCIA устройств заключается в распознавании события при установке устройства и определении типа карты, для которого следует загрузить драйвер.

В настоящее время оборудование PCMCIA скрывается в тени USB и/или Firewire устройств. Несмотря на то, что PCMCIA обладает более удобным конструктивом (обычно скрывающий карту в корпусе), USB более универсален для широкого круга машин. В результате многие устройства, прежде выпускавшиеся в стандарте PCMCIA, нынче оформляются в виде устройств в стиле USB "dongle", а старые доступны только на распродажах.

### Определение PCMCIA устройства (часть первая)

В современных ядрах -- 2.4 и выше -- поддержка PCMCIA доступна в виде модуля ядра. Современные дистрибутивы такую поддержку включают, но если вы собираете свое собственное ядро, включите опции **CONFIG\_HOTPLUG**, **CONFIG\_PCMCIA** и **CONFIG\_CARDBUS**. Ранее такая поддержка обеспечивалась пакетом **pcmcia-cs**.

Модули **pcmcia\_core** и **pcmcia** отвечают за поддержку PCMCIA устройств. Модуль **yenta\_socket** также загружается для поддержки интерфейса CardBus (PCI-over-PCMCIA):

```
% lsmod | egrep '(yenta)|(pcmcia)'  
pcmcia           21380  3 atmel_cs  
yenta_socket     19584   1  
pcmcia_core      53568  3 atmel_cs,pcmcia,yenta_socket
```

При установке карты в PCMCIA слот демон **cardmgr** обращается к базе данных **/etc/pcmcia/config** и загружает тот драйвер, который нужен.

### Определение PCMCIA устройства (часть вторая)

Теперь взглянем на процесс идентификации PCMCIA устройства в действии. Я вставляю карту в Linux лэптоп со слотом PCMCIA, с включенной поддержкой указанных ранее модулей ядра. Я могу воспользоваться программой **cardctl** для просмотра информации об имеющемся оборудовании:

```
% cardctl ident  
Socket 0:  
product info: "Belkin", "11Mbps-Wireless-Notebook-Network-Adapter"  
manfid: 0x01bf, 0x3302
```

```
function: 6 (network)
```

Эта информация предоставляется модулем ядра `pcmcia_core`, запросившего ее у физической карты. Как только идентификация проведена, `cardmgr` сканирует базу данных для того, чтобы найти соответствующий драйвер. Выглядит это примерно так:

```
% grep -C 1 '0x01bf,0x3302' /etc/pcmcia/config
card "Belkin FSD6020 v2"
  manfid 0x01bf,0x3302
  bind "atmel_cs"
```

В этом случае нам нужен модуль ядра `atmel_cs` для поддержки wireless интерфейса, предоставляемого этой картой. Вы можете увидеть это, заглянув в `/var/lib/pcmcia/stab` или `/var/run/stab`, в зависимости от вашей системы:

```
% cat /var/run/stab
Socket 0: Belkin FSD6020 v2
0      network atmel_cs      0      eth2
```

## Получение отладочной информации о PCMCIA устройстве

В приведенном выше примере выполнявшиеся этапы были незаметны. Карта была опознана, драйвера загружены, и все было подключено. Это идеальная ситуация. Если же что-то не в порядке, вы можете обнаружить, что драйвер, который надо загрузить, не найден.

Если вы уверены, что ваше PCMCIA может использовать имеющийся драйвер (например, он совместим с другим чипсетом), вы можете запустить `insmod` вручную для загрузки подходящего модуля. Или же, если вы постоянно используете эту карту, вы можете отредактировать `/etc/pcmcia/config` для того, что бы добавить поддержку этой карты, указав необходимый драйвер. Однако в случае, если ваше предположение относительно модуля не оправдалось, вам следует убедиться, что карта действительно совместима с какой-то другой известной PCMCIA картой.

Настройка загрузки PCMCIA может быть сделана через установочный скрипт `/etc/pcmcia/`, поименованный согласно функциональной категории. Например, когда 802.11b карта, подобная той, что загружалась в предыдущем примере, запускается скрипт `/etc/pcmcia/wireless`. Вы можете настроить эти скрипты, если устройство требует специальных настроек.

## Использование "схем" для различных конфигураций

Если у вас возникает необходимость использовать PCMCIA устройство в различных конфигурациях, вы можете воспользоваться командой `cardctl scheme` для установки (или запроса) конфигурации. Например:

```
% sudo cardctl scheme foo
checking: eth2
/sbin/ifdown: interface eth2 not configured
Changing scheme to 'foo'...
Ignoring unknown interface eth2=eth2.
% cardctl scheme
Current scheme: 'foo'.
```

В этом случае, я в действительности не определяю схему `foo`, но если вы ее измените, то произойдет реконфигурация. Схемы могут быть использованы в настроечных скриптах при помощи анализа переменной `$ADDRESS`:

```
# in /etc/pcmcia/network.opts (called by /etc/pcmcia/network)
case "$ADDRESS" in
work,*,*,*)
    # definitions for network in work scheme ...
;;
default,*,*,*)
    # definitions for network in default scheme ...
;;
esac
```

Конечно, вы можете устанавливать схемы в инициализационных скриптах или переключать их событиями (через задания `cron`, GUI интерфейс и т.д.).

**Описание:** Это четвертое из восьми пособий, где David Mertz и Brad Huntting продолжают подготовку к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Вы научитесь добавлять и настраивать оборудование в Linux™ системах, включая RAID-массивы, PCMCIA-карты, другие запоминающие устройства, мониторы, видео-карты и прочие компоненты.

## Конфигурирование Universal Serial Bus устройств

### USB

Как уже было отмечено в разделе о PCMCIA, USB представляет из себя новую технологию, вытесняющую сейчас PCMCIA. USB поддерживает до 127 устройств с помощью гибкой радиальной топологии состоящей из хабов у оконечных устройств. USB имеет несколько версий с последовательно увеличивающимися скоростями передачи, последняя из них -- 2.0. Эта последняя версия USB теоретически поддерживает скорость обмена до 480 MBsec. USB 1.1 поддерживает меньшие скорости, до 12 MBsec. На практике, по ряду причин некоторые устройства фактически работают на меньших скоростях, чем предусмотренные теоретически -- тем не менее, более быстрый интерфейс более перспективен.

### Распознавание USB устройств (часть первая)

С точки зрения администрирования, USB работает подобно PCMCIA. Он обслуживается модулем ядра `usbcore`. В ядрах 2.4+, предусмотрена более совершенная поддержка, чем в ядрах 2.2. На следующем уровне за `usbcore`, вступает в действие один из следующих модулей: `uhci`, `uhci_hcp`, `ohci`, `ohci_hcp`, `ehci`, `ehci_hcp`. Какой именно модуль понадобится, зависит от чипсета, использованного в вашем компьютере. Модуль `ehci` подключается, если они поддерживают высокоскоростную передачу по USB 2.0. Вообще же говоря, если ваш компьютер поддерживает `ehci` (или `ehci_hcp`), может потребоваться загрузка и модуля `ehci` для обеспечения обратной совместимости. Книга Брэда Хардса "The Linux USB sub-system" содержит детальное описание соответствий между различными чипсетами и модулями ядра. При создании ядра, которое будет использоваться на различных машинах, вам следует собрать все USB модули.

Для обеспечения корректной поддержки ядра система hotplug должна обеспечивать загрузку любых драйверов, необходимых для обслуживания подключенного устройства USB. Файл `/proc/bus/usb/devices` содержит детальную информацию о доступных в настоящее время USB устройствах (как хабов, так и периферийных устройств).

### Распознавание USB устройств (часть вторая)

Обычно шина USB монтируется как динамически генерируемая файловая система, подобная

файловой системе /proc/. В зависимости от дистрибутива, /proc/bus/usb/ может монтироваться или в стартовых скриптах, типа /etc/rcS.d/S02mountvirtfs, или же через конфигурацию /etc/fstab. В последнем случае, вы сможете увидеть там строку подобную следующей:

```
# /etc/fstab
none /proc/bus/usb usbdevfs defaults 0 0
```

Инициализационный же скрипт может выглядеть следующим образом:

```
mount -t usbdevfs none /proc/bus/usb
```

Механизмы распознавания устройств и управление всей подсистемой USB кроется в /etc/hotplug/, в первую очередь, в /etc/hotplug/usb.rc и /etc/hotplug/usb.agent. Установка USB устройства будет проводиться через операцию **modprobe** для нужного драйвера. Вы можете провести и дальнейшую настройку для данного устройства путем создания скрипта /etc/hotplug/usb/\$DRIVER для вашего конкретного устройства.

## Ресурсы

### Научиться

- [Оригинал данной главы](#) на developerWorks.
- В [LPIC Program](#) (Программе LPIC) вы найдете список заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.
- " [Common threads: Advanced filesystem implementor's guide, Parts 1 - 13](#)" (developerWorks, starting June 2001) замечательные руководства по файловым системам Linux.
- В "[Understanding Linux configuration files](#) (developerWorks, декабрь 2001) описаны приемы настройки в системах Linux конфигурационных файлов, которые используются для управления правами пользователей, системными приложениями, демонами, службами и решения других задач администрирования в многопользовательском, многозадачном окружении.
- Найдите другие ресурсы для разработчиков Linux на [developerWorks Linux zone](#).

### Получить продукты и технологии

- [Закажите SEK для Linux](#), набор из двух DVD, содержащих trial-версии последнего программного обеспечения IBM для Linux от DB2®, Lotus®, Rational®, Tivoli® и WebSphere®.
- Постройте ваш следующий проект разработки для Linux с использованием [IBM trial software](#), загрузив его непосредственно с developerWorks.

# Учебник для экзамена LPI 201: Файловая система

*Администрирование, средний уровень (LPIC-2) тема 203*

Дэвид Мерц, автор, Gnosis Software, Inc.

**Описание:** Это третий из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите, как монтировать, проверять, создавать новые и исправлять поврежденные файловые системы.

[Больше статей из этой серии](#)

**Дата:** 31.08.2005

**Уровень сложности:** средний

## Перед тем как начать

Узнайте, чему эти обучающие программы могут научить вас, и как извлечь из них больше пользы.

## Об этой серии учебных пособий

Linux Professional Institute (LPI) производит сертификацию системных администраторов Linux двух уровней: для начинающих и среднего уровня. Чтобы получить сертификат каждого уровня, вы должны сдать два экзамена LPI.

Каждый из экзаменов состоит из нескольких тем, и каждая тема имеет свой рейтинг. Рейтинги указывают относительную важность каждой темы. Грубо говоря, вы вправе ожидать больше вопросов на экзамене на темы с более высоким рейтингом. Темы и их рейтинги для экзамена LPI 201:

### Тема 201

Ядро Linux (рейтинг 5).

### Тема 202

Запуск системы (рейтинг 5).

### Тема 203

Файловая система (рейтинг 10). Тема данной главы.

### Тема 204

Оборудование (рейтинг 8).

### Тема 209

Совместное использование файлов и служб (рейтинг 8).

### Тема 211

Поддержка системы (рейтинг 4).

### Тема 213

Настройка работ и автоматическое выполнение заданий (рейтинг 3).

### Тема 214

Устранение неполадок (рейтинг 6).

Linux Professional Institute® не приветствует использование для подготовки к экзаменам материалов и технологий от третьих лиц. Для более подробной информации обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## **Об этом руководстве**

Добро пожаловать в главу "Файловая система", третий из восьми учебников, разработанных для подготовки к экзамену LPI 201. На этом этапе вы узнаете, как управлять монтированием и отмонтированием файловых систем, проверять имеющиеся файловые системы, создавать новую и исправлять поврежденную файловую систему.

Это руководство организовано по следующим разделам, которые LPI относит к данной теме:

### **2.203.1 Создание и конфигурирование файловой системы (рейтинг 3)**

Вы научитесь правильно создавать и конфигурировать стандартную файловую систему Linux. Вы изучите различные типы файловых систем и их настройки, а также возможности управления файловыми системами, чтобы приспособиться к требованиям дискового пространства или присоединению устройств.

### **2.203.2 Управление файловой системой Linux (рейтинг 3)**

Вы научитесь автоматически монтировать файловые системы. Также узнаете, как настраивать автомонтирование для файловых систем устройств и сети и создавать не-ext2 файловые системы для устройств типа CD-ROM.

### **2.203.3 Поддержка файловой системы Linux (рейтинг 4)**

Вы научитесь должным образом осуществлять поддержку файловой системы Linux, используя системные утилиты. Кроме того, вы узнаете, как управлять стандартной файловой системой ext2.

Это учебное пособие посвящено элементам Linux, а также внешним инструментам, которые полезны для работы с системами Linux. Поддержка файловых систем, устройств и разделов либо включена в основное ядро либо включена в модули ядра.

Различные инструменты, которые Вы, вероятно, будете использовать для управления файловыми системами Linux, - это утилиты пространства пользователя, и поэтому обычно включены только в дистрибутивы Linux, но не является неотъемлемой частью Linux. Тем не менее, инструменты файловой системы необходимы для работы практически с каждой системой Linux независимо от ее предполагаемого использования (даже для несетевых или внутренних систем).

## **Требования**

Чтобы работа с этим учебником была максимально плодотворна, вы уже должны быть достаточно хорошо знакомы с ОС Linux и иметь компьютер с ОС Linux, чтобы на практике иметь возможность самостоятельно проверять работу команд, описанных в этом руководстве.

# **Учебник для экзамена LPI 201: Файловая система**

*Администрирование, средний уровень (LPIC-2) тема 203*

[Дэвид Мерц](#), автор, Gnosis Software, Inc.

**Описание:** Это третий из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите, как монтировать, проверять, создавать новые и исправлять поврежденные файловые системы.

[Больше статей из этой серии](#)

**Дата:** 31.08.2005

**Уровень сложности:** средний

## **Создание и конфигурирование файловой системы**

Начнем с создания и конфигурирования файловых систем и их параметров.

### **Создание разделов**

Прежде чем вы сможете пользоваться файловыми системами Linux, вам нужно создать их. Для того чтобы создать файловую систему, вам сначала необходимо создать раздел, на котором вы впоследствии будете создавать файловую систему. На жестком диске с архитектурой x86 может быть создано до четырех первичных (primary) разделов, при этом последний из этих первичных разделов может быть расширенным (extended) и включать в себя множество логических разделов.

До недавнего времени существовали ограничения на порядковый номер цилиндра, на котором может находиться загрузочный раздел, на максимальные размеры дисков, на местоположение первичного раздела на больших дисках и так далее. Однако в последние несколько лет практически все системные BIOS научились управлять дисками независимо от их размера, и современные загрузчики (по крайней мере для Linux) не имеют никаких особых ограничений, касающихся размеров разделов или их местоположения.

Единственный оставшийся момент, о котором следует побеспокоиться, это файловые системы, отличные от Linux. Для некоторых из них время от времени все же появляется необходимость располагаться на первичном разделе в начале жесткого диска. Разделы же Linux могут прекрасно жить как на расширенном разделе, так и на любом доступном диске.

Существует несколько широко используемых инструментов в мире Linux для того, чтобы создавать и управлять разделами на жестких дисках. Самый старый из них - *fdisk*. Несколько позже становится популярным основанный на *curses* инструмент *cfdisk*. Также часто используется для разбиения жестких дисков программа *parted* сообщества GNU. Ну а, в свою очередь, программы инсталляции, предлагаемые большинством известных дистрибутивов, и/или их графическая оболочка, обеспечивают удобный интерфейс для процедуры разбиения диска и просмотра таблицы разделов.

Из всех этих инструментов, утилита *fdisk* остается самым гибким и нетребовательным инструментом. Но не стоит обольщаться. Случайная запись неправильной таблицы разделов влечет проблемы независимо от того, какой программой вы пользуетесь. Но если ваши разделы были созданы нестандартными способами, например, не инструментами Linux, то, возможно, *fdisk* будет работать там, где другие инструменты могли бы отказаться пробовать вообще. Однако, программа *cfdisk* более дружественна пользователю и более интерактивна (в том случае, если она не откажется работать). Кроме того, *parted* обладает великолепными возможностями для изменения размера и перемещения существующего раздела не хуже чем *fdisk* или *cfdisk*.

Какую бы программу вы не использовали, чтобы создать разделы, все они работают похожим образом. Эти действия вы должны выполнять как суперпользователь, лучше всего, в однопользовательском режиме. Трудно переоценить важность этого момента, поэтому будьте очень осторожны, когда вы изменяете разделы, сделайте резервную копию всех важных данных и обратите особое внимание на то, какие изменения вы делаете.

Прежде чем вы начнете изменять таблицу разделов, неплохо было бы выяснить, какие разделы уже существуют. С помощью команды ***fdisk -l /dev/hda*** (или такой же, но для других дисков, например, */dev/hdb* или */dev/sda*) вы можете узнать, какие разделы имеются в системе. Команда ***mount*** также полезна для выяснения, как фактически используются существующие разделы. Если вы хотите создать новый раздел, имейте в виду какие-нибудь свободные секторы на последнем первичном разделе, которые можно было бы использовать для нового расширенного раздела.

Взгляните на пример таблицы разделов в моей ОС Linux:

## Листинг 1. Пример обычной таблицы разделов

```
% fdisk -l /dev/sda
Disk /dev/sda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders

Device Boot Start End Blocks Id System
/dev/sda1 * 1 1216 9767488+ 7 HPFS/NTFS
/dev/sda3 1217 4255 24410767+ 83 Linux
/dev/sda4 4256 9729 43969905 5 Extended
/dev/sda5 4256 4380 1004031 82 Linux swap /
Solaris
/dev/sda6 4381 5597 9775521 83 Linux
```

Из этой таблицы можно кое-что почерпнуть. Во-первых, можно увидеть, что первый раздел, вероятно, используется другой операционной системой. Далее введем `mount`, чтобы выяснить, как используются разделы:

```
% mount | head -1
/dev/sda3 on / type reiserfs (rw,noatime,notail,commit=600)
```

Таким образом, существующая система смонтирована как корневая файловая система на `/dev/sda3`. Возможно, наиболее интересным наблюдением является то, что раздел `/dev/sda4` простирается до 9729 цилиндра, но этот расширенный раздел использует только часть всего имеющегося места.

После обнаружения свободного места, доступного на жестком диске, используем его для создания нового раздела с помощью `fdisk`:

```
% fdisk /dev/sda
```

Наш жесткий диск насчитывает 9729 цилиндров. Это не является нарушением, но все же больше чем 1024 и, при определенных условиях, может вызвать проблемы с:

1. Загрузочным программным обеспечением (типа старых версий LILO)
2. Загрузочными и разбивающими жесткие диски программами от других операционных систем (например, DOS FDISK или OS/2 FDISK)

## Листинг 2. Создание раздела

```
Command (m for help): n
Command action
l logical (5 or over)
p primary partition (1-4)
l
First cylinder (5598-9729), default 5598:
Using default value 5598
Last cylinder or +size or +sizeM or +sizeK (5598-9729, default 9729):
+10000M

Command (m for help): w
The partition table has been altered!
```

Все, что следует за двоеточием, должно быть введено пользователем (вами). Таким образом,

мы создали новый раздел Linux размером 10 GB.

```
/dev/sda7 5598 6814 9775521 83 Linux
```

Далее будет рассказано, как использовать вновь созданный раздел. Возможно, понадобится перезагрузить систему, чтобы получить доступ к новому разделу.

## Создание файловой системы

Одного лишь наличия раздела недостаточно; вы должны создать на нем файловую систему. Мы создали новый раздел Linux /dev/sda7 и теперь должны выбрать, какой тип файловой системы, поддерживаемой Linux, использовать на этом разделе. Может быть, мы хотим создать исторически использующуюся по умолчанию файловую систему ext2? Или более новую журналируемую расширенную файловую систему ext3? Возможно, мы хотим создать одну из продвинутых файловых систем, привнесенных в Linux другими компаниями, например, ReiserFS, XFS или JFS. Или нам нужно иметь файловую систему, которая может взаимодействовать с другой операционной системой, типа Minix, MS-DOS, или VFAT (некоторые другие могут быть прочитаны, если уже созданы, но не всегда могут быть созданы инструментами Linux).

Для того, чтобы создавать новые файловые системы, нужно использовать следующие принятые обозначения `mkfs.*`. Таким образом, ваша файловая система может быть создана при помощи `mkfs.ext2`, `mkfs.minix`, `mkfs.xfs`, и так далее, обычно это установлено в `/sbin/`. Также вы можете задать любую из них, используя синтаксис `mkfs -t <fstype>`. Для некоторых (не всех) типов файловых систем имеется короткая форма записи, например, `mke3fs`. Возможность создания определенного типа файловой системы зависит от дистрибутива, который вы используете, и его версии, а также, от дополнительного ПО, которое вы самостоятельно установили. При этом заметим, что `mkfs.ext2` имеется практически в любом дистрибутиве.

Создать файловую систему достаточно просто. Вам нужно только применить инструмент `mkfs.*` к нужному разделу (к тому, на котором вы хотите ее создать). Например:

```
% mkfs.xfs /dev/sda7
```

Сообщения, которые вы далее увидите, зависят от файловой системы, которую вы предпочли. Вообще, эти сообщения дают вам информацию относительно числа inod'ов, блоков, типа журнализации (если имеется журналируемость), протяженности, и другую, соответствующую характеру использующейся файловой системы. Многие (но не все, к сожалению) из средств, создающих файловую систему предупредят вас, если вы решите создать новую файловую систему на разделе с уже существующей файловой системой, так что приступайте с большой осторожностью (создание новой файловой системы поверх старой может привести к потере данных).

## Создание файловой системы ISO при помощи mkisofs

Отдельный и особенный случай создания файловой системы - это создание *файловой системы ISO*, которая является образом системы и может быть записана на компакт-диск или DVD-диск. Файловая система ISO является особенной в том смысле, что является действительно только файлом (правда, большим) с данными, выстроенным определенным способом, а не упорядочиванием физического устройства, такого как `/dev/cdrom` или `/dev/hdb3`.

Основная идея создания файловой системы ISO, которая означает либо ISO 9660 либо HFS, состоит в том, чтобы просто разместить файлы в одной или более существующих иерархий и представить их в образе ISO. Сам ISO9660 ограничивается простым DOS 8.3, но Rock Ridge и расширение Joliet разрешают более длинные названия и/или дополнительные свойства файла. Например, чтобы создать образ проекта, Вы могли бы использовать команду:

```
% mkisofs -o ProjectCD.iso -r ~/project-files ~/project-extras
```

В этом случае, мы создаем образ ISO, который использует атрибуты Rock Ridge (но, в отличие от **-R**, устанавливает более полезные значения, как, например, все файлы доступны для чтения), и содержит все файлы двух каталогов. Другие функции позволили бы нам добавлять загружаемые заголовки к образу, создавать образ HFS, размещать каталоги в указанных местоположениях, отличных от корневого, и более тонко настраивать атрибуты файла.

### Создание файловой системы ISO при помощи cdrecord

Передача образа ISO на записываемый компакт-диск или DVD-диск в настоящее время часто осуществляется посредством связывающих инструментов, таких как интерфейс GUI.

Например, и Gnome и KDE осуществляют запись компакт-дисков через интерфейс файлового менеджера. Существуют также удобные коммерческие программы, но для системного администратора старая добрая команда **cdrecord** представляется наиболее надежной из средств, имеющихся в большинстве современных дистрибутивов, и намного ближе к "стандарту", чем другие программы. Вообще, нужно только определить устройство, на которое вы хотите записать, да файл ISO, который вы хотите записать.

Вы можете также определить как обычно множество параметров процесса записи, например, **-overburn** для компакт-дисков больших, чем 650 МБ или определенную скорость записи для вашего записывающего устройства. Советую прочесть страницу помощи **man** для **cdrecord** для уточнения деталей.

Вы можете обнаружить записывающее устройство при помощи **-scanbus**. Устройство, которое вам нужно, имеет числовой индикатор шины, состоящий из трех цифр, и не является блочным устройством в вашей файловой системе. Например, можно увидеть нечто, похожее на следующее (сокращенно):

### Листинг 3. Обнаружение записывающего устройства

```
% cdrecord -scanbus
[...]
scsibus0:
0,0,0    0) 'ATA      ' 'WDC WD800UE-00HC' '09.0' Disk
0,1,0    1) *
[...]
scsibus1:
1,0,0    100) 'Slimtype' 'DVDRW S0SW-852S' 'PSB2' Removable
CD-ROM
[...]
```

Получив информацию с шины, вы можете записать образ:

```
% sudo cdrecord -overburn -v speed=16 dev=1,0,0
/media/KNOPPIX_V3.6-2004-08-16-EN.iso
```

В этом случае образ небольшого размера, и я знаю, что мое записывающее устройство поддерживает скорость 16x. Вывод команды будет весьма многословным из-за использования флага **-V**, но это помогает в понимании процесса.

### Создание файловой системы ISO при помощи dd

В заключение, иногда бывает необходимо создать новый образ ISO не из каталогов в вашей главной файловой системе, а из уже существующего компакт-диска или DVD-диска. Чтобы сделать образ ISO из компакт-диска, используйте команду **dd**, но обращайтесь к физическому

адресу блочного устройства для компакт-дисков, а не к точке монтирования:

```
% dd if=/dev/cdrom of=project-cd.iso
```

Вы могли бы задаться вопросом, почему не используют команду [cp](#), если цель состоит в том, чтобы просто скопировать байты. Фактически, если вы игнорируете ошибку ввода/вывода, о которой сообщается, когда устройство исчерпывает байты, подлежащие копированию, команда [cp](#), вероятно, может работать. Все же, команда [dd](#) обладает лучшими качествами (она не жалуется, а вместо этого сообщает об итогах своей работы).

**Описание:** Это третий из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите, как монтировать, проверять, создавать новые и исправлять поврежденные файловые системы.

## Управление файловой системой Linux

### Монтирование и отмонтирование при помощи `mount` и `umount`

Одна из особенностей гибкости систем Linux -- это прекрасная детальная настройка контроля, который имеет пользователь, над подмонтированными и отмонтированными файловыми системами. В отличие от Windows и некоторых других операционных систем, местоположения разделов не автоматически закреплены ядром Linux, а присоединены к иерархии корневой файловой системы командой [mount](#). Кроме того, различные типы файловых систем (даже на различных устройствах) могут быть смонтированы в рамках той же самой иерархии. Вы можете отмонтировать конкретный раздел командой [umount](#), назначать любую точку монтирования (например, `/home`) или адрес устройства (например, `/dev/hda7`).

Когда производится восстановление файловой системы, возможность управлять точками монтирования позволяет вам проводить анализ состояния разделов, используя `fsck` или другие инструменты, без риска дальнейшего повреждения уже поврежденной файловой системы. Вы можете также в обычном порядке монтировать файловую систему, используя различные параметры; самые важные из них монтируют файловую систему для использования только в режиме чтения с помощью одного из синонимов `-r` или `-o ro`.

В качестве примера, вы могли бы хотеть заменить местоположение каталога одного пользователя на каталог другого, или из-за повреждения раздела, или просто хотите расширить дисковое пространство, или переместиться на более быстрый диск. Такое изменение можно выполнить, используя:

```
# umount /home # old /dev/hda7 home dir
# mount -t xfs /dev/sda1 /home # new SCSI disk using XFS
# mount -t ext3 /dev/sda2 /tmp # also put the /tmp on SCSI
```

### Монтирование по умолчанию

Для повседневной работы, вам будет удобно, чтобы необходимый конкретный набор подмонтирований осуществлялся автоматически при каждой загрузке системы. Вы управляете точками монтирования, которые происходят в процессе загрузки, прописывая нужные строки конфигурации в файл `/etc/fstab`. Типичная конфигурация могла бы выглядеть так:

### Листинг 4. Пример конфигурации для монтирования при загрузке

```
# <file system> <mount point>  <type>  <options>      <dump>  <pass>
    proc           /proc        proc  defaults        0        0
```

```

/dev/sda3      /          reiserfs notail      0      1
/dev/sda5      none       swap    sw      0      0
/dev/sda6      /home      ext3    rw      0      2
/dev/scd0      /media/cdrom0 udf,iso9660 ro,user,noauto 0      0
/media/Ubuntu-5.04-install-i386.iso /media/Ubuntu_5.04 iso9660
rw,loop 0 0

```

В этом листинге, первое поле (<file system>) - обычно является названием блочного устройства, подлежащего монтированию. Второе (<mount point>) – точка монтирования. В некоторых специальных случаях сначала пишется вовсе не обозначение блочного устройства. Для устройств **supermount**, вы увидите **none**. **/proc** - другой особый случай. Вы могли бы также подмонтировать loopback устройства, которые являются обычно обычными файлами.

Третье (<type>) и четвертое (<options>) поля являются довольно простыми; эти параметры зависят от типа файловой системы и предполагаемого использования. Пятое поле (<dump>) – обычно ноль. Шестое поле (<pass>) должно содержать 1 -- для корневой файловой системы и 2 -- для других файловых систем, которые должны быть проверены при помощи **fsck** во время загрузки системы.

### **Автоматическое монтирование с помощью AMD и automount**

В Linux существует довольно много способов автоматического монтирования ресурсов, которые являются сменными (дискеты, компакт-диски, USB-устройства) или не находятся в состоянии готовности (например, файловых систем NFS). Цель всех этих инструментов схожа, но все они работают немного по-разному.

Инструмент AMD (демон автомонтирования) несколько старше других и работает в пространстве пользователя. AMD периодически запускается, чтобы проверить, не стали ли какие-нибудь подлежащие монтированию файловые системы доступны; в основном, для файловых систем NFS. В большинстве своем, AMD был заменен в дистрибутивах Linux на Autofs, который работает уже как ядерный процесс.

При сборке ядра, которое вы будете использовать, следует включить Autofs. После этого, поведением демона Autofs (обычно **/etc/init.d/autofs**) управляет файл **/etc/auto.master**, который, в свою очередь, ссылается на **map** файл. Например:

```

# Sample auto.master file
# Format of this file: mountpoint map options
/mnt /etc/auto.mnt --timeout=10

```

Файл **/etc/auto.mnt**, на который здесь ссылаются, определяет один или более подкаталогов **/mnt**, которые будут смонтированы (если доступ будет затребован).

Отмонтирование в этом случае произойдет автоматически спустя 10 секунд после последнего доступа.

```

# Sample /etc/auto.mnt
floppy -fstype=auto,rw,sync,umask=002 :/dev/fd0
cdrom -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom
remote -fstype=nfs example.com:/some/dir

```

### **Автоматическое монтирование при помощи supermount и submount**

Инструменты **supermount** и **submount**, работают на уровня ядра (либо включаются в основное ядро, либо в модули ядра), используются для автоматического монтирования сменных ресурсов при обращении к ним. **submount** немного поновее, но в большем количестве дистрибутивов все еще, вероятно, используется **supermount**. Ни один из них не

поможет при удаленном монтировании по NFS, но нет инструментов лучше Autofs, при подключении локальных ресурсов.

Все устройства, требующие автомонтирования, перечислены в файле /etc/fstab. Инструменты используют в /etc/fstab немного разный, но достаточно простой синтаксис. Запускающийся посредством `supermount` /etc/fstab, мог бы содержать следующее:

```
# Example of supermount in /etc/fstab
none /mnt/cdrom supermount fs=auto,dev=/dev/cdrom 0 0
none /mnt/floppy supermount fs=auto,dev=/dev/fd0,--,user,rw 0 0
```

`submount` описывает блочное устройство в его постоянном местоположении, а не как точку монтирования. Например:

```
/dev/cdrom /mnt/cdrom subfs fs=cdfss,ro,users 0 0
/dev/fd0 /mnt/floppy subfs fs=floppyfss,rw,users 0 0
```

### Что сейчас подмонтировано?

Пользователь Linux имеет возможность увидеть список текущих монтирований несколькими способами. Команда `mount` без дополнительных аргументов (или с выбором `-l`) перечисляет монтирования, установленные в настоящее время. Вы можете при желании фильтровать результаты выбором аргумента `-t fstype`.

Основная динамическая информация, касающаяся подмонтированных файловых систем, находится в /etc/mtab. Команды `mount` и `umount` и другие системные процессы обновляют этот файл, чтобы отразить текущий статус; вы должны рассматривать этот файл как “файл только для чтения”. Дополнительную информацию относительно текущего состояния `mount` можно найти в /proc/mounts.

### Специальные инструменты

Инструмент `sync` передвигает незаписанные блоки на диске. У вас нет необходимости использовать этот инструмент в нормальных ситуациях, но вы можете иногда проверять диск, but you can sometimes check for disk problems by checking for a non-zero exit status. Современные и, особенно, журналируемые файловые системы, такие как ext3, Reiser, и JFS эффективно делают синхронизацию при каждой записи.

Если захотите, вы можете вручную запретить или позволить использование свопинга или сопоставить свопинг конкретному устройству. Обычно, любое устройство, отмеченное в /etc/fstab как `swap`, используется для свопинга.

**Описание:** Это третий из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите, как монтировать, проверять, создавать новые и исправлять поврежденные файловые системы.

### Поддержка файловой системы Linux

#### Исправление файловой системы при помощи fsck

Ваш лучший друг в восстановлении поврежденной файловой системы это `fsck`.

То, что мы называем `fsck`, является только началом множества более тонких инструментов `fsck.*`, например: `fsck.ext2`, `fsck.ext3`, или `fsck.reiser`. Можно определить тип точно, используя аргумент `-t`, но `fsck` предпримет усилие понять это самостоятельно. Прочитайте страницы помощи `man` для `fsck` или `fsck.*` для уточнения деталей. Основное, что вам нужно знать, это то, что при использовании аргумента `-a` программа будет пытаться исправить все найденные ошибки.

Вы можете проверить неподмонтированную файловую систему, упоминая местонахождение

устройства, на котором она находится. Например, введите `fsck /dev/hda8`, чтобы проверить неиспользуемый раздел. Вы можете также проверить корневую файловую систему, набрав, например, `fsck /home`, но, как правило, делают это, только если файловая система уже смонтирована как "только для чтения", а не для "чтения-записи".

### Проверка блоков с помощью `badblocks`

Утилита `badblocks` производит проверку качества блочного устройства (или раздела) на более низком уровне, чем это делает `fsck`. `badblocks` исследует надежность блоков на устройстве, записывая и читая тестовые образцы. Используйте аргумент `-n` для более медленного исследования, при котором сохраняются существующие данные. Для совершенно нового раздела без существующих файлов, вы можете (и вероятно должны), использовать аргумент `-w`. Этот вариант просто сообщит вам про плохие блоки, не восстанавливая и не отмечая их.

На практике все же более предпочтительнее использовать для проверки на плохие блоки `fsck.*` для вашей файловой системы. Например: `e2fsck` (также можно вызывать `fsck.ext2`) имеет аргумент `-c` чтобы найти и пометить плохие блоки, которые может обнаружить `badblocks`. ReiserFS имеет аналогичные аргументы `--check` и `--badblocks` (но не совсем автоматические). Прочитайте документацию для вашей конкретной файловой системы по использованию `badblocks`.

### Поиск других программ поддержки

Существует несколько инструментов для исследования и настройки файловых систем Linux. Для работы в обычном режиме вам будет достаточно настроек по умолчанию, но иногда нужно произвести более детальные исследования и масштабные действия, например, на поврежденных системах, или настроить работу в системе в точном соответствии с шаблоном.

Каждый тип файловой системы имеет свой собственный набор инструментов. Для получения более подробной информации проверьте документацию для файловой системы, которую вы используете. Большинство из них имеют сходный набор инструментов. Вот некоторые примеры:

- `dump2fs`: Выходная информация о файловой системе ext2/3.
- `tune2fs`: Регулировка параметров файловой системы для ext2/3.
- `debugfs`: Настройка и проверка файловой системы ext2/3 в интерактивном режиме.
- `debugreiserfs`: Выходная информация о файловой системе Reiser.
- `reiserfstune`: Регулировка параметров файловой системы для Reiser.
- `xfs_admin`: Регулировка параметров файловой системы для XFS.

## Ресурсы

### Научиться

- [Оригинал данной главы](#) на developerWorks.
- В [LPIC Program](#) (Программе LPIC) вы найдете список заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.
- ["Common threads: Advanced filesystem implementor's guide, Parts 1 - 13"](#) (developerWorks, starting June 2001) замечательные руководства по файловым системам Linux.
- Найдите другие ресурсы для разработчиков Linux на [developerWorks Linux zone](#).

## **Получить продукты и технологии**

- [Закажите SEK для Linux](#), набор из двух DVD, содержащих trial-версии последнего программного обеспечения IBM для Linux от DB2®, Lotus®, Rational®, Tivoli® и WebSphere®.
- Постройте ваш следующий проект разработки для Linux с использованием [IBM trial software](#), загрузив его непосредственно с developerWorks.

# Подготовка к экзамену LPI 201: Предоставление доступа к файлам и сервисам

*Intermediate Level Administration (LPIC-2) тема 209*

Дэвид (David) Мертц (Mertz), Developer, Gnosis Software, Inc.

Бред Хантинг, Mathematician, Университет Колорадо

**Описание:** В этом руководстве Бред Хантинг и Дэвид Мертц продолжат готовить к сдаче экзамена 201 в Linux Professional Institute для достижения уровня Intermediate Level Administration (LPIC-2). В этом пятом из восьми руководств вы научитесь, как использовать систему Linux в качестве файла сервера, используя любой поддерживаемый протокол в Linux.

[Больше статей из этой серии](#)

**Дата:** 02.09.2005

**Уровень сложности:** средний

## Прежде чем начать

Посмотрите, чему могут научить эти руководства, и как вы можете извлечь из них максимум пользы.

## Список руководств

Институт [Linux Professional Institute](#)(LPI) производит сертификацию системных администраторов Linux на двух уровнях -- junior и intermediate. Чтобы достигнуть каждого уровня, необходимо сдать два экзамена LPI.

Каждый экзамен содержит несколько тем, каждая тема имеет свой вес. Вес указывает степень важности темы. На экзамене будет больше вопросов по темам с наивысшими весами. Темы и их веса по экзамену LPI 201 следующие:

### Тема 201

ядро Linux (вес 5).

### Тема 202

Загрузка системы (вес 5).

### Тема 203

Файловые системы (вес 10).

### Тема 204

Оборудование (вес 8).

### Тема 209

Предоставление доступа к файлам и сервисам (вес 8). Материал этого руководства.

### Тема 211

Поддержка системы (вес 4).

### Тема 213

Настройка и автоматизация системы (вес 3).

### Тема 214

Разрешение проблем (вес 6).

Linux Professional Institute не рекомендует никаких сторонних материалов подготовки к экзаменам или техник в частности. За подробностями, обращайтесь по адресу[info@lpi.org](mailto:info@lpi.org).

## **Об этом руководстве**

Добро пожаловать в "Предоставление доступа к файлам и сервисам", пятое из восьми руководств, направленных на подготовку вас к сдаче экзамена LPI 201. В этом руководстве вы научитесь, как использовать систему с Linux в качестве файлового сервера, используя любой доступный протокол в Linux.

Руководство организовано в соответствии с требованиями LPI для этой темы следующим образом:

### **2.209.1 Настройка сервера Samba (вес 5)**

Вы сможете настроить сервер Samba для различных клиентов. Эта задача включает настройку скрипта входа для клиентов Samba и настройку сервера WINS nmbd. Также рассмотрено изменение рабочей группы, к которой принадлежит сервер, определение общего каталога в smb.conf, определение общего принтера в smb.conf, использование nmblookup для тестирования функциональности сервера WINS, а также использование команды smbmount для монтирования SMB каталогов на клиенте Linux.

### **2.209.2 Настройка сервера NFS (вес 3)**

Вы сможете создать файл экспорта и определить экспортируемые файловые системы. Эта задача включает редактирование файла экспорта для ограничения доступа определенных узлов, подсетей и сетевых групп. Также рассматривается определение в файле экспорта опций монтирования, настройки отображения пользовательского идентификатора, монтирование NFS на стороне клиента, а также использование опций монтирования для определения нижнего и верхнего уровня фоновых попыток доступа, поддержку сигналов, блокирование и определение размера блока. Вы также научитесь настраивать tcpwrappers для дальнейшей защиты NFS.

Настоящая тема 209 этого экзамена LPI охватывает NFS и Samba. Но если вы системный администратор, проектирующий сервер, то также должны принять во внимание протоколы FTP, SCP/SSH, HTTP или же другие протоколы, которые удовлетворяют вашим требованиям.

Одним из значительных преимуществ Linux, особенно при использовании в качестве сервера, является возможность предоставлять доступ к файлам клиентам системы. На самом деле, именно для предоставления файлов сеть и используется больше всего. Это руководство -- и на самом деле эта серия руководств -- не будет рассказывать о пиринговых файлообменных сетях как BitTorrent. Напротив, это руководство рассматривает только старую архитектуру клиент-сервер: центральный сервер, который предоставляет дисковое пространство множеству клиентов. Даже когда клиенты загружают файлы на сервер, они всегда хранятся и обслуживаются сервером, а не децентрализовано.

Широко используемые протоколы для предоставления доступа к файлам это HTTP (WWW), TFTP (Trivial File Transfer Protocol), FTP (File Transfer Protocol), SCP (Secure Copy Protocol, особая версия SSH), RCP (Remote Copy Protocol, редко используется), NFS (Network File System) и Samba (блок сообщений сервера). HTTP и SSH будут обсуждаться в последующих руководствах по экзамену LPI 202, а также вопросы безопасности FTP. TFTP и RCP имеют особое назначения и\или уже не используются, поэтому о них рассказываться не будет.

Это руководство рассказывает о NFS и Samba и кратко описывает работу FTP. NFS и Samba -- это протоколы обмена файлами, которые позволяют получить прозрачный доступ к удаленным файловым системам. FTP может потребовать программу-клиент FTP, хотя многие рабочие среды или инструменты (в Linux или других системах) прячут подробности и просто представляют удаленные системы как подключенные диски NFS или Samba.

## **Предварительные замечания**

Чтобы извлечь максимум пользы из этого руководства, вы должны иметь общее представление о Linux, а также работающую систему Linux, на которой вы можете исполнять

команды этого руководства.

# Подготовка к экзамену LPI 201: Предоставление доступа к файлам и сервисам

*Intermediate Level Administration (LPIC-2) тема 209*

Дэвид (David) Мертц (Mertz), Developer, Gnosis Software, Inc.

Бред Хантинг, Mathematician, Университет Колорадо

**Описание:** В этом руководстве Бред Хантинг и Дэвид Мертц продолжат готовить к сдаче экзамена 201 в Linux Professional Institute для достижения уровня Intermediate Level Administration (LPIC-2). В этом пятом из восьми руководств вы научитесь, как использовать систему Linux в качестве файла сервера, используя любой поддерживаемый протокол в Linux.

## Настройка сервера NFS

### Использование NFS на стороне клиента

Если сервер правильно настроен, а у клиента есть соответствующие права, то монтирование удаленной файловой системы NFS потребует только использовать команду `mount`:

```
mount -t nfs my.nfs.server.com:/path/on/server /path/on/client
```

или же вставить соответствующую запись в `/etc/fstab`:

```
my.nfs.server.com:/path/on/server /path/on/client nfs rw,soft 0 0
```

Опция `soft` сообщает ядру, что надо посыпать ошибку I/O (EIO) пользовательскому процессу в случае проблем с сетью. Опция по умолчанию `hard` вызовет зависание процесса, если сервер NFS недоступен.

В дополнение, программы помощники `rpc.lockd`, `rpc.statd` и `rpc.quotad` могут быть запущены на стороне клиента и/или сервера.

### Настройка сервера NFS (часть первая)

Для полноценной работы сервера NFS требуется три различные программы, и ещё три могут быть запущены дополнительно.

Когда клиент NFS монтирует файловую систему NFS, он общается с демонами сервера, большинство из которых должны работать отдельно (а не быть запущены из `inetd`):

- `portmap`: Иногда именуется как `portmapper` или `rpc.bind`.
- `rpc.mountd`: Иногда `mounted`.
- `rpc.nfsd`: Иногда `nfsd`.

Дополнительно существуют три опциональные программы помощника: `rpc.lockd`, `rpc.statd`, и `rpc.quotad`, которые, соответственно, обеспечивают глобальное блокирование, ускорение семейства системных вызовов `lstat` (используется `ls -l` и другие), и обеспечивают поддержку квот.

### Настройка сервера NFS (часть вторая)

Все три демона, относящиеся к NFS, используют TCP оболочки (`tcpd`) для управления доступом и поэтому могут потребовать записей в `/etc/host.allow`.

Ни `nfsd` ни `portmap` обычно не требует никакой другой настройки кроме как `/etc/hosts.allow`.

Файлом конфигурации для `mountd` является (косвенно) `/etc(exports`. Он расписывает, какие файловые системы могут монтироваться определенными клиентами. В Linux реализации NFS, `/etc(exports` не обрабатывается прямо `mountd`. Вместо этого команда `exportfs -a` обрабатывает `/etc(exports` и затем пишет результат в `/var/lib/nfs/xtab`, который может быть прочитан `mountd`. Существуют другие флаги у `exportfs`, которые позволяют этим двум файлам быть рассинхронизированными. То есть, вы можете временно добавить или удалить экспортируемые каталоги, не модифицируя постоянных записей в `/etc(exports`.

Администраторам других Unix-подобных серверов следует принять во внимание, что синтаксис файла `/etc(exports` в Linux значительно отличается от того, что используется в SunOS или BSD.

### Настройка `/etc/hosts.allow` и `/etc/hosts.deny`

Файл конфигурации `/etc/hosts.allow` описывает узлы, которые могут подключаться к системе Linux. Этот файл не относится к NFS, но системе необходимо разрешение на подключение к машине, чтобы она смогла использовать сервер NFS. Аналогично, файл `/etc/hosts.deny` это список узлов, которым запрещено подключаться.

Довольно неинтуитивно, сначала идет поиск разрешенных узлов, затем запрещенных узлов, но если машина не найдена, значит ей можно подключиться. Это не значит, что механизмы входа отдельных серверов нельзя изменять, внимательный администратор может запретить все подключения, которые не разрешены явно (немного паранойи не помешает) с помощью:

```
# /etc/hosts.deny
ALL:ALL EXCEPT localhost:DENY
```

Установив запрет в `/etc/hosts.deny` на все (кроме соединений из LOCALHOST) получаем, что возможны только указанные явно соединения. Например:

```
#/etc/hosts.allow
# Allow localhost and intra-net domain to use all servers
ALL : 127.0.0.1, 192.168.
# Let everyone ssh here except 216.73.92.* and .microsoft.com
sshd: ALL EXCEPT 216.73.92. .microsoft.com : ALLOW
# Let users in the *.example.net domain ftp in
ftpd: .example.net
```

### Настройка `/etc(exports`

Вот простой пример файла `/etc(exports`:

```
# sample /etc(exports file / master(rw)
trusty(rw,no_root_squash) /projects proj*.local.domain(rw) /usr
*.local.domain(ro) @trusted(rw) /home/joe
pc001(rw,all_squash,anonuid=150,anongid=100) /pub
(ro,insecure,all_squash)
```

Обычно, `root`(uid 0) на стороне клиента выглядит как `nobody`(uid 65534) на стороне сервера; это называется *сменением root* так как оно позволяет защитить файлы, владельцем которых является `root` (а члены группы/остальные не имеют права на запись) от изменений клиентами NFS. Опция `no_root_squash` отменяет такое поведение, и позволяет пользователю `root` иметь **доверенный** полный доступ к разделу `/`. Такой доступ может быть полезен при установке и настройке программного обеспечения.

Раздел /usr может быть только прочитан всеми узлами, кроме тех, что находятся в сетевой группе "trusted".

Когда /home/joe монтируется с `rw,150,100`, все удаленные пользователи (в независимости от uid/gid) будут иметь uid=150, gid=100. Это может быть полезно, если удаленный клиент NFS представляет собой однопользовательскую рабочую станцию или не поддерживает несколько пользователей (как в случае с DOS).

Обычно, Linux (и другие Unix-подобные операционные системы) резервируют TCP и UDP порты от 1-1023 (так называемые *безопасные порты*) для использования процессами пользователя root. Чтобы удостовериться, что именно root инициировал удаленное подключение NFS, сервер NFS обычно требует, чтобы удаленные клиенты использовали безопасные порты. Это соглашение, однако, не соблюдается некоторыми операционными системами (например Windows). В таких случаях опция `insecure` позволяет клиенту NFS использовать любой порт TCP/UDP. Обычно она требуется при обслуживании клиентов Windows.

## Утилиты NFS

`nfsstat` отображает статистику NFS (клиента и/или сервера) подобно тому, как на локальной машине эти делаю утилиты `iostat` и `vmstat`.

Команда `showmount` запрашивает у `mountd` и показывает, какие клиенты в настоящий момент смонтировали файловые системы. NFS это протокол, не использующий информацию о состоянии и обращение к демону `mountd` происходит нечасто, поэтому вывод `showmount` может быть некорректным. К сожалению, нет способа заставить `showmount` быть точным. Однако, неточность `showmount` заключается лишь в том, что она всегда показывает устаревшие записи.

В этом контексте "не использующий информацию о состоянии" означает, что демон `nfsd`, который работает с реальной информацией, не помнит о том, ни какие файлы были открыты ни какие клиенты смонтировали какие разделы. Каждый запрос (readblock, writeblock и другие) содержит всю необходимую информацию для его выполнения (номер раздела, который предоставляет `mountd`, inode номер, номер блока, read/write/и другие, данные). Протокол HTTP очень похож в этом отношении. Другой стороной не использования информации о состоянии является то, что если сервер перезагрузится, то клиенты заметят только небольшой период времени потери доступа к системе.

**Описание:** В этом руководстве Бред Хантинг и Дэвид Мертц продолжат готовить к сдаче экзамена 201 в Linux Professional Institute для достижения уровня Intermediate Level Administration (LPIC-2). В этом пятом из восьми руководств вы научитесь, как использовать систему Linux в качестве файл сервера, используя любой поддерживаемый протокол в Linux.

## Настройка сервера samba

### Настройка Samba сервера

Сервер Samba обеспечивает доступ к файлам и принтерам (в основном для клиентов Windows). В то время как он может быть запущен из `inetd`, обычно он запускается как отдельный демон `smbd -D`. `nmbd` это сервер имен NetBios (или WINS сервер). Он тоже может запускаться из `inetd`, но обычно его запускают отдельно `nmbd -D`. Samba может работать как сервер в рабочей группе Windows, а также как главный контроллер домена.

Файлом настройки для `smbd` и `nmbd` является `/etc/samba/smb.conf`. Огромное количество параметров описано в man-странице `smb.conf`. Файл `lmhosts` используется для отображения имен NetBios на IP адреса. Его формат схож с (но не идентичен) файлом `/etc/hosts`.

Есть несколько великолепных HOWTO и книг о том, как настраивать Samba. Этот раздел затрагивает несколько основных идей с указанием, где можно найти больше информации.

## Предоставление доступа к домашнему каталогу

Следующий фрагмент smb.conf позволит пользователям получать доступ к домашним (локальным) каталогам с удаленных клиентов Samba:

```
[homes]
comment = Home Directories
browseable = no
```

Обычно эти строки включены по умолчанию в smb.conf.

## Разделение доступа к принтеру с помощью CUPS

Из многочисленных систем печати Unix CUPS одна из самых старых и, возможно, наиболее популярная. В зависимости от вашего дистрибутива, CUPS может как включена, так и отключена по умолчанию в smb.conf. Вот простой пример предоставления совместного доступа к принтеру с помощью CUPS:

```
[global]
load printers = yes
printing = cups
printcap name = cups

[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
public = yes
guest ok = yes
writable = no
printable = yes
printer admin = root

[print$]
comment = Printer Drivers
path = /etc/samba/drivers
browseable = yes
guest ok = no
read only = yes
write list = root
```

CUPS может предоставить ppd (Postscript printer description) файлы и драйверы Windows клиентам, которые, будучи правильно установлены, позволят удаленным пользователям использовать весь список возможностей принтеров (выбор цветной или черно-белой печати, разрешения, выбор размера бумаги, двусторонней или односторонней печати и так далее). Традиционные системы печати Unix довольно неудобны. Смотрите man-страницу cupsaddsmb для получения более подробной информации.

## Аутентификация

Samba (в отличие от NFS) требует, чтобы каждый пользователь прошел аутентификацию. Как и с другими сетевыми сервисами, требующими аутентификацию, следует удостовериться, что пароли не передаются по сети незашифрованными. Смотри раздел шифрование паролей в man-странице smb.conf.

Существует несколько механизмов, с помощью которых Samba может аутентифицировать

удаленных пользователей (клиентов). По своей природе, большинство из них несовместимо со стандартным хэшем паролей Unix. Никогда не передавайте пароли по сети в открытом виде. Это всегда плохая идея.

Предположим, что вы шифруете пароли в сети, **smbpasswd** обычно будет использоваться для установки пользователям начальных паролей Samba. Опция "Unix password sync" позволит **smbpasswd** изменить пароли Unix, когда пользователи меняют свои пароли Samba.

В другом случае настроенный модуль **param\_smb** может аутентифицировать пользователей Linux используя базу данных Samba. Если этого недостаточно, то можно настроить LDAP для аутентификации пользователей Samba и/или Linux.

### Отладка Samba

При настройке сервера Samba довольно полезной может оказаться команда **testparm** (также называемая **smbtestparm**). Она просмотрит smb.conf и сообщит об ошибках.

Команда **nmblookup** делает для Samba тоже, что и **nslookup** для DNS; она делает запрос справочника NetBios. На man-странице **nmblookup** можно узнать подробности.

### Настройка клиентов Samba

Команда **smbclient** предоставляет доступ к общим папкам Samba. Прозрачный доступ к папкам SMB довольно непрост; на man-странице **smbmount** можно узнать подробности.

**Описание:** В этом руководстве Бред Хантинг и Дэвид Мертц продолжат готовить к сдаче экзамена 201 в Linux Professional Institute для достижения уровня Intermediate Level Administration (LPIC-2). В этом пятом из восьми руководств вы научитесь, как использовать систему Linux в качестве файл сервера, используя любой поддерживаемый протокол в Linux.

## Настройка сервера File Transfer Protocol

### Об FTP

FTP это старый и широко используемый сетевой протокол. FTP обычно работает на двух раздельных портах -- 20 и 21. Порт 21 используется для контрольного потока (передающий информацию о входе в систему и команды) в то время как порт 20 используется для потока данных, по которому идет передача файлов.

В общем FTP это не очень безопасный протокол по той причине, что в режиме по умолчанию управляющий поток -- а значит логин и пароль -- передается в открытом виде. Поток данных также передается открыто, также как в NFS и Samba (для усиления безопасности SSH/SCP лучший выбор). Можно перенаправить управляющий поток FTP через SSH, таким образом, защитив его.

Традиционные клиенты FTP предоставляют собственные среды для работы, по которым передаются команды и настраиваются соединения. Иногда GUI интерфейсы поставляются вместе с клиентами для более удобной работы с файлами. Однако в наши дни многие инструменты включают FTP -- начиная от менеджера файлов до текстовых редакторов, они часто работают с файлами, лежащими на FTP сервере.

### Анонимный FTP

Для тех применений, где FTP используется чаще всего, безопасность не является проблемой. Чаще всего FTP сервера используются как "анонимные FTP" -- то есть размещенные на них данные доступны миру и не требуют большой безопасности. По соглашению пользователь *anonymous* может получить доступ к файлам, предоставив произвольный пароль (обычно адрес почты), который не проверяется. Иногда имя пользователя/пароль требуются, полученная комбинация не проходит более сложную степень аутентификации (например для

людей, которые хотят стать добровольцами в каком-либо проекте).

Большинство Web-браузеров и файловых менеджеров, а также инструментов прозрачно поддерживают FTP сервера. Часто этим инструментам требуется FTP URL для получения файла (а также для загрузки файла на сервер). Например, инструмент командной строки `wget` скачает файл с FTP сервера с помощью следующей команды:

```
$ wget ftp://example.net/pub/somefile  
$ wget ftp://user:passwd@example.net/pub/somefile
```

Файловые менеджеры часто монтируют FTP сервера как локальную систему или диски NFS или Samba (но не полностью точно так же, не используйте `mount` и `/etc/fstab`; такие псевдоразделы обычно именуются по своему URL).

## Выбор FTP серверов

FTP старый и повсеместно внедренный, поэтому существует огромное число его реализаций и установок на различных Linux дистрибутивах. Настройка выбранного FTP сервера потребует от вас обращения к руководству по установке.

Некоторые популярные Linux FTP сервера:

- wu-ftp.d.
- vsftpd.
- ProFTPD.
- BSD ftpd.
- TUX FTP.

Многие другие реализации не так популярны. В большинстве случаев настройка конкретного сервера будет находиться в файле `/etc/FOOftpd.conf` (для соответствующего сервера "FOO"). Мне нравится vsftpd, так как он быстр и довольно надежен в плане защищенности ("vs" означает "очень защищенный").

## Простой пример настройки FTPd

Синтаксис настройки каждого сервера будет различен. Но некоторые концепции, взятые из `/etc/vsftpd.conf` позволят понять типы опций остальных серверов. Что касается vsftpd, то у каждой опции имеется формат вида `option=value` с использованием знака решетки для обозначения комментариев. Большинство остальных файлов настроек FTPd похожи.

- `anonymous_enable`: управляет возможностью входа пользователя `anonymous`.
- `anon_world_readable_only`: Когда включено, то только `anonymous` пользователям позволено скачивать файлы для чтения.
- `chroot_local_user`: Если включено, то локальные пользователи будут помещены в `chroot()` окружение в своих домашних каталогах после входа.
- `pasv_enable`: Должен ли сервер использовать "пассивный FTP" режим, в котором клиенты инициируют порты (помогает, когда у клиентов есть файервол).
- `ssl_enable`: Если включено, то vsftpd будет поддерживать SSL соединения.
- `tcp_wrappers`: Если включено, то все входящие соединения будут проходить контроль доступа (как `/etc/hosts.allow` и `/etc/hosts.deny`).

## Запуск FTP сервера

В простейшем случае вы можете запустить FTP сервер, так же как и любой другой демон:

```
% sudo vsftpd
```

Теперь сервер будет принимать входящие соединения согласно правилам в его файле настроек. Вы также можете запустить FTP сервер из "сетевого супер-сервера", такого как inetd или xinetd. Руководства LPI 202 расскажут о супер-серверах.

Запуск демона отдельно, даже в скриптах загрузки, как для какого-то определенного уровня загрузки, так и в /etc/rcS.d/, даст вам точный контроль над поведением FTP сервера.

## Ресурсы

### Научиться

- В разделе [Программа LPIC](#) вы найдете списки заданий, простые вопросы и подробное описание требований для трех уровней сертификации системных администраторов в Linux Professional Institute.
- "[Введение в Samba](#)" (developerWorks, Июнь 2000) это серия из двух частей, в которой рассказано, как установить и настроить сервер Samba.
- Другие ресурсы для разработчиков Linux вы можете найти в разделе [developerWorks Linux](#).

### Получить продукты и технологии

- Создайте ваш следующий проект на Linux с помощью [пробных программ IBM](#), доступных для прямого скачивания с developerWorks.

# Учебник для экзамена LPI 201: Поддержка системы

*Администрирование, средний уровень (LPIC-2) тема 211*

Дэвид Мерц, автор, Gnosis Software, Inc.

**Описание:** Это шестой из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите вопросы системного журналирования, упаковывания программного обеспечения и резервного копирования.

[Больше статей из этой серии](#)

**Дата:** 02.09.2005

**Уровень сложности:** средний

## Перед тем как начать

Узнайте, чему эти обучающие программы могут научить вас, и как извлечь из них больше пользы.

## Об этой серии учебных пособий

Linux Professional Institute (LPI) производит сертификацию системных администраторов Linux двух уровней: для начинающих и среднего уровня. Чтобы получить сертификат каждого уровня, вы должны сдать два экзамена LPI.

Каждый из экзаменов состоит из нескольких тем, и каждая тема имеет свой рейтинг. Рейтинги указывают относительную важность каждой темы. Грубо говоря, вы вправе ожидать больше вопросов на экзамене на темы с более высоким рейтингом. Темы и их рейтинги для экзамена LPI 201:

### Тема 201

Ядро Linux (рейтинг 5).

### Тема 202

Запуск системы (рейтинг 5).

### Тема 203

Файловая система (рейтинг 10).

### Тема 204

Оборудование (рейтинг 8).

### Тема 209

Совместное использование файлов и служб (рейтинг 8).

### Тема 211

Поддержка системы (рейтинг 4). Тема данной главы.

### Тема 213

Настройка работ и автоматическое выполнение заданий (рейтинг 3).

### Тема 214

Устранение неполадок (рейтинг 6).

Linux Professional Institute® не приветствует использование для подготовки к экзаменам материалов и технологий от третьих лиц. Для более подробной информации обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## **Об этом руководстве**

Добро пожаловать в главу "Поддержка системы", шестой из восьми учебников, разработанных для подготовки к экзамену LPI 201. На этом этапе вы изучите основные концепции системного журналирования, упаковывания программного обеспечения и стратегии резервного копирования.

Это руководство организовано по следующим разделам, которые LPI относит к данной теме:

### **2.211.1 Журналирование системных сообщений (рейтинг 1)**

В этой части рассказывается, как конфигурировать syslogd, чтобы он мог работать как сетевой сервер журналирования, а также посыпать системные сообщения центральному серверу, как журналировать удаленные соединения и использовать grep и другие утилиты для работы с текстом, чтобы автоматизировать анализ продуктов журналирования.

### **2.211.2 Упаковывание программного обеспечения Packaging software (рейтинг 1)**

В этой части рассказывается, как собирать пакеты. Вы узнаете, как два главных формата упаковки, RPM и DEB, используются для сборки (и пересборки) пакетов.

### **2.211.3 Резервное копирование (рейтинг 2)**

Вы сможете создать удаленное хранилище резервного копирования.

Это учебное пособие для подготовки к экзамену LPI является блоком из нескольких тем, которые не попадают в другие категории. Журналирование системных сообщений и анализ файлов системных журналов - достаточно важные задачи для системного администратора, чтобы была необходимость ознакомиться с ними. Также у хорошего системного администратора должна существовать разумная стратегия резервного копирования данных с использованием стандартных инструментов Linux.

Не каждый системный администратор должен будет создавать специальные пакеты программ, но для администраторов многократных инсталляций установка специальных внутренних программных пакетов может входить в обязанности. Эта обучающая программа рассматривает форматы пакетов Debian и RPM, а также затрагивает основные архиваторы.

## **Требования**

Чтобы работа с этим учебником была максимально плодотворна, вы уже должны быть достаточно хорошо знакомы с ОС Linux и иметь компьютер с ОС Linux, чтобы на практике иметь возможность самостоятельно проверять работу команд, описанных в этом руководстве.

# **Учебник для экзамена LPI 201: Поддержка систем**

*Администрирование, средний уровень (LPIC-2) тема 211*

Дэвид Мертц, автор, Gnosis Software, Inc.

**Описание:** Это шестой из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите вопросы системного журналирования, упаковывания программного обеспечения и резервного копирования.

## **Журнилирование системных сообщений**

### **О журнилировании**

Многие процессы и серверы под Linux записывают информацию, касающуюся изменения состояния, в так называемые "log файлы" (файлы системных журналов). Эти файлы системного журнала обычно находятся в каталоге `/var/log/` и часто начинаются с фиксирования момента времени, указывающего, когда описанное событие произошел. Но как бы там ни было, нет ни условленной последовательности записи, ни определенного формата файлов системного журнала. Одна особенность, на которую вы в значительной степени все-таки можете рассчитывать -- то, что файлы системного журнала Linux являются простыми файлами ASCII, и в каждой строке файла содержится одно "событие". Часто (но не всегда) файлы системного журнала содержат относительно последовательный набор полей или разграниченных таблицей областей данных.

Некоторые процессы, особенно службы интернет, управляют файлом системного журнала, записываемого в пределах их собственного процесса. На самом деле, запись в файл системного журнала -- это, всего лишь, приобщение к открытому файлу дополнительных данных. Но многие программы (особенно демоны и процессы [cron](#)) используют стандарт syslog API, чтобы позволить управлять процессом журнилирования демонам syslogd или klogd.

### **Анализ файлов системных журналов**

Как именно вы будете анализировать то, что записано в log файле, зависит от формата, который используется. Для файлов системного журнала с форматом таблицы, вероятно, будут полезны инструменты типа *cut*, *split*, *head* и *tail*. *grep* -- самый мощный инструмент для обнаружения и фильтрации интересующего вас содержания для всех файлов системного журнала. Для более сложных задач обработки вы, вероятно, будете использовать *sed*, *awk*, *perl* или *python*.

Хорошим введением в область инструментов обработки текста, которые вы должны будете и, скорее всего, будете использовать для обработки и анализа файлов системного журнала, мог бы быть учебник IBM developerWorks Дэвида об утилитах обработки текста GNU.

Существует также множество инструментов высокого уровня, чтобы обрабатывать файлы системного журнала, но эти инструменты обычно зависят от дистрибутива и/или не стандартизированы (но часто являются свободными).

## **Журнилирование системных сообщений с помощью syslogd и klogd**

Демон klogd перехватывает и журнилирует сообщения ядра Linux. Как правило, klogd использует более общие способности syslogd, но в специальных случаях может записывать сообщения непосредственно в файл.

Общий демон syslogd обеспечивает протоколирование для многих программ. Каждое системное сообщение содержит, по крайней мере, поле для времени, поле для имени host'а и обычно поле для имени программы. Поведением syslogd управляет файл конфигурации `/etc/syslog.conf`. Сообщения от приложений (включая ядро) могут быть зарегистрированы в файлах, которые обычно находятся в `/var/log/` или удаленно по сетевому сокету.

### **Конфигурирование /etc/syslog.conf**

Файл `/etc/syslog.conf` содержит ряд правил, по одному в каждой строчке. Пустые строки и строки, начинающиеся с "\*" игнорируются. Каждое правило состоит из двух полей, разделенных пробелом, поля отбора и поля действия. Поле отбора, в свою очередь, содержит одну или более разделенных точкой пар средство - приоритет. Средство - подсистема, которая

хотела бы регистрировать свои сообщения и может иметь значения: `auth`, `authpriv`, `cron`, `daemon`, `ftp`, `kern`, `lpr`, `mail`, `mark`, `news`, `security`, `syslog`, `user`, `uucp` и `local0` через `local7`.

Приоритеты имеют определенный порядок, и данному приоритету соответствует значение "этот или выше", если в начале используется "=" (или "!="). Приоритеты в порядке возрастания: `debug`, `info`, `notice`, `warning` или `warn`, `err` или `error`, `crit`, `alert`, `emerg` или `panic` (несколько имен имеют синонимы). `none` означает, что нет приоритета.

И средства и приоритеты могут принимать значение "\*" (wildcard). Несколько средств могут быть разделены запятой, и разные селекторы могут быть разделены точкой с запятой.

Например:

```
# from /etc/syslog.conf
# all kernel messages
kern.*           -/var/log/kern.log
# `catch-all' logfile
*.=info;*.=notice;*.=warn; \
    auth,authpriv.none; \
    cron,daemon.none; \
    mail,news.none      -/var/log/messages
# Emergencies are sent to everybody logged in
*.emerg          *
```

## Конфигурирование удаленного журналирования системных сообщений

Чтобы сделать возможным удаленное журналирование сообщений `syslogd` (сообщения реальных приложений, но обработанные `syslogd`), вы должны сначала разрешить службе "syslog" прослушивание машин и пересылку. Для этого нужно добавить следующую строчку к каждому файлу конфигурации `/etc/services`:

```
syslog      514/UDP
```

Чтобы настроить локальный `syslogd` для отправления сообщений удаленному хосту, вы определяете обычное средство и приоритет, но при обозначении действия начинаете с символа "@" для адреса хоста. Хост можно сконфигурировать обычным способом либо в файле `/etc/hosts` либо через DNS (имя хоста не обязательно должно быть разрешено через `resolving`, когда `syslogd` запускается впервые). Например:

```
# from /etc/syslog.conf
# log all critical messages to master.example.com
*.crit          @master.example.com
# log all mail messages except info level to mail.example.com
mail.*;mail.!=info  @mail.example.com
```

## Ротация файлов системных сообщений

Вряд ли вы захотите, чтобы ваши файлы системных журналов неограниченно росли. Можно использовать `logrotate`, чтобы заархивировать старую зарегистрированную информацию. Обычно `logrotate` выполняется как ежедневное задание `cron`. `logrotate` позволяет выполнять автоматическую ротацию, сжатие, удаление и отправку по почте файлов системного журнала. Каждый файл системного журнала может быть обработан ежедневно, еженедельно, ежемесячно или только тогда, когда становится слишком большим.

Поведением `logrotate` управляет файл конфигурации `/etc/logrotate.conf` (или специально

определенный другой файл). Файл конфигурации может содержать и глобальные опции и опции, определенные для файла. Вообще, заархивированные сообщения сохраняются в течении конечного периода времени, и им присваиваются последовательные резервные названия. Например, одна из моих систем содержит следующие файлы в результате ее списка ротации:

```
- rw- r----- 1 root adm 4135 2005-08-10 04:00 /var/log/syslog
- rw- r----- 1 root adm 6022 2005-08-09 07:36 /var/log/syslog.0
- rw- r----- 1 root adm 883 2005-08-08 07:35 /var/log/syslog.1.gz
- rw- r----- 1 root adm 931 2005-08-07 07:35 /var/log/syslog.2.gz
- rw- r----- 1 root adm 888 2005-08-06 07:35 /var/log/syslog.3.gz
- rw- r----- 1 root adm 9494 2005-08-05 07:35 /var/log/syslog.4.gz
- rw- r----- 1 root adm 8931 2005-08-04 07:35 /var/log/syslog.5.gz
```

**Описание:** Это шестой из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите вопросы системного журналирования, упаковывания программного обеспечения и резервного копирования.

## Упаковывание программного обеспечения

### В начале был tarball

Для настройки программного обеспечения в Linux требуется гораздо меньше усилий, чем вы могли бы подумать. В Linux есть довольно четкий стандарт того, где файлы различных типов должны находиться, и установка специального программного обеспечения, в сущности, требует простого размещения правильных файлов в правильных местах.

Инструмент Linux *tar* (для "ленточных архивов", даже если не нужно, и обычно не нужно, использует формат ленты) прекрасно подходит для создания архивов файлов с указанными местоположениями в файловой системе. Для дистрибутива вы можете сжать архив *tar* при помощи *gzip* (или *bzip2*). См. заключительную главу этого руководства о резервном копировании для получения дополнительной информации об этих утилитах. Сжатый *tar* архив обычно обозначают расширениями *.tar.gz* или *.tgz* (или *.tar.bz2*).

Ранние дистрибутивы Linux (и некоторые современные, такие как Slackware) используют простой *tar* архив как механизм развертывания дистрибутива. Для специализированных дистрибутивов собственного программного обеспечения систем Linux, поддерживаемых централизованно, такой способ остается самым доступным.

### Специальные форматы архивов

Различные языки программирования и другие инструменты, появляющиеся вместе со специальными дистрибутивами, обычно не имеют предпочтений между дистрибутивами Linux и даже другими операционными системами. Python имеет свой собственный формат архива и инструменты *distutils*; Perl использует архивы CPAN; Java использует *.jar* файлы; Ruby использует *.gem*. Большинство неязыковых приложений имеют стандартную систему для установки плагинов или других инструментов для расширения базовых возможностей приложений.

Например, чтобы установить пакет Python вы можете использовать такой формат пакета как DEB или RPM, но часто разумнее следовать тому стандарту, для которого пакет создан.

Конечно, для системных утилит и приложений и для большинства пользовательских приложений, тот стандарт, который используется, и есть стандарт дистрибутива Linux. Но для некоторых инструментов, написанных на специальных языках программирования, проще использовать какой-то ваш специальный инструмент вместо предложенного дистрибутивом или платформой (или внутреннее, или внешнее использование является конечной целью).

### "Большая двойка" форматов пакетов

Есть два главных формата пакетов, используемые дистрибутивами Linux: Redhat Package Manager (RPM) и Debian (DEB). Оба они подобны в своих целях, но различны в деталях. Это форматы для "расширенного" архива файлов. Продвинутость, обеспеченная этими форматами пакетов, включает аннотации для номера версии, зависимости одного приложения от других приложений или библиотек, удобочитаемые описания инструментов пакета и общий механизм для управления установкой, обновлением или удалением инструментов пакета.

В формате DEB, вложенный конфигурационный файл *control* содержит большинство метаданных пакета. Для файлов RPM, эту роль играет файл *spec*. Более полные детали правильного создания пакетов в любом формате находятся вне этого учебного пособия, а здесь мы выделим только основы.

### Что находится в .deb файле?

Пакет DEB создан при помощи инструмента архивирования, родственника tar -- ar (или другим более высокуюровневым инструментом, который использует *ar*). Поэтому мы можем использовать только *ar*, чтобы увидеть, что находится в .deb файле. Обычно используют высокуюровневые инструменты, такие как *dpkg*, *dpkg-deb* или *apt-get*, чтобы фактически работать с пакетом DEB. Например:

```
% ar tv unzip_5.51-2ubuntu1.1_i386.deb
rw-r--r-- 0/0      4 Aug  1 07:23 2005 debian-binary
rw-r--r-- 0/0    1007 Aug  1 07:23 2005 control.tar.gz
rw-r--r-- 0/0  133475 Aug  1 07:23 2005 data.tar.gz
```

Debian-бинарный файл просто содержит версию DEB (в настоящее время 2.0). Архив data.tar.gz содержит собственно файлы приложений - исполняемые, документацию, страницы помощи, файлы конфигурации и так далее.

Архив control.tar.gz представляется самым интересным. Давайте посмотрим на пакет DEB, который мы выбрали:

```
% tar tvfz control.tar.gz
drwxr-xr-x root/root      0 2005-08-01 07:23:43 .
-rw-r--r-- root/root    970 2005-08-01 07:23:43 ./md5sums
-rw-r--r-- root/root    593 2005-08-01 07:23:43 ./control
```

Как мы могли бы ожидать, *md5sums* содержит смесь шифров всех файлов дистрибутива в целях проверки. Файл *control* сообщает, где находятся метаданные. В некоторых случаях вы могли бы также захотеть включить в control.tar.gz скрипты *postinst* и *prerm*, чтобы принять специальные меры после установки или, соответственно, перед удалением.

### Создание control-файла DEB

Инсталляционные скрипты могут делать то, что мог бы делать скрипт shell. (Посмотрите на некоторые примеры в существующих пакетах, чтобы понять, что имеется в виду.) Но такие скрипты являются дополнительными, и часто в них нет необходимости. Для .deb пакета необходим его control-файл. Формат этого файла содержит различные поля метаданных, и лучше всего это можно проиллюстрировать на примере:

```
% cat control
Package: unzip
Version: 5.51-2ubuntu1.1
Section: utils
Priority: optional
Architecture: i386
```

```

Depends: libc6 (>= 2.3.2.ds1-4)
Suggests: zip
Conflicts: unzip-crypt (<< 5.41)
Replaces: unzip-crypt (<< 5.41)
Installed-Size: 308
Maintainer: Santiago Vila <sanvila@debian.org>
Description: De-archiver for .zip files
  InfoZIP's unzip program. With the exception of multi-volume archives
  (ie, .ZIP files that are split across several disks using PKZIP's /& option),
  this can handle any file produced either by PKZIP, or the corresponding
  InfoZIP zip program.

This version supports encryption.

```

В основном, кроме специальных случаев, ваш control-файл должен выглядеть точно также, как и этот. Для неконкретизированных типов процессоров – либо скрипты, либо пакеты документации, либо исходный код – используют **Architecture: all**.

## Создание DEB пакета

Создание пакета DEB происходит при помощи *dpkg-deb*. Мы не можем здесь раскрыть все секреты изготовления хороших пакетов, но основная идея в том, что необходимо создать рабочий каталог *./debian/*, и поместить в него необходимое содержание перед запуском *dpkg-deb*. Вы можете пожелать установить соответствующие права на ваши файлы после установки. Например:

```

% mkdir -p ./debian/usr/bin/
% cp foo-util ./debian/usr/bin                      # copy executable/script
% mkdir -p ./debian/usr/share/man/man1
% cp foo-util.1 ./debian/usr/share/man/man1          # copy the manpage
% gzip --best ./debian/usr/share/man/man1/foo-util.1
% find ./debian -type d | xarg chmod 755            # set dir permissions
% mkdir -p ./debian/DEBIAN
% cp control ./debian/DEBIAN    # first create a matching 'control'
% dpkg-deb --build debian      # create the archive
% mv debian.deb foo-util_1.3-1all.deb # rename to final package name

```

## Еще о создании DEB пакета

В предыдущем примере можно увидеть, что локальная структура каталогов внутри *./debian/* организовывается так, чтобы соответствовать намеченной инсталляционной структуре. Для создания хорошего пакета необходимо еще несколько пунктов.

- Вообще, вы должны создать файл, названный с частью названия вашего дистрибутива *./debian/usr/share/doc/foo-util/copyright* (добавьте к названию пакета).
- Хорошее дело – создавать файлы *./debian/usr/share/doc/foo-util/changelog.gz* и *./debian/usr/share/doc/foo-utils/changelog.Debian.gz*.
- Инструмент *lintian* позволяет убедиться, что при создании пакета не было допущено ни одной из стандартных ошибок, он проверит пакет DEB на предмет сомнительных свойств. Не все, на что *lintian* жалуется, необходимо обязательно исправить; но если вы собираетесь расширять дистрибутив, будет неплохо, если вы уберете все проблемы, которые возникли.
- Инструмент *fakeroot* позволит вам создавать пакеты не как пользователь *root* и полезен для того, чтобы собирать пакеты с правильным владельцем. Обычно нужны инструменты, установленные как *root*, а не как индивидуальный пользователь,

который, может быть, собирал пакет (lintian предупредит об этом). Вы можете достигнуть этого с:

```
% fakeroot dpkg-deb --build debian
```

### Что находится в .rpm файле?

RPM придерживается немного другой стратегии при создании пакетов, чем DEB. Его файл конфигурации называют *spec*, а не *control*, но файл *spec* делает больше, чем файл *control*. Все детали шагов, необходимых для предустановки, постустановки, предудаления и непосредственно установки, содержатся, как вложенные скрипты, в конфигурации *spec*. Фактически, формат *spec* даже содержит макросы для общих действий.

Пакеты RPM создают при помощи утилиты *rpm -b*. Например:

```
% rpm -ba foo-util-1.3.spec # perform all build steps
```

Этот процесс сборки пакетов базируется не на специфически названных каталогах, как в случае с DEB, а скорее на каталогах, прописанных в более сложном файле *spec*.

### Создание метаданных RPM

Основные метаданные в RPM очень похожи на аналогичные в DEB. Например, *foo-util-1.3.spec* мог бы содержать что-то похожее на следующее:

```
# spec file for foo-util 1.3
Summary: A utility that fully foos
Name: foo-util
Version: 1.3
Release: 1
Copyright: GPL
Group: Application/Misc
Source: ftp://example.com/foo-util.tgz
URL: http://example.com/about-foo.html
Distribution: MyLinux
Vendor: Acme Systems
Packager: John Doe <jdoe@acme.example.com>

%description
The foo-util program is an advanced fooer that combines the
capabilities of OneTwo's foo-tool and those in the GPL bar-util.
```

### Скрипты в RPM

Несколько разделов в RPM *spec*-файле могут содержать небольшие скрипты shell. Они включают:

- **%prep**: Шаги, которые нужно предпринять, чтобы получить готовую сборку, например, удалить более ранние сборки. Часто следующий макрос полезен и достаточен:  

```
%prep
%setup
```
- **%build**: Шаги, чтобы фактически собрать пакет. Если вы используете средство *make*, то можно написать:  

```
%build
```

```
make
```

- **%install:** Шаги, чтобы установить пакет. И опять, если вы используете `make`, это могло бы означать:

```
%install  
make install
```

- **%files:** Вы *должны* включать список файлов, которые являются частью пакета. Даже если бы ваш Makefile использовал эти файлы, то менеджер пакетов (rpm) не будет знать о них, если вы не включите их сюда:

```
%files  
%doc README  
/usr/bin/foo-util  
/usr/share/man/man1/foo-util.1
```

**Описание:** Это шестой из восьми учебников, с помощью которых David Mertz продолжает готовить вас к экзамену Linux Professional Institute® Администрирование, средний уровень (LPIC-2) 201. Здесь вы изучите вопросы системного журналирования, упаковывания программного обеспечения и резервного копирования.

## Операции резервного копирования

### О резервном копировании

Первое правило при создании резервных копий: Сделайте это! Слишком легко при администрировании сервера или настольной машины Linux пренебречь резервным копированием при составлении списка ваших потребностей.

Самый простой способ наладить систематическое резервное копирование – это настроить его, как одну из задач [cron](#). См. Тему 213 нашего учебного пособия, где обсуждается конфигурирование *crontab*. Некоторым образом расписание резервного копирования зависит от инструментов, с помощью которых вы собираетесь его производить, и носителей, которые вы собираетесь использовать.

Резервное копирование на ленту – традиционный способ, и ленточные устройства продолжают предлагать наибольшую вместимость относительно недорогих носителей. Но в последнее время повсеместно стали использоваться записываемые и перезаписываемые компакт-диски и DVD-диски, и часто становится разумным использовать такие сменные носители для резервных копий.

### Что подлежит резервному копированию

Что в системе Linux хорошо, так это то, что в ней используется определенное, иерархическое построение файлов. Как следствие, вам нет необходимости часто делать копию всей иерархии файловой системы; большая часть ее может быть заново установлена с вашего дистрибутива Linux достаточно легко. В больших структурах образ мастер-сервера мог бы быть взят за основу системы Linux, которая, в свою очередь, уже могла бы быть настроена при помощи восстановления нескольких специально отобранных файлов, которые и подлежали бы резервному копированию.

В основном, то, что вы хотите сохранить – это каталоги `/home/`, `/etc/`, `/usr/local/` и, возможно, `/root/` и `/boot/`. Часто хотят также сделать копию некоторой части `/var/`, а именно `/var/log/` и `/var/mail/`.

## Резервное копирование при помощи cp и scp

Может быть, самый простой способ сделать резервную копию – использовать `cp` или `scp` с опцией `-r` (рекурсивно). Первая из этих команд копирует на локальные ресурсы (но включая монтирования NFS), а вторая может копировать на удаленные серверы надежно зашифрованным способом. В любом случае, вы должны иметь смонтированный ресурс с достаточным количеством свободного места, чтобы разместить файлы, которые вы хотите скопировать. Чтобы быть действительно застрахованным, резервная копия ваших данных должна быть размещена на другом физическом ресурсе.

Копирование при помощи `cp` или `scp` может быть только одной из частей большого списка резервного копирования. Здесь можно схитрить с помощью утилиты `find`, чтобы выяснить, какие файлы были изменены недавно. В следующем простом примере мы копируем из `/home/` все файлы, которые были изменены за день:

```
#!/bin/bash

# File: backup-daily.sh
# ++ Run this on a daily cron job ++
#-- first make sure the target directories exist
for d in `find /home -type d` ; do mkdir -p /mnt/backup$d ; done
#-- then copy all the recently modified files (one day)
for f in `find /home -mtime -1` ; do cp $f /mnt/backup$f ; done
```

Команда `cp -u` немного похожа, но более зависима от целостности файловой системы, куда происходит копирование между backup'ами. Существует рецепт, при котором эта команда прекрасно работает: замените точку монтирования `/mnt/backup` на другой адрес NFS. Найденный способ также хорошо работает и с `scp`, если вы зададите для удаленного ресурса информацию, необходимую для входа в систему.

## Резервное копирование при помощи tar

Хотя `cp` и `scp` подходят для создания резервной копии, инструмент `tar` имеет более широкое использование, так как разработан специально для создания ленточных архивов. Несмотря на название, `tar` одинаково подходит для создания, как простого `.tar` файла, так и для записи необработанных данных на ленточное устройство. Например, вы могли бы создать копию на ленточном устройстве, используя команду:

```
% tar -cvf /dev/rmt0 /home      # Archive /home to tape
```

Внесением небольших изменений направляем выход в файл:

```
% tar -cvf /mnt/backup/2005-08-12.tar /home
```

Фактически, так как инструмент `gzip` может быть включен в поток, вы можете легко сжать архив в процессе создания:

```
% tar -cv /home | gzip - > /mnt/backup/2005-08-12.tgz
```

Вы можете комбинировать `tar` таким же образом, как было показано для `cp` или `scp`. Чтобы составить список файлов на ленточном устройстве, вы могли бы использовать:

```
% tar -tvf /dev/rmt0
```

Чтобы восстановить определенный файл:

```
% tar -xvf /dev/rmt0 file.name
```

## Резервное копирование при помощи cpio

Утилита *cpio* - это мощная производная от tar. *cpio* работает с архивами tar, но также и с другими форматами и, кроме того, имеет множество встроенных опций. Сріо можно использовать с большим количеством аргументов, позволяющих отфильтровать скопированные файлы, и даже имеет встроенные аргументы, поддерживающие удаленное резервное копирование (вместо того, чтобы использовать канал с применением scp и др.) Главное преимущество, которое имеет сріо по сравнению с tar, что вы можете как добавить файлы к уже имеющемуся архиву, так и удалить файлы из архива.

Вот несколько примеров использования сріо:

- Создаем файловый архив на ленточном устройстве: % `find /home -print | cpio -ocBv /dev/rmt0`.
- Составляем список записей в файловом архиве на ленточном устройстве: % `cpio -itcvB < /dev/rmt0`.
- Восстанавливаем файл из ленточного устройства: % `cpio -icvdBum file.name < /dev/rmt0`.

## Резервное копирование при помощи dump и restore

Чтобы сделать копию всей файловой системы сразу, иногда используется ряд инструментов, типа *dump* и *restore* (или их производных). К сожалению, эти инструменты являются специфическими для разных типов файловых систем и не всегда пригодны. Например, оригинальные *dump* и *restore* подходят только для ext2/3 файловых систем, в то время как инструменты *xfsdump* и *xfsrestore* используются для файловых систем XFS. Не каждый тип файловой системы имеет подходящую версию инструмента, но даже если они работают, аргументы могут быть разными.

Полезно быть осведомленным об этих утилатах, но они не равнозначны для разных систем Linux. В некоторых случаях, например, если вы используете только разделы XFS, использование *dump* и *restore* может быть гораздо полезнее, чем использование простого tar или сріо.

## Расширенное резервное копирование при помощи rsync

*rsync* - утилита, которая обеспечивает быструю расширенную передачу файлов. Часто для автоматизированного удаленного резервного копирования *rsync* является наилучшим инструментом для работы. Хорошая особенность *rsync* по сравнению с другими инструментами - то, что *rsync* может произвольно предписать двухстороннюю синхронизацию. Таким образом, вместо того, чтобы просто копировать наиболее новые или измененные файлы, *rsync* может автоматически удалить из отдаленной резервной копии файлы, удаленные на локальной машине.

Чтобы понять смысл аргументов, полезно посмотреть на этот не очень сложный скрипт (расположенный на Web-страницах *rsync*):

```
#!/bin/sh
# This script does personal backups to a rsync backup server. You will
# end up with a 7 day rotating incremental backup. The incrementals will
```

```

# go into subdirs named after the day of the week, and the current
# full backup goes into a directory called "current"
# tridge@linuxcare.com
# directory to backup
BDIR=/home/$USER
# excludes file - this contains a wildcard pats of files to exclude
EXCLUDES=$HOME/cron/excludes
# the name of the backup machine
BSERVER=owl
# your password on the backup server
export RSYNC_PASSWORD=XXXXXX
BACKUPDIR=`date +%A`
OPTS="--force --ignore-errors --delete-excluded --exclude-from=$EXCLUDES
      --delete --backup --backup-dir=/${BACKUPDIR} -a"
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin
# the following line clears the last weeks incremental directory
[ -d $HOME/emptydir ] || mkdir $HOME/emptydir
rsync --delete -a $HOME/emptydir/ $BSERVER::$USER/${BACKUPDIR}/
rmdir $HOME/emptydir
# now the actual transfer
rsync $OPTS ${BDIR} $BSERVER::$USER/current

```

## Ресурсы

### Научиться

- [Оригинал данной главы](#) на developerWorks.
- В [LPIC Program](#) (Программе LPIC) вы найдете список заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.
- " [Using the GNU text utilities](#) " ("Использование текстовых утилит GNU") (developerWorks, March 2004) рассказывает, как вы можете применять коллекцию текстовых утилит GNU для обработки файлов системных журналов, документации, баз данных и других текстовых данных или контента.
- В " [Understanding Linux configuration files](#) " (developerWorks, декабрь 2001) описаны приемы настройки в системах Linux конфигурационных файлов, которые используются для управления правами пользователей, системными приложениями, демонами, службами и решения других задач администрирования в многопользовательском, многозадачном окружении.
- " [Windows-to-Linux roadmap: Part 8. Backup and recovery](#) " (developerWorks, November 2003) краткое руководство по резервному копированию и восстановлению Linux систем.
- В [Installation Guide and Reference: Software Product Packaging Concepts](#) обсуждаются концепции упаковки программного обеспечения.
- Найдите другие ресурсы для разработчиков Linux на [developerWorks Linux zone](#).

### Получить продукты и технологии

- Вы можете найти [sample Samba scripts](#) (примеры Samba скриптов) на интернет странице rsync.
- [Закажите SEK для Linux](#), набор из двух DVD, содержащих trial-версии последнего программного обеспечения IBM для Linux от DB2®, Lotus®, Rational®, Tivoli® и WebSphere®.

- Постройте ваш следующий проект разработки для Linux с использованием [IBM trial software](#), загрузив его непосредственно с developerWorks.

# Экзамен LPI 201: Настройка работ и автоматическое выполнение заданий

*Администрирование Linux, средний уровень (LPIC-2) тема 213*

Дэвид Мерц, автор, Gnosis Software, Inc.

Бред Хантинг, Mathematician, Университет Колорадо

**Описание:** В этом учебном пособии Дэвид Мерц и Бред Хантинг начинают готовить вас к сдаче Linux™ Professional Institute® Администрирование, средний уровень (LPIC-2) 201. В этом седьмом из восьми учебных пособий вы изучите основы написания командных скриптов и автоматизации выполнения заданий, таких как создание отчетов и отслеживание состояний, очистка системы и общая поддержка.

[Больше статей из этой серии](#)

**Дата:** 01.09.2005

**Уровень сложности:** средний

## Прежде чем начать

Узнайте, чему может научить вас это учебное пособие и как извлечь из него максимум.

## Об этой серии учебных пособий

Linux Professional Institute (LPI) осуществляет сертификацию системных администраторов Linux на уровень начинающих и средний уровень. Для достижения любого из этих уровней вам нужно будет сдать два LPI экзамена.

Каждый экзамен охватывает несколько тем, причем каждая из этих тем имеет свой рейтинг. Рейтинг показывает относительную важность каждого раздела. Грубо говоря, чем выше рейтинг темы, тем больше она может содержать вопросов. Темы экзамена LPI 201 и их рейтинги:

### Тема 201

Ядро Linux (рейтинг 5).

### Тема 202

Запуск системы (рейтинг 5).

### Тема 203

Файловая система (рейтинг 10).

### Тема 204

Оборудование (рейтинг 8).

### Тема 209

Совместное использование файлов и служб (рейтинг 8).

### Тема 211

Поддержка системы (рейтинг 4).

### Тема 213

Настройка работ и автоматическое выполнение заданий (рейтинг 3). В фокусе этого учебного пособия.

### Тема 214

Устранение неполадок (рейтинг 6).

Linux Professional Institute не одобряет использование при подготовке к экзаменам любых учебных материалов или технологий, разработанных третьими лицами. За разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## **Об этом учебном пособии**

Добро пожаловать в учебное пособие "Настройка работ и автоматическое выполнение заданий", седьмое из восьми пособий, разработанных для подготовки к экзамену LPI 102. Из этого пособия вы узнаете несколько основных приемов создания скриптов и автоматизации системных событий, таких как генерация отчетов и сводок, очистка системы и ее общая поддержка.

Это учебное пособие организовано в соответствии с рабочей программой LPI по этой теме:

### **2.213.1 Автоматизация заданий при помощи скриптов (рейтинг 3)**

Вы сможете писать простейшие Perl скрипты, которые будут использовать соответствующие модули, использовать Perl taint mode для защиты данных и инсталлировать Perl модули из репозитория Comprehensive Perl Archive Network (CPAN). Сюда также входит использование sed и awk скриптов, использование скриптов для контроля хода выполнения процессов и генерация извещений через e-mail или pager когда процесс умирает. Кроме того, вы научитесь писать и отслеживать автоматическое выполнение скриптов для анализа системных журналов на предмет исключительных ситуаций и осуществлять автоматическую рассылку их системным администраторам по e-mail, синхронизацию файлов на разных машинах при помощи rsync, мониторинг файлов на предмет их изменения и генерацию извещений по электронной почте и создание скриптов, которые оповещают администраторов, когда определенные пользователи входят в систему или выходят из нее.

Одна из задач системного администрирования заключается в автоматизации заданий, которые должны происходить периодически, и эффективном обслуживании тех событий, которые возникают в непредсказуемые моменты. Для автоматизации таких задач основным средством для вас является [cron](#) и [at](#). Задания, которые выполняются регулярно или запускаются вручную, могут быть запрограммированы в виде скриптов на различных языках программирования, включая bash, awk, Perl или Python. Программы из набора GNU text utilities частенько используются как часть различных заданий; чаще всего они применяются в bash скриптах, в то время как в более изощренных языках, таких как awk, Perl и Python предоставляемые text utilities средства просто встроены в язык.

## **Необходимые условия**

Чтобы извлечь максимум из этого учебного пособия, вы должны иметь базовые знания о Linux и рабочую версию системы Linux, где вы сможете упражняться в выполнении команд, приведенных в этом пособии.

# **Экзамен LPI 201: Настройка работ и автоматическое выполнение заданий**

*Администрирование Linux, средний уровень (LPIC-2) тема 213*

[Дэвид Мерц](#), автор, Gnosis Software, Inc.

[Бред Ханting](#), Mathematician, Университет Колорадо

**Описание:** В этом учебном пособии Дэвид Мерц и Бред Хантиг начинают готовить вас к сдаче Linux™ Professional Institute® Администрирование, средний уровень (LPIC-2) 201. В этом седьмом из восьми учебных пособий вы изучите основы написания командных скриптов и автоматизации выполнения заданий, таких как создание отчетов и отслеживание состояний, очистка системы и общая поддержка.

[Больше статей из этой серии](#)

**Дата:** 01.09.2005

**Уровень сложности:** средний

## Автоматизация периодически запускаемых заданий

### Конфигурирование cron

Демон **cron** используется для периодического запуска команд. Вы можете использовать **cron** для широкого круга задач по поддержке и администрированию. Если это событие или задание возникает с определенной реглярностью, его следует обслуживать при помощи **cron**. **Cron** пробуждается каждую минуту и проверяет, не нужно ли чего-нибудь сделать, но он не может запускать эти задания чаще чем раз в минуту. (Если вам требуется такая функциональность, возможно вам необходим демон, а не "cron job.") **Cron** журналирует свои действия через механизм syslog.

**Cron** ищет свои конфигурационные файлы, в которых устанавливаются переменные окружения и команды, которые следует выполнять, в разных местах. Первый из них -- это /etc/crontab, содержащий системные задания. Каталог /etc/cron.d/ может содержать различные конфигурационные файлы, дополняющие /etc/crontab. Отдельные пакеты могут добавлять файлы (с именами, соответствующими имени пакета) в /etc/cron.d/, но системному администратору следует использовать /etc/crontab.

ПОЛЬЗОВАТЕЛЬСКИЕ конфигурации для **cron** хранятся в /var/spool/cron/crontabs/\$USER. Однако, они должны быть созданы при помощи программы **crontab**. При помощи **crontab** пользователи могут задавать свои собственные периодически запускаемые задания.

### Ежедневный, еженедельный и ежемесячный запуск заданий

Задания, которые должны выполняться ежедневно, еженедельно или ежемесячно -- а это наиболее распространенная ситуация -- описываются согласно специальным соглашениям. Каталоги /etc/cron.daily/, /etc/cron.weekly/ и /etc/cron.monthly/ созданы для хранения наборов соответствующих скриптов. Добавление или удаление скриптов в этих каталогах -- это простейший путь для управления системными заданиями. Например, систему, которую обслуживаю я, осуществляет ежедневную ротацию файлов журналов при помощи скрипта:

### Листинг 1. Пример скриptового файла, запускаемого ежедневно

```
$ cat /etc/cron.daily/logrotate
#!/bin/sh
test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

### Cron и anacron

Вы можете использовать **anacron** для выполнения команд с частотой в определенное количество дней. В отличие от **cron**, **anacron** проверяет каждое задание, которое должно было быть исполнено в течение последних *n* дней (где *n* -- это период, определенный для данного задания, в отличие от проверки текущего времени для определения момента запуска). Если оно не запускалось, **anacron** производит запуск с задержкой, указанной в минутах в параметре *delay*. Таким образом, на машинах, которые не включены постоянно, периодически запускаемые задачи выполняются единожды, когда машина работает (конечно, точное время запуска может варьироваться, но задание не будет забыто).

**Anacron** читает список заданий из конфигурационного файла /etc/anacrontab. Каждая запись

включает в себя период в днях, задержку в минутах, уникальный идентификатор задания и команду оболочки. Например, на одной из Linux систем, которую я поддерживаю, [anacron](#) используется для ежедневного, еженедельного и ежемесячного запуска заданий, даже если машина была выключена в тот момент, когда оно должно было быть запущено:

## Листинг 2. Пример конфигурационного файла anacron

```
$ cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
# These replace cron's entries
1      5  cron.daily    nice run-parts --report /etc/cron.daily
7      10 cron.weekly   nice run-parts --report /etc/cron.weekly
@monthly 15 cron.monthly nice run-parts --report /etc/cron.monthly
```

## Содержимое crontab

Формат /etc/crontab (или содержимое файлов /etc/cron.d/) несколько отличается от пользовательских файлов crontab. По сути дела, они содержат одно дополнительное поле, указывающее пользователя, под которым должна запускаться данная команда. Для пользовательских crontab файлов это не нужно, так как они уже содержат имя пользователя в своем названии (/var/spool/cron/crontabs/\$USER).

Каждая строчка /etc/crontab или устанавливает переменную окружения, или описывает задание. Комментарии и пустые строки игнорируются. Для заданий cron первые пять полей задают время запуска (где каждое поле может задаваться списком или диапазоном). Поля обозначают минуты, часы, дни месяца, месяцы, дни недели (разделяются пробелами или табуляциями). Asterisk (\*) в любой позиции обозначает *любой*. Например, для запуска задания в полночь по вторникам и четвергам с августа по октябрь, следует сделать так:

```
# line in /etc/crontab
0 0 * 7-9 2,5 root /usr/local/bin/the-task -opt1 -opt2
```

## Использование специальных scheduling переменных

Наиболее распространенными scheduling pattern'ам назначены сокращенные имена, которые вы можете использовать в первых пяти полях:

### @reboot

Запускать один раз, при старте.

### @yearly

Запускать один раз в год, "0 0 1 1 \*".

### @annually

Тоже самое, что и @yearly.

### @monthly

Запускать один раз в месяц, "0 0 1 \* \*".

### @weekly

Запускать один раз в неделю, "0 0 \* \* 0".

### @daily

Запускать один раз в день, "0 0 \* \* \*".

### @midnight

Тоже самое, что и @daily.

### @hourly

Запускать раз в час, "0 \* \* \* \*".

Например, конфигурационный файл может содержать:

```
@hourly root /usr/local/bin/hourly-task
0,29 * * * * root /usr/local/bin/twice-hourly-task
```

### Использование crontab

Для установки пользовательских заданий используйте команду [crontab](#) (в отличие от файла /etc/crontab). Конкретнее, [crontab -e](#) запускает текстовый редактор для правки файла. Вы можете вывести ваш текущий список заданий при помощи [crontab -l](#) и удалить задание при помощи [crontab -r](#). Или же вы можете задать [crontab -u user](#) для управления заданиями указанного пользователя user, но по умолчанию это будете вы (ограничения на права доступа играют свою роль).

Если в системе присутствует файл /etc/cron.allow, то он должен содержать имена всех пользователей, которым разрешено управление такими заданиями. С другой стороны, если файла /etc/cron.allow нет, то пользователь не должен быть помещен в файл /etc/cron.deny, если ему должно быть разрешено управление заданиями. Если ни одного из этих файлов нет, кто угодно может использовать [crontab](#).

**Описание:** В этом учебном пособии Дэвид Мерц и Бrad Хантинг начинают готовить вас к сдаче Linux™ Professional Institute® Администрирование, средний уровень (LPIC-2) 201. В этом седьмом из восьми учебных пособий вы изучите основы написания командных скриптов и автоматизации выполнения заданий, таких как создание отчетов и отслеживание состояний, очистка системы и общая поддержка.

## Автоматизация одноразовых заданий

### Использование команды at

Если вам надо запустить задание в какое-то время, вы можете использовать команду [at](#), которая берет команды со стандартного ввода STDIN или из файла (через опцию [-f](#)) и принимает описание времени запуска в различных, достаточно гибких, форматах.

Семейство команд, связанных с [at](#) включает в себя: [atq](#) -- выводит список отложенных заданий; [atrm](#) -- удаляет задание из очереди; и [batch](#) -- работает подобно [at](#), за исключением того, что она откладывает выполнение задания до тех пор, пока загрузка системы не будет низкой.

### Права

Подобно /etc/cron.allow и /etc/cron.deny, команда [at](#) имеет файлы /etc/at.allow и /etc/at.deny для управления правами. Файл /etc/at.allow, если он присутствует, должен содержать всех пользователей, которым разрешено управлять запуском заданий. С другой стороны, если файла /etc/at.allow нет, пользователь должен отсутствовать в /etc/at.deny, если запуск заданий ему разрешен. Если ни одного из этих файлов не существует, все могут использовать [at](#).

### Указание времени

Обратитесь к справочному руководству [man at](#) для получения полной информации о вашей версии [at](#). Вы можете указать конкретное время в часах и минутах виде **HH:MM** для события, которое должно произойти, когда это время настанет. (Если это время уже прошло, это означает, что событие наступит завтра). Если вы используете 12-часовую систему измерения времени, вы можете также добавлять а.м. или р.м. Вы можете указывать дату в виде **MMDDYY**, **MM/DD/YY**, **DD.MM.YY** или **month-name-day**. Вы можете также прибавлять время к текущему следующим образом: [now + N units](#), где *N* это число, а *units* это minutes, hours, days или weeks. Слова *today* и *tomorrow* имеют очевидное значение ("сегодня" и "завтра"), так же как *midnight* ( полночь ) и *noon* ( полдень ), *teatime* это 4 р.м. Несколько

примеров:

```
% at -f ./foo.sh 10am Jul 31 % echo 'bar -opt' | at 1:30 tomorrow
```

Точное определение временных спецификации см. в /usr/share/doc/at/timespec.

**Описание:** В этом учебном пособии Дэвид Мерц и Бrad Хантинг начинают готовить вас к сдаче Linux™ Professional Institute® Администрирование, средний уровень (LPIC-2) 201. В этом седьмом из восьми учебных пособий вы изучите основы написания командных скриптов и автоматизации выполнения заданий, таких как создание отчетов и отслеживание состояний, очистка системы и общая поддержка.

## Заметки о скриптах

### Внешние ресурсы

О awk, Perl, bash и Python существует большое количество отличных книг. Соавтор этого руководства (естественно) рекомендует собственное издание, [\*Text Processing in Python\*](#), в качестве хорошей стартовой точки по написанию скриптов на Python.

Большинство скриптов, которые пишутся с ориентацией на системное администрирование, ориентируются на манипуляцию текстовыми данными, такими как извлечение величин из системных журналов и конфигурационных файлов и генерация отчетов и сводок. Сюда также относятся процедуры очистки системы и рассылка извещений о результатах исполнения заданий.

В Linux большинство обычных скриптов для системных администраторов пишутся на bash. Сам по себе bash имеет относительно мало встроенных возможностей, но зато может с легкостью использовать внешние программы (включая такие стандартные утилиты, как ls, find, rm и cd) и программы для обработки текстов (подобные тем, которые можно найти в GNU text utilities).

### Заметки о bash

Одной из наиболее полезных установок, которые можно включать в bash скрипты, применяемые для обработки заданий, является опция `set -x`, выводящая исполняемые команды на стандартный вывод ошибок STDERR. Это очень полезно при отладке скриптов, когда они не выполняют того, что от них ожидается. Другая полезная для тестирования опция `--set -n`, которая помогает отследить синтаксические проблемы в скрипте без его реального исполнения. Разумеется, у вас не возникнет потребности использовать `-n` при запуске программы через `cron` или `at`, но для запуска и проверки их работоспособности это может быть полезно.

### Листинг 3. Простейшее задание для cron, запускающее bash скрипт

```
#!/bin/bash
exec 2>/tmp/my_stderr
set -x
# functional commands here
```

В этом случае вывод на STDERR перенаправляется в файл и выводит запускаемые команды на STDERR. Последующее изучение этого файла может оказаться полезным.

Системное руководство man для bash конечно хорошее, но уж слишком большое. Наиболее интересными являются опции, доступные через встроенную команду `set`.

Обычной задачей программирования для системного администрирования является процесс сбора файлов, как правило файлов, найденных по тем или иным критериям командой `find`. Однако, при наличии в именах файлов пробелов или символов перевода строки могут возникать проблемы. Большое количество процессов, использующих перебор в цикле и обработке имен файлов, могут исполняться некорректно при наличии пробельных символов в именах. Например, следующие две команды различаются:

```
% rm foo bar baz bam  
% rm 'foo bar' 'baz bam'
```

Первая команда удаляет четыре файла (они для этого должны существовать); вторая же удаляет только два файла, каждый из которых включает пробелы в имени файла. Имена файлов с пробелами достаточно обычное явление для мультимедийного контента.

К счастью, GNU версия команды `find` имеет опцию `-print0`, ограничивающую каждый результат NULL'ем; а команда `xargs` имеет соответствующую опцию `-0` для обработки аргументов, разделенных NULL'ем. Совместив эти две команды, вы можете удалить затерявшиеся файлы, содержащие в именах пробельные символы:

#### Листинг 4. Очистка имен файлов от пробелов

```
#!/bin/bash  
# Cleanup some old files  
set -x  
find /home/dqm \( -name '*.core' -o -name '#*' \) -print0 \  
| xargs -0 rm -f
```

#### Perl taint mode

Perl имеет удобную опцию `-T` для переключения в taint mode. В этом режиме Perl предпринимает некоторые предосторожности, связанные с повышением защищенности, одной из которых является накладывание ограничений на команды, связанными с внешним вводом. Если вы используете запуск через `sudo`, taint mode может быть включен по умолчанию, но более надежным является запуск ваших административных скриптов через:

```
#!/usr/local/bin/perl -T
```

Как только вы сделаете это, все аргументы командной строки, переменные окружения, информация о locale (см. `perllocale`), результаты работы системных вызовов (`readdir()`, `readlink()`, переменные `shmread()`, сообщения порождаемые `msgrecv()`, поля `password`, `gcos` и `shell`, возвращаемые `getpwxxx()`) и ввод из всех файлов, помеченные как "tainted". Такие данные не могут использоваться прямо или косвенно любыми командами, ни запускаемыми через вложенный вызов командного интерпретатора, ни в любой команде, модифицирующей файлы, каталоги или процессы, за некоторыми исключениями.

Существует возможность провести отмену taint-режима (untaint) для отдельных внешних переменных через исчерпывающую проверку по определенным образцам:

#### Листинг 5. Untainting внешних переменных окружения

```
if ($data =~ /^([-@\w.]+)$/) {  
    $data = $1;                      # $data now untainted  
} else {  
    die "Bad data in $data";        # log this somewhere  
}
```

## Пакеты Perl CPAN

Одна из полезнейших вещей, доступных в Perl, -- это наличие стандартного механизма для установки дополнительных пакетов; он называется Comprehensive Perl Archive Network (CPAN). RubyGems обладает аналогичной функциональностью. Python, к несчастью, пока не обладает механизмом автоматической установки, но зато имеет достаточно широкий набор в комплекте поставки. Более простые языки, подобные bash и awk, не имеют возможностей для установки расширений, подобных описанным выше.

Страницы руководства man по команде [cpan](#) -- это хорошая отправная точка, особенно если перед вами стоит задача, которая, как вы думаете, во многом уже решена другими.

Кандидатами таких модулей могут являться те, что вы можете найти на [CPAN](#).

[cpan](#) может работать в интерактивном режиме и режиме командной строки. Единожды сконфигурированный (запустите интерактивный режим и в ходе первого такого запуска у вас будут запрошены опции конфигурации), [cpan](#) отслеживает взаимозависимости и загружает их автоматически. Например, положим перед вами стоит задача, которая требует обработки конфигурационных файлов в формате YAML (yaml Ain't Markup Language). Установка поддержки для YAML так же проста, как:

```
% cpan -i YAML # maybe with 'sudo' first
```

После установки ваши скрипты могут включать `use YAML;` в начале файла. Это позволяет вам использовать все возможности, предоставленные разработчиком пакета.

## Ресурсы

### Научиться

- [Оригинал этого учебного пособия](#) на developerWorks.
- В [Программе LPIC](#) вы найдете список заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.
- В "[Understanding Linux configuration files](#)" (developerWorks, декабрь 2001) описаны приемы настройки в системах Linux конфигурационных файлов, которые используются для управления правами пользователей, системными приложениями, демонами, службами и решения других задач администрирования в многопользовательском, многозадачном окружении.
- Учебное пособие от developerWorks "[Using the GNU text utilities](#)" (developerWorks, март 2004) знакомит вас с утилитами для создания скриптов.
- Эти [статьи о создании скриптов](#) от developerWorks предоставляют множество средств создания скриптов, позволяющих автоматизировать задачи в Linux.
- [Text Processing in Python](#), написанный Дэвидом Мерцем, одним из соавторов этого учебного пособия, -- отличный источник скриптов на Python.
- Найдите другие ресурсы для разработчиков Linux на [developerWorks Linux zone](#).

### Получить продукты и технологии

- [CPAN Module Archive](#) -- источник модулей Perl.
- [Закажите SEK для Linux](#), набор из двух DVD, содержащих trial-версии последнего программного обеспечения IBM для Linux от DB2®, Lotus®, Rational®, Tivoli® и WebSphere®.
- Постройте ваш следующий проект разработки для Linux с использованием [IBM trial](#)

[software](#), загрузив его непосредственно с developerWorks.

# Подготовка к экзамену LPI 201: Устранение неполадок

Средний уровень администрирования (LPIC-2) тема 214

[Brad Huntting](#), Mathematician, University of Colorado

[Дэвид \(David\) Мертц \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** В этом учебном пособии Брэд Хантинг и Дэвид Мертц продолжают готовить вас к сдаче Профессионального Института Linux® экзамена 201 Администрирования Среднего Уровня (LPIC-2). В этом, последнем из восьми, учебном пособии сделан упор на то, как поступать, когда что-нибудь идет не так, как надо. Оно построено на материале, уже освещенном более подробно в ранних пособиях.

[Больше статей из этой серии](#)

**Дата:** 02.09.2005

**Уровень сложности:** средний

## Создание дисков восстановления

### Перед восстановлением

Когда установленная Linux становится настолько запущенной, что она просто не может нормально загрузиться, хорошим первым шагом в исправлении проблемы будет попытка загрузиться в однопользовательском режиме. Однопользовательский режим (уровень запуска 1) позволяет вам исправлять множество проблем, не беспокоясь о правах доступа или файлах, заблокированных другими пользователями или процессами. Более того, однопользовательский режим запускается с минимальным набором служб, демонов и задач, которые могут вас запутывать или же служить причиной основной проблемы.

В некоторых случаях, однако, файлы, необходимые для однопользовательского режима, такие как /etc/passwd, /etc/fstab, /etc/inittab, /sbin/init, /dev/console и прочие, могут быть поврежденными и система не загрузится даже в однопользовательском режиме. В таких случаях для того, чтобы вновь поднять систему на ноги, используется диск восстановления (загрузочный диск, содержащий основы, необходимые для починки испорченного раздела, иногда называемый *repair disk*).

### Компакт-диски восстановления

На сегодняшний день дистрибутивы, собранные в виде live-CD (или DVD), такие как Knoppix, являются бриллиантами в короне дисков восстановления, предлагая широкий набор драйверов, средств отладки, и даже Веб-браузер. Даже если у вас есть особые запросы, например, нестандартные драйверы (модули ядра) или ПО для восстановления, вам лучше скачать один из Linux LiveCD, например, Knoppix, и скорректировать содержимое ISO-образа под свои цели перед тем как записать его на диск. Обратитесь к учебному пособию на тему 203 за информацией о монтировании ISO-образа.

### Гибкие диски восстановления

На очень старых системах, не имеющих загрузочных CD-ROMов, для восстановления должны использоваться гибкие диски, содержащие только самое необходимое. Всякие там изящности типа *emacs* и *vim* обычно слишком громоздки, чтобы поместиться на одну дискету, так что опытный системный администратор должен быть готовым к использованию строчного редактора наподобие *ed* для исправления поврежденных конфигурационных файлов.

Для создания загрузочной дискеты, следует прочесть "Linux-Bootdisk HOWTO" от Linux Documentation Project. Во многих случаях, однако, дистрибутив Linux предлагает создать дискету восстановления во время установки или позже, используя встроенные утилиты. Убедитесь, что вы создали и храните диски восстановления *перед* тем, как они вам действительно понадобятся!

**Описание:** В этом учебном пособии Брэд Хантинг и Девид Мертц продолжают готовить вас к сдаче Профессионального Института Linux® экзамена 201 Администрирования Среднего Уровня (LPIC-2). В этом, последнем из восьми, учебном пособии сделан упор на то, как поступать, когда что-нибудь идет не так, как надо. Оно построено на материале, уже освещенном более подробно в ранних пособиях.

## Распознавание стадий загрузки

### О загрузке

Учебное пособие по теме 202 содержит куда более широкую информацию об очередности загрузки Linux. В том пособии мы только кратко рассмотрим эти стадии.

Первая стадия загрузки немного изменилась с тех пор, как появились первые IBM®-совместимые PC с жесткими дисками. BIOS считывает первый сектор загрузочного диска в память и запускает его. Эта 512-байтная Master Boot Record (MBR), где также располагается метка fdisk, загружает "загрузчик ОС" (либо GRUB, либо LILO) из "активного" раздела.

### Загрузка ядра

По мере того, как BIOS в системе x86 (в системах с другими архитектурами имеются небольшие отличия) запускает MBR, происходит несколько стадий, приводящих к загрузке ядра Linux. Если загрузка не удается, то первый шаг в определении того, что нужно исправить, должен состоять в поиске стадии, где произошел сбой.

- Загрузка Boot Loader (LILO/Grub).
- Запускается Boot loader и передает управление ядру.
- Ядро: Загружается база ядра, а также основные модули ядра.
- Инициализация и установка устройств:
  - Запускается ядро.
  - Инициализируется инфраструктура ядра (VM, планировщик, и прочее).
  - Зондирование и присоединение драйверов устройств.
  - Монтируется корневая файловая система.

### Загрузка пользовательского пространства

Полагая, что база ядра, модули ядра и корневая файловая система стартовали успешно (или, по крайней мере, достаточно успешно, не полностью зависнув), начинается процесс инициализации системы:

- Инициализация и установка демонов
  - /sbin/init запускается как процесс 1
  - /sbin/init считывает /etc/inittab
  - /etc/init.d/rc запускает сценарии /etc/rc<n>.d/S\*
  - Диски проверяются fsck
  - Настраиваются сетевые интерфейсы
  - Запускаются демоны
  - getty(s) запускается на клавиатуре и последовательных портах

**Описание:** В этом учебном пособии Брэд Хантинг и Девид Мертц продолжают готовить вас к сдаче Профессионального Института Linux® экзамена 201 Администрирования Среднего Уровня (LPIC-2). В этом, последнем из восьми, учебном пособии сделан упор на то, как поступать, когда что-нибудь идет не так, как надо. Оно построено на материале, уже освещенном более подробно в ранних пособиях.

## Устранение неполадок LILO и GRUB

### Поиск загрузочного сектора

При запуске GRUB на экране высвечивается "GRUB loading, please wait...", а затем GRUB переходит в свое загрузочное меню. LILO обычно выводит приглашение **LILO:** (но некоторые версии выводят меню сразу же). Если экран LILO или GRUB недостижим, то, вероятно, пришло время для диска восстановления. Вообще говоря, вы узнаете об этой проблеме вместе с каким-нибудь сообщением BIOS о том, что не найден загрузочный сектор или даже жесткий диск.

Если ваша система не может отыскать пригодный загрузочный сектор, то, возможно, что ваша LILO или GRUB стала поврежденной. Иногда некоторые операционные системы (или не в меру старательные "программы защиты от вирусов" под Windows®) перезаписывают загрузочные сектора, и попутно создают другие проблемы. Диск восстановления позволит вам переустановить LILO или GRUB. Как правило, желательно применить **chroot** к старой файловой системе, чтобы использовать конфигурационные файлы прежних LILO или GRUB.

В случае **hard-disk-not-found**, вероятно, имеет место сбой оборудования (и вам остается надеяться, что сделана резервная копия); часто это имеет место и в случае **no-boot-sector**. Обратитесь учебному пособию на тему 202 за более подробной информацией о LILO и GRUB.

### Настройка LILO и GRUB

LILO настраивается при помощи файла /etc/lilo.conf или другим конфигурационным файлом, задаваемом в командной строке. Чтобы переустановить LILO, сначала убедитесь, что ваш файл lilo.conf действительно отвечает вашей текущей системной конфигурации (в особенности информации о правильных разделах и номере диска; они могут изменяться например при добавлении жесткого диска и/или разделов). После того, как такая проверка сделана, просто запустите /sbin/lilo под root (или в однопользовательском режиме).

Настройка и переустановка GRUB, в сущности, представляет собой тот же процесс, что и для LILO. Файлом конфигурации GRUB'a является /boot/grub/menu.lst, но он считывается при каждой загрузке системы. Аналогично, оболочка GRUB располагается в /boot/grub/, но с помощью опции **grroot=** в /boot/grub/menu.lst можно выбирать, на каком жестком диске располагается основная GRUB MBR. Для установки GRUB в MBR, запустите команду наподобие **grub-install /dev/hda**, которая проверит подключенный на данный момент /boot/ для своей конфигурации.

**Описание:** В этом учебном пособии Брэд Хантинг и Девид Мертц продолжают готовить вас к сдаче Профессионального Института Linux® экзамена 201 Администрирования Среднего Уровня (LPIC-2). В этом, последнем из восьми, учебном пособии сделан упор на то, как поступать, когда что-нибудь идет не так, как надо. Оно построено на материале, уже освещенном более подробно в ранних пособиях.

## Общее устранение неполадок

### Структура файловой системы Linux

Дистрибутивы Linux имеют небольшие различия в том, где они содержат файлы. Linux Filesystem Hierarchy Standard делает успехи в дальнейшей стандартизации этого. Уже являются стандартными несколько каталогов, и туда, в частности, важно смотреть при проблемах с загрузкой и выполнением:

- `/proc/` -- это виртуальная файловая система с информацией о процессах и о состоянии системы. В сущности, все внутренности работающей системы находятся здесь. За большей информацией смотрите тему 201.
- `/var/log/` -- то место, где находятся файлы журналов. Если что-то идет не так, здесь можно найти полезную информацию, содержащуюся в некоторых файлах журнала.
- `/` -- вообще говоря, корень файловой системы под Linux, попросту содержащий другие каталоги, вложенные в него. На некоторых системах загрузочные файлы, такие как `vmlinuz` и `initrd.img` могут находиться прямо здесь, а не в `/boot/`.
- `/boot/` хранит файлы, непосредственно используемые в процессе загрузки ядра.
- `/lib/modules/` -- это то, где располагаются модули ядра, вложенные в подкаталоги этого каталога. Имена подкаталогов совпадают с номером текущей версии ядра (если загружаются несколько версий ядра, то должны присутствовать несколько каталогов). Например:

```
% ls /lib/modules/2.6.10-5-386/kernel/
    arch crypto drivers fs lib net security sound
```

### Обнаружение сообщений загрузки

Во время загрузки системы сообщения могут прокрутиться очень быстро, и вы можете не успеть опознать проблему или неожиданное действие при инициализации. Некоторая интересующая вас информация, возможно, журналируется `syslog`, но основные сообщения ядра и модулей ядра можно исследовать с помощью утилиты `dmesg`.

### Программы, распознающие Hardware/system

Учебное пособие на тему 203 (аппаратное обеспечение) содержит больше информации о распознавании оборудования. Вообще говоря, следует помнить про следующие системные утилиты:

- `lspci`: Выдает все устройства PCI
- `lsmod`: Выдает список загруженных модулей ядра.
- `lsusb`: Выдает устройства USB.
- `lspnp`: Выдает устройства Plug-and-Play.
- `lshw`: Выдает список оборудования.

Не совсем для аппаратного обеспечения, но все-таки полезны:

- `lsof`: Выдает список открытых файлов.
- `insmod`: Загружает модули ядра.
- `rmmod`: Выгружает модули ядра.
- `modprobe`: Обертка для `insmod/rmmod/lsmod` высокого уровня.
- `uname`: Выдает системную информацию (версия ядра и тп.).
- `strace`: Отслеживает системные вызовы.

Если вы пришли в отчаяние, пытаясь использовать программы, или же библиотеки, или приложения, вас может спасти утилита `strings` (но имейте ввиду, что здесь придется поработать). Решающая информация, такая как hard-coded пути иногда зарыта в исполняемых файлах и их можно найти, и (но с большой долей проб и ошибок), посредством

поиска строк в бинарниках.

**Описание:** В этом учебном пособии Брэд Хантинг и Дэвид Мертц продолжают готовить вас к сдаче Профессионального Института Linux® экзамена 201 Администрирования Среднего Уровня (LPIC-2). В этом, последнем из восьми, учебном пособии сделан упор на то, как поступать, когда что-нибудь идет не так, как надо. Оно построено на материале, уже освещенном более подробно в ранних пособиях.

## Устранение неполадок в системных ресурсах и в конфигурации окружения

### The initialization

В учебнике на тему 202 подробнее рассказывалось, что запуск Linux после загрузки ядра управляет процессом init. Основная конфигурация init лежит в /etc/inittab. Этот файл /etc/inittab содержит подробности того, какие шаги следует произвести на каждом уровне запуска. Но, возможно, наиболее решающим является тот факт, что там задается уровень запуска для последующих действий. Если система имеет проблемы во время загрузки, установка другого уровня запуска может помочь. Ключевой строкой будет что-нибудь вроде этого:

```
# The default runlevel. (in /etc/inittab)
id:2:initdefault:
```

### Сценарии инициализации

Сценарии инициализации ("rc-скрипты") запускаются во время загрузки, завершения, и всегда, когда система меняет уровни запуска, и они ответственны за запуск и остановку большинства системных демонов. В большей части (читай *в современных*) дистрибутивах Linux, они находятся в каталоге /etc/init.d/ и ссылаются на каталоги /etc/rc<N>.d/ (при N=уровень запуска), где они имеют имена "S\*" для сценариев запуска и "K\*" для сценариев завершения. Система никогда не запускает сценариев из каталога /etc/init.d/, а ищет их в /etc/rc<N>.d/[SK]\*.

### Оболочка системы

Иногда, но бывает так, что системный администратор желает изменить общий для всей системы сценарий запуска оболочки /etc/profile. Эта смена влияет на всякую интерактивную оболочку (за исключением пользователей /bin/tcsh и других не-/bin/sh-совместимых оболочек). Повреждение этого файла может с легкостью привести к ситуации, когда никто не сможет зайти в систему, и для исправления потребуется диск восстановления. Обычный способ изменения поведения оболочки на индивидуальной основе состоит в изменении /home/\$USER/.bash\_profile и /home/\$USER/.bashrc.

### Настройка параметров ядра

Система *sysctl* (см. man-страницу для sysctl) была взята из BSD UNIX® и используется для настройки некоторых системных ресурсов. Выполните *sysctl -a*, чтобы увидеть, какие переменные могут управляться sysctl, и какие значения они принимают. Утилита sysctl наиболее полезна при настройке как параметров сети, так и некоторых параметров ядра. Файл /etc/sysctl.conf используется, чтобы задать параметры sysctl во время загрузки.

### Динамические библиотеки

В большинстве систем, динамические библиотеки постоянно добавляются, обновляются, заменяются и удаляются. Поскольку в системе почти каждой программе требуется найти и загрузить динамическую библиотеку, имена, номера версий, и местоположение большинства динамических библиотек кэшируются программой *ldconfig*. Обычно кэшируются динамические библиотеки из системных каталогов /lib/ и /usr/lib/. Чтобы добавить больше

каталогов к глобальному списку каталогов для поиска по умолчанию, следует добавить имена этих каталогов (например /usr/X11R6/lib) в файл /etc/ld.so.conf и запустить ldconfig под root.

## Системное журналирование

Тема 211 подробно рассматривает syslog. Основной файл, про который следует вспомнить, если у вас появились проблемы (или если вы хотите убедиться, что вы сможете проанализировать их позже) -- это /etc/syslog.conf. Изменяя содержимое этой конфигурации, у вас есть детальнейший контроль над тем, какие события журналируются, и куда пишутся файлы журнала, возможно даже включая почту и удаленные машины. Если появляются проблемы, убедитесь, что подсистемы, где, по вашему мнению, они возникают, журналируют информацию в таком стиле, чтобы вы смогли быстро проверить.

## Периодические события

Почти на каждой системе Linux запущен демон cron. За подробностями о работе cron и crontab обратитесь к теме 213. Вообще говоря, следует помнить, что источником потенциальной проблемы могут стать сценарии, выполняющиеся с временными интервалами. Возможно то, что, по вашему мнению, является проблемой ядра или приложения, происходит от не связанных с ними сценариев, которые запускает cron за вашей спиной.

Экстремальный метод состоит в полном прекращении работы cron. Номер его процесса может быть найден с помощью `ps ax | grep cron`, а `kill` может его прекратить. Менее крутая мера состоит в редактировании /etc/crontab, чтобы выполнялся более традиционный набор задач; также, отредактировав /etc/cron.allow и /etc/cron.deny, убедитесь, что запланированные задачи пользователей отменены. Хотя пользователи и не имеют достаточных прав, чтобы вызвать проблемы для всей системы, по-хорошему, первым шагом стоит временно заблокировать пользовательскую конфигурацию crontab и посмотреть, разрешило ли это ваши проблемы.

# Подготовка к сдаче экзамена LPI: Конфигурирование сети

*Intermediate Level Administration (LPIC-2) тема 205*

Дэвид (David) Мертц (Mertz), Developer, Gnosis Software, Inc.

**Описание:** Это первое из семи руководств, описывающих сетевое администрирование в Linux®. В этом руководстве Дэвид Мертц расскажет, как сконфигурировать простую сеть TCP/IP, начиная с уровня оборудования (обычно Ethernet, модем, ISDN или 802.11), и заканчивая сетевой адресацией. О серверах более высокого уровня, которые могут работать в таких сетях, рассказано подробно в последующих руководствах.

[Больше статей из этой серии](#)

**Дата:** 08.11.2005

**Уровень сложности:** средний

## Прежде чем начать

Посмотрите, чему могут научить эти руководства, и какую пользу вы можете из них извлечь.

## Об этих руководствах

Институт [Linux Professional Institute](#)(LPI) осуществляет сертификацию системных администраторов Linux по двум уровням: *junior level* (также называемый "certification level 1") и *intermediate level* (также называемый "certification level 2"). Чтобы достигнуть certification level 1, вы должны сдать экзамены 101 и 102; чтобы достигнуть certification level 2, вы должны сдать экзамены 201 и 202.

developerWorks предлагает руководства для самоподготовки по каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем, а каждой теме соответствует свое руководство на developerWorks. Что касается экзамена LPI 202, то вот семь его тем и соответствующие им руководства на developerWorks:

*Таблица 1. Экзамен LPI 202: Руководства и темы*

Тема экзамена LPI 202	руководство на developerWorks	Краткая информация о руководстве
Тема 205	Подготовка к экзамену LPI 202 (тема 205): Конфигурирование сети	(Это руководство) Изучите, как настроить простую сеть TCP/IP, начиная с уровня оборудования (обычно Ethernet, модем, ISDN или 802.11), и заканчивая сетевой адресацией. Смотри подробные <a href="#">задачи</a> ниже.
Тема 206	Подготовка к экзамену LPI 202 (тема 206): Почта и новости	Скоро будет готово
Тема 207	Подготовка к экзамену LPI 202 (тема 207): DNS	Скоро будет готово
Тема 208	Подготовка к экзамену LPI 202 (тема 208):	Скоро будет готово

## Web сервисы

Тема 210	Подготовка к экзамену LPI 202 (тема 210): Управление сетевым клиентом	Скоро будет готово
Тема 212	Подготовка к экзамену LPI 202 (тема 212): Безопасность системы	Скоро будет готово
Тема 214	Подготовка к экзамену LPI 202 (тема 214): Разрешение сетевых проблем	Скоро будет готово

Чтобы начать подготовку на certification level 1, смотри [руководства LPI на developerWorks по экзамену 101](#). Чтобы приготовиться к другому экзамену на certification level 2, смотрите [руководства LPI на developerWorks по экзамену 201](#). Узнайте больше о [всей серии руководств LPI на developerWorks](#).

Linux Professional Institute не рекомендует никаких сторонних материалов для подготовки к экзаменам и техник в частности. За подробностями обращайтесь в[info@lpi.org](mailto:info@lpi.org).

## Об этом руководстве

Добро пожаловать в "Конфигурирование сети", первое из семи руководств, описывающих сетевое администрирование в Linux. В этом руководстве вы научитесь, как сконфигурировать простую сеть TCP/IP , начиная с уровня оборудования (обычно Ethernet, модем, ISDN или 802.11), и заканчивая сетевой адресацией. О серверах более высокого уровня, которые могут работать в таких сетях, рассказано подробно в последующих руководствах.

Это руководство организовано в соответствии с требованиями LPI по каждой теме. На экзамене можете ожидать больше вопросов по темам с более высокими весами.

*Таблица 2. Конфигурирование сети: Экзаменационные задачи этого руководства*

Экзаменационная задача LPI	Вес задачи	Краткое описание задачи
2.205.1 <a href="#">Простое конфигурирование сети</a>	Вес 5	Настроить сетевое устройство для подключения к LAN или WAN. Задача включает в себя возможность взаимодействия между разными подсетями одной сети, настройке телефонного доступа с помощью mgetty, настройке телефонного доступа с помощью модема или ISDN, настройки протоколов аутентификации, таких как PAP и CHAP, и настройке протоколирования TCP/IP.
2.205.2 <a href="#">Сложное конфигурирование сети и разрешение проблем</a>	Вес 3	Сконфигурировать сетевое устройство для реализации различных схем аутентификации. Задача включает в себя настройку нескольких профилей для устройства, настройку виртуальной частной сети, а также вопросы разрешения сетевых проблем и сетевого взаимодействия.

## Предварительные рассуждения

Чтобы получить максимум пользы от этого руководства, у вас должны быть знания об основах Linux, а также работающая система с Linux, на которой вы можете выполнять команды этого руководства.

# Подготовка к сдаче экзамена LPI: Конфигурирование сети

*Intermediate Level Administration (LPIC-2) тема 205*

[Дэвид \(David\) Мертц \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** Это первое из семи руководств, описывающих сетевое администрирование в Linux®. В этом руководстве Дэвид Мертц расскажет, как сконфигурировать простую сеть TCP/IP, начиная с уровня оборудования (обычно Ethernet, модем, ISDN или 802.11), и заканчивая сетевой адресацией. О серверах более высокого уровня, которые могут работать в таких сетях, рассказано подробно в последующих руководствах.

**Дата:** 08.11.2005

**Уровень сложности:** средний

Подготовка к сдаче экзамена LPI: Конфигурирование сети

*Intermediate Level Administration (LPIC-2) тема 205*

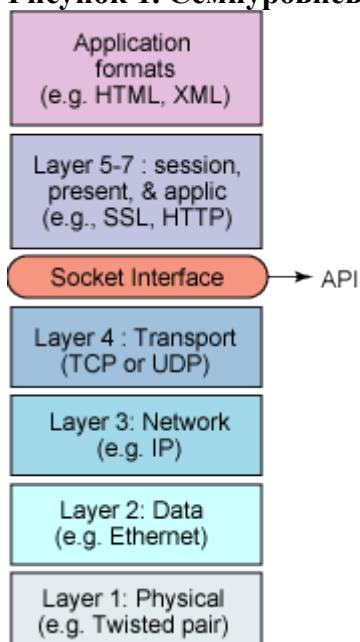
[Дэвид \(David\) Мертц \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** Это первое из семи руководств, описывающих сетевое администрирование в Linux®. В этом руководстве Дэвид Мертц расскажет, как сконфигурировать простую сеть TCP/IP, начиная с уровня оборудования (обычно Ethernet, модем, ISDN или 802.11), и заканчивая сетевой адресацией. О серверах более высокого уровня, которые могут работать в таких сетях, рассказано подробно в последующих руководствах.

## О настройке сети

При обсуждении сетей в Linux и настройке сети следует держать в уме *семиуровневую модель OSI*:

**Рисунок 1. Семиуровневая модель OSI**



То, что мы называем "настройкой сети" на самом деле затрагивает настройку параметров второго и третьего уровней -- *канального и сетевого* -- а также интерфейсов между ними. На практике это включает в себя настройку Ethernet или последовательных интерфейсов наподобие модема для канального уровня, и настройку протокола Internet Protocol (IP) для сетевого уровня. Последующие руководства этой серии рассматривают более высокие уровни организации, хотя большинство серверных приложений, обсуждаемых здесь, не различают четко все семь уровней (или даже четыре верхних уровня, на которых они работают).

Первый уровень это *физический уровень*, провода (или же беспроводные каналы) и связи между ними. Реальному сетевому администратору необходимо быть готовым к исследованию кабелей и установке новой сетевой периферии время от времени (однако эти вопросы выходят за рамки данных руководств). Ясно, что плохой провод, сетевая карта или сломанный коннектор могут создать проблемы в сети, также как и правильно настроенное программное обеспечение.

Четвертый уровень это *транспортный уровень*; а именно TCP или UDP в IP сетях. TCP и UDP используются на верхних уровнях посредством Berkeley Sockets Interface, хорошо оттестированной библиотеке, встречаемой на всех современных компьютерных системах. О том, как приложения (как те, о которых рассказано далее в этой серии руководств) используют TCP или UDP смотрите руководства "[Программирование сокетов в Linux, Часть 1](#)" и "[Программирование сокетов в Linux, Часть 2](#)".

## Другие источники

Как и для большинства инструментов Linux, man-страницы содержат ценную информацию. За более подробной информацией можно обратиться в проект Linux Documentation Project, содержащий много полезных документов, особенно HOWTO. Множество книг по сетям Linux может быть особенно полезным, как например книга издательства O'Reilly под названием *Сетевое администрирование TCP/IP* автора Крэйга Ханта. Вы найдете эти и другие источники в разделе [Ресурсы](#) этого руководства.

## Настройка сети

### Протокол разрешения адреса

Первое, что необходимо помнить об устройствах Ethernet -- как беспроводных 802.11a/b/g, так и более традиционных сетевых карт CAT5/CAT6 -- это то, что у каждого Ethernet устройства имеется уникальный 6 байтный идентификатор. Эти идентификаторы распределены по группам, каждая из которых присвоена производителю; вы можете посмотреть эти группы в IANA. Ethernet в общем "просто работает" на физическом уровне, однако системе требуется отобразить идентификатор Ethernet на используемый IP адрес, чтобы была возможность работы с IP.

Протокол Address Resolution Protocol (ARP) позволяет машинам узнавать IP адреса друг друга внутри локальной Ethernet сети. Что касается протокола, ARP в основном реализован в драйвере сетевого устройства (как модуль ядра); инструмент arp позволяет вам посмотреть статус системы ARP и немного ее настроить. В данный момент мы предположим, что у каждой машины есть свой IP адрес, либо статический либо полученный с помощью DHCP.

Когда система Linux (или любое другое устройство Ethernet) желает обратиться к IP адресу, то с помощью широковещательного запроса Ethernet ARP посылает сообщение с запросом "кто есть X.X.X.X сообщите Y.Y.Y.Y". Целевая система формирует ARP ответ "X.X.X.X это hh:hh:hh:hh:hh" и посыпает его запрашивающему устройству. Ответ ARP кэшируется короткое время в /proc/net/arp, чтобы избежать постоянного восстановления отображения между аппаратными Ethernet адресами и IP адресами.

Горри Фэйрхерст предоставляет хорошее описание ARP (смотри [Ресурсы](#)).

## Утилита arp

Утилита Linux arp позволяет вам изучать и модифицировать статус ARP отображений. Простейший доклад о статусе может выглядеть как в Листинге 1:

### Листинг 1. Доклад о статусе ARP

```
$ arp -n
Address      HWtype  HWaddress          Flags Mask   Iface
192.168.2.1   ether   00:03:2F:09:61:C7   C      00:03:2F:09:61:C7  eth0
```

Здесь говорится, что определенному устройству назначен в этой сети адрес 192.168.2.1 (судя по виду, этот адрес соответствует маршрутизатору\шлюзом, что в данном случае так и есть). Тот факт, что только одна запись содержится в этом списке, не означает, что в сети больше не существует других устройств, так как записи ARP других устройств могут быть просрочены. ARP стирает записи после короткого промежутка времени -- в течение нескольких минут, а не секунд или часов -- чтобы позволить сетям самим реконфигурироваться в случае добавления или удаления устройств или изменения установок на машинах. Кэшируя запись ARP в течение короткого времени, можно не посыпать новые запросы во время работы большинства сетевых сессий.

Любой вид IP запроса машины, которая может находиться в локальной сети, заставляет ядро посылать ARP запрос; если получен ответ ARP, то машина добавляется в кэш ARP (как в Листинге 2):

### Листинг 2. Взаимодействие с другими IP адресами

```
$ ping -c 1 192.168.2.101 > /dev/null
$ ping -c 1 192.168.2.101 > /dev/null
$ ping -c 1 192.168.2.102 > /dev/null
$ ping -c 1 192.168.32.32 > /dev/null
$ ping -c 1 192.168.32.32 > /dev/null
$ arp -n
Address      HWtype  HWaddress          Flags Mask   Iface
192.168.2.1   ether   00:03:2F:09:61:C7   C      00:03:2F:09:61:C7  eth0
192.168.2.101  ether   00:30:65:2C:01:11   C      00:30:65:2C:01:11  eth0
192.168.2.100  ether   00:11:24:9D:1E:4B   C      00:11:24:9D:1E:4B  eth0
192.168.2.102  ether   00:48:54:83:82:AD   C      00:48:54:83:82:AD  eth0
```

В этом случае, первые четыре адреса действительно существуют в сети Ethernet, но 192.168.32.32 не существует, поэтому от него ARP ответ не получен. Заметим, что если вам удалось подключиться к адресам не через локальный маршрут, то в кэш ARP ничего не добавится (смотри Листинг 3):

### Листинг 3. Ничего не добавляется в кэш ARP

```
$ ping -c 1 google.com
PING google.com (216.239.57.99) 56(84) bytes of data.
64 bytes from 216.239.57.99: icmp_seq=1 ttl=235 time=109 ms
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

```

rtt min/avg/max/mdev = 109.123/109.123/109.123/0.000 ms
$ arp -n
Address      Hwtype   Hwaddress      Flags Mask    Iface
192.168.2.1   ether     00:03:2F:09:61:C7   C          eth0

```

Google *доступен* (так как маршрут уже настроен), но 216.239.57.99 не локален, поэтому в ARP ничего не добавляется. Седьмое руководство в этой серии, по теме 214, рассматривает проблемы с сетью и демонстрирует, как установить значение ARP вручную.

### **PPP, PAP и CHAP**

Протокол Point-to-Point Protocol (PPP) используется для установления связи с Internet через dial-up модемы, прямые последовательные соединения, DSL, и другие типы связи точка-точка (иногда включая PPPoE как "псевдо-уровень" поверх Ethernet, этот протокол обеспечивает установление подключения). Демон pppd работает совместно с включенным в ядро PPP драйвером, чтобы установить и поддерживать PPP соединение с другой системой (обычно называемой узлом) и установить адреса Internet Protocol (IP) для каждого конца соединения.

PPP, в особенности pppd, запрашивает аутентификацию у второго участника соединения и предоставляет ему свои аутентификационные данные. Такая аутентификация выполняется с использованием простой системы паролей Password Authentication Protocol (PAP) или сессионной системы Challenge Handshake Authentication Protocol (CHAP). Из двух упомянутых, CHAP более безопасен, если обе стороны его поддерживают.

Опции PPP, как правило, хранятся в /etc/ppp/options. Конфигурация PAP осуществляется через файл паролей PAP /etc/ppp/pap-secrets, для CHAP она осуществляется через файл паролей CHAP /etc/ppp/chap-secrets.

### **Файл паролей PAP/CHAP**

Файл /etc/ppp/pap-secrets содержит разделенные пробелами поля *клиента, сервера, пароля, и допустимого локального IP адреса*. Последнее поле может быть пустым (и как правило оно именно пустое при динамическом назначении IP адреса). Файл паролей PAP должен быть сконфигурирован отдельно для каждого пользователя. Хотя PPP -- это протокол взаимодействия равноправных систем, в целях подключения мы будем называть запрашивающую машину клиентом, а ожидающую машину сервером. Например, машина bacchus в моей сети может иметь следующий файл настроек:

#### **Листинг 4. Настройка pap-secrets на bacchus**

```

# Every regular user can use PPP and uses passwords from /etc/passwd
# INBOUND connections
# client    server    secret                acceptable local IP addresses
*          bacchus   ""                   *
chaos      bacchus   chaos-password
# OUTBOUND connections
bacchus    *         bacchus-password

```

Машина bacchus будет принимать соединения от любых обычных пользователей, а также принимать соединения с машины chaos (требуя пароль **chaos-password** в последнем случае). При подключении к другим машинам bacchus будет просто использовать свое

собственное имя и предлагать пароль **bacchus - password** каждому узлу.

Соответственно машина **chaos** в моей сети может содержать следующий файл:

### Листинг 5. Машина **chaos** более консервативна в выборе соединений

```
# client    server   secret           acceptable local IP addresses
chaos      bacchus  chaos-password
bacchus    chaos    bacchus-password
```

Машина **chaos** более консервативна в отношении к кому она будет подключаться. Она обменялась параметрами доступа только с **bacchus**. Вы можете настроить каждый файл **/etc/ppp/options** и определить имя пользователя и пароль, если требуется.

Использование паролей CHAP требует, чтобы оба узла могли аутентифицировать друг друга. При условии, что двусторонняя аутентификация настроена в паролях PAP, файл паролей CHAP может выглядеть, как в приведенных выше примерах.

### Соединение с помощью **mgetty**

Файл паролей PAP может быть использован с функцией **AUTO\_PPP mgetty**. **mgetty 0.99+** уже настроена на запуск **rpppd** с опцией **login**. Она сообщает, **rpppd** надо обратиться за справкой к **/etc/passwd** (и **/etc/shadow** в свою очередь) после того как пользователь передал этот файл.

В общем, программа **getty** может быть настроена, чтобы принимать соединения от последовательных устройств, включая модемы и последовательные порты. Например, для проводной линии или консоли **tty**, вы можете запустить:

```
/sbin/getty 9600 ttyS1
```

в вашем терминале. Для старых телефонных линий с модемом в 9600/2400/1200 бод можно запустить команду:

```
/sbin/getty -mt60 ttyS1 9600,2400,1200.
```

### Настройка маршрутизации

В разделе обсуждения протокола Address Resolution Protocol мы видели, как назначаются адреса в локальной сети. Однако чтобы взаимодействовать с машинами вне локальной сети, необходимо иметь **маршрутизатор**. В общих чертах маршрутизатор -- это просто компьютер, который подсоединяется к нескольким сетям, и поэтому может брать пакеты из одной сети и передавать их в другие. Именно отсюда и пошло название "Internet": это "сеть, состоящая из сетей", в которой каждый маршрутизатор, в конечном счете, может достучаться до любой другой сети, которая "подключена к Internet."

Пятое руководство этой серии по теме 210 рассматривает управление клиентом сети и DHCP. DHCP назначит как IP адреса, так и адрес маршрутизатора. Однако, если у клиента фиксированный IP адрес, или в целях тестирования, команда Linux **route** позволит вам просмотреть и модифицировать таблицы маршрутизации. Более новая команда **ip** также позволит вам модифицировать таблицы маршрутизации, используя более мощный синтаксис.

Таблица маршрутизации просто позволяет вам определить, через какой маршрутизатор или узел посыпать пакет, на основе определенного шаблона в адресе. Шаблон адреса определяется комбинацией адреса и **маски подсети**. Маска подсети -- это битовый шаблон, обычно представляется в форме групп чисел, разделенных точками, которые сообщают ядру о том, какие биты адреса доставки считать как **сетевой адрес**, а какие оставшиеся биты считать как **подсеть**. Команда **ip** может принять упрощенный **/NN** формат битовых масок. В

общем, в маске и адресе нулевые биты это "метасимволы".

Например, простая сеть с одним внешним шлюзом может содержать таблицу маршрутизации как в Листинге 6:

#### Листинг 6. Типичная простая таблица маршрутизации

```
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref Use Iface
192.168.2.0     0.0.0.0       255.255.255.0   U      0      0      0 eth0
0.0.0.0          192.168.2.1  0.0.0.0       UG     0      0      0 eth0
```

Это значит, что сетевой пакет на любой IP адрес, который соответствует шаблону "192.168.2.\*" предназначен компьютеру из локальной сети и будет направлен прямо на нужный узел (полученный при помощи ARP). Все остальные пакеты будут посыпаться на маршрутизатор "192.168.2.1", который перенаправит их по назначению. Машина 192.168.2.1 должна быть подсоединенена к одной или нескольким внешним сетям.

Однако в более сложном случае вы можете по другому определить шаблон назначения. Придумаем пример, положим, что вы хотите направить определенные адреса /16 через другие шлюзы. Вы можете сделать это, как показано в Листинге 7:

#### Листинг 7. Изменение маршрута сетей /16

```
$ route add -net 216.109.0.0 netmask 255.255.0.0 gw 192.168.2.2
$ route add -net 216.239.0.0 netmask 255.255.0.0 gw 192.168.2.3
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref Use Iface
192.168.2.0     0.0.0.0       255.255.255.0   U      0      0      0 eth0
216.109.0.0     192.168.2.2  255.255.0.0     UG     0      0      0 eth0
216.239.0.0     192.168.2.3  255.255.0.0     UG     0      0      0 eth0
0.0.0.0          192.168.2.1  0.0.0.0       UG     0      0      0 eth0
```

Пакеты на адреса вида "216.109.\*" и "216.239.\*" будут теперь направляться через маршрутизаторы 192.168.2.2 и 192.168.2.3, соответственно (оба находятся в локальной сети). Пакеты на локальные адреса или адреса, несоответствующие шаблону, будут направляться так же, как и раньше. Вы можете использовать команду [route delete](#), чтобы удалить маршруты.

### Сложная настройка сети и разрешение проблем

#### О сетевых утилитах

Linux поставляется с набором стандартных утилит, которые вы можете использовать для настройки и разрешения проблем с сетью. Хотя большая часть сетевого кода Linux находится в самом ядре, почти все, что касается поведения сети, можно настроить с помощью утилит командной строки. Многие дистрибутивы поставляются с инструментами более высокого уровня, возможно графическими. Но они могут сделать туже самую работу, что и инструменты командной строки.

## Утилита ping

Самый простой способ проверить, имеет ли узел с Linux доступ к IP адресу (или К именованному узлу, в случае настроенных DNS и/или /etc/hosts) состоит в использовании утилиты **ping**. ping работает на уровне IP и не полагается на канальный уровень как TCP или UDP. ping вместо них использует протокол Internet Control Message Protocol (ICMP). Если вы не можете достичь узла с помощью ping, то почти наверняка вы не сможете с ним связаться и с помощью других инструментов, поэтому ping всегда является первым шагом в установлении возможности подключения к узлу (man ping может предоставить описание параметров команды).

По умолчанию, ping посылает сообщение каждые две секунды до тех пор, пока не будет прервана ее работа, однако вы можете изменить время, ограничить число сообщений и подробности вывода. Во время работы ping выводит время путешествия пакета и число потерянных пакетов, но в большинстве случаев вы либо сможете получить ответ от узла, либо нет. В Листинге 8 приведены некоторые примеры:

### Листинг 8. Работа ping с локальными и нелокальными узлами

```
$ ping -c 2 -i 2 google.com
PING google.com (216.239.37.99): 56 data bytes
64 bytes from 216.239.37.99: icmp_seq=0 ttl=237 time=43.861 ms
64 bytes from 216.239.37.99: icmp_seq=1 ttl=237 time=36.956 ms

--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 36.956/40.408/43.861 ms

$ ping 192.168.2.102
PING 192.168.2.102 (192.168.2.102): 56 data bytes
64 bytes from 192.168.2.102: icmp_seq=0 ttl=255 time=4.64 ms
64 bytes from 192.168.2.102: icmp_seq=1 ttl=255 time=2.176 ms
^C
--- 192.168.2.102 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 2.176/3.408/4.64 ms
```

## Утилита ifconfig

Сетевые интерфейсы настраиваются с помощью инструмента **ifconfig**. Обычно она запускается как часть процесса инициализации, но в некоторых случаях интерфейсы могут быть модифицированы и настроены позже (особенно при отладке). Если вы запустите ifconfig без ключей, то увидите отображение текущего сетевого статуса. Вы можете использовать **ifconfig <interface> up** и **ifconfig <interface> down**, чтобы запустить и остановить сетевые интерфейсы. Некоторые ключи изменяют формат отображения или ограничивают вывод только для конкретных интерфейсов. В man ifconfig можно узнать подробности.

Дополнительная информация может выглядеть как в Листинге 9:

### Листинг 9. Использование ifconfig для просмотра информации о сетевых интерфейсах

```
$ ifconfig
eth0 Link encap:Ethernet Hwaddr 00:12:F0:21:4C:F8
      inet addr:192.168.2.103 Bcast:192.168.2.255 Mask:255.255.255.0
```

```

inet6 addr: fe80::212:f0ff:fe21:4cf8/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:540 errors:0 dropped:0 overruns:0 frame:0
TX packets:233 errors:0 dropped:0 overruns:0 carrier:1
collisions:0 txqueuelen:1000
RX bytes:49600 (48.4 KiB) TX bytes:42067 (41.0 KiB)
Interrupt:21 Base address:0xc000 Memory:ffcfe000-ffcfefff

ppp0 Link encap:Point-Point Protocol
inet addr:10.144.153.104 P-t-P:10.144.153.51 Mask:255.255.255.0
UP POINTOPOINT RUNNING MTU:552 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0
TX packets:0 errors:0 dropped:0 overruns:0

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:4043 errors:0 dropped:0 overruns:0 frame:0
TX packets:4043 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:368044 (359.4 KiB) TX bytes:368044 (359.4 KiB)

```

В этом листинге видно, что настроено две сети, одна Ethernet и одна PPP (и еще присутствует повсеместный локальный интерфейс loopback). В других случаях у вас может быть настроено несколько интерфейсов Ethernet или других типов интерфейсов. Если это так, то говорят, что система *сильно связана*.

### Утилита netstat

Утилиты Linux могут иметь схожий функционал. Инструмент **netstat** отображает информацию, которую также можно получить у нескольких утилит, как ifconfig и route. Вы также можете узнать общую расширенную статистику о сетевой активности. Например:

#### Листинг 10. Отчет сетевой статистики

```

$ netstat -s
Ip:
    12317 total packets received
    0 forwarded
    0 incoming packets discarded
    12255 incoming packets delivered
    11978 requests sent out
Icmp:
    1 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        echo replies: 1
    0 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
Tcp:
    7 active connections openings
    5 passive connection openings
    0 failed connection attempts
    0 connection resets received
    3 connections established

```

```

11987 segments received
11885 segments send out
0 segments retransmitted
0 bad segments received.
3 resets sent
Udp:
 101 packets received
 0 packets to unknown port received.
 0 packet receive errors
 92 packets sent
TcpExt:
 1 TCP sockets finished time wait in fast timer
 1490 delayed acks sent
 Quick ack mode was activated 5 times
 3632 packets directly queued to recvmsg prequeue.
 126114 of bytes directly received from backlog
 161977 of bytes directly received from prequeue
 1751 packet headers predicted
 3469 packets header predicted and directly queued to user
 17 acknowledgments not containing data received
 4696 predicted acknowledgments
 0 TCP data loss events

```

## Другие утилиты

Есть также другие утилиты, о которых вы должны знать при настройке сети. Как обычно соответствующие им man-страницы содержат полную информацию по их использованию. Подробно они обсуждаются в седьмом руководстве по теме 214, которая рассматривает вопросы разрешения проблем, этой серии руководств.

**tcpdump** позволяет отслеживать все пакеты, которые проходят через сетевые интерфейсы, опционально можно ограничиться определенными интерфейсами или произвести фильтрацию по различным критериям. Часто такой отчет, который потом обрабатывается текстовыми утилитами, полезен при диагностике проблемы сети. Например, вы можете исследовать пакеты, которые приходят от конкретного удаленного узла.

**lsof** выводит список всех открытых файлов в работающей Linux системе. Но в частности, вы можете использовать опцию **lsof -i**, чтобы просмотреть только псевдофайлы на наличие определенного IP соединения и вообще все сетевые соединения. Например:

### Листинг 11. Использование lsof для просмотра псевдофайлов на наличие соединений

```
$ lsof -i
COMMAND      PID USER      FD      TYPE DEVICE SIZE NODE
          NAME
vino-serv 7812 dqm    33u   IPv4  12824
          TCP *:5900 (LISTEN)
gnome-cup 7832 dqm    18u   IPv4  12865
          TCP localhost.localdomain:32771->localhost.localdomain:ipp (ESTABLISHED)
telnet    8909 dqm     3u   IPv4  15771
          TCP 192.168.2.103:32777->192.168.2.102:telnet (ESTABLISHED)
```

**nc** и **netcat** -- это псевдонимы. netcat это простая утилита UNIX, которая читает и пишет данные по сети, используя протокол TCP или UDP. Это "back-end" инструмент, который

можно использовать в других программах или скриптах. Во многих отношениях netcat похожа на telnet, но более гибка в плане работы с UDP и пересылке двоичных данных.

## Ресурсы

### Научиться

- Просмотрите [весь список руководств подготовки к LP I](#)на developerWorks, чтобы узнать об основах Linux и подготовиться к сертификации на системного администратора.
- В [Программе LPIC](#) вы найдете списки заданий, вопросы, и подробные требования для каждого из трех уровней сертификации по системному администрированию в Linux Professional Institute.
- Прочтайте руководство Квена Леве по [Сборке собственного ядра](#), чтобы узнать о том, как это можно сделать.
- Узнайте о том, как приложения используют TCP или UDP из этих руководств:
  - "[Программирование сокетов в Linux, Часть 1](#)" (developerWorks, Октябрь 2003)
  - "[Программирование сокетов в Linux, Часть 2](#)" (developerWorks, Январь 2004)
- [Сетевое администрирование TCP/IP, Третье издание](#) Крейга Ханта (O'Reilly, Апрель 2002) является превосходным источником по сетям в Linux.
- Узнайте [О номерах ethernet](#) в IANA.
- Горри Фэйрхерст предоставляет [хорошее описание ARP](#).
- Во многих Группах Пользователей Linux есть местные или удаленные группы подготовки к сдаче LPI экзаменов -- вот список из более чем [700 групп по всему миру](#).
- Проект документации [Linux Documentation Project](#) содержит множество полезных документов, особенно HOWTO.
- Вы можете найти больше [руководств для разработчиков Linux](#) в разделе [Linux на developerWorks](#).

### Получить продукты и технологии

- Получите исходные коды ядра Linux в [Linux Kernel Archives](#).
- Создайте свой следующий проект на Linux с помощью [пробных программ IBM](#), доступных для прямого скачивания с сайта developerWorks.

# Учебник для экзамена LPI: Почта и новости

*Администрирование для специалистов (LPIC-2) тема 206*

[Дэвид \(David\) Мертз \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** Это второй из семи учебников, посвященных сетевому администрированию Linux для специалистов®. В этом учебнике Дэвид Мэртз [David Mertz] рассматривает использование Linux в качестве почтового сервера или сервера новостей. Вообще, электронная почта (e-mail) это, вероятно, основное предназначение Internet, и, возможно, что Linux является наилучшей платформой для работы сервисов электронной почты. В этом учебнике рассматриваются почтовые протоколы передачи данных (mail transport), локальная фильтрация почты, и программное обеспечение для обслуживания списков рассылки. А также кратко обсуждается серверное программное обеспечение для NNTP протокола.

[Больше статей из этой серии](#)

**Дата:** 31.01.2007

**Уровень сложности:** средний

## Перед тем как начать

Узнайте, чему может научить вас этот учебник и как извлечь из него максимальную пользу.

## Об этой серии

[Linux Professional Institute](#) (LPI) производит сертификацию системных администраторов Linux двух уровней: *junior level* [для начинающих] (также называемый "уровень сертификации 1") *intermediate level* [для специалистов] (также называемый "уровень сертификации 2"). Для прохождения первого уровня сертификации, вы должны сдать экзамены LPI 101 и LPI 102; для прохождения второго уровня, вы должны сдать экзамены LPI 201 и LPI 202.

developerWorks предоставляет учебники, призванные помочь вам в подготовке к каждому из этих четырех экзаменов. Каждый экзамен охватывает несколько тем, и каждая тема имеет соответствующий учебник developerWorks для самостоятельной подготовки по ней. Экзамен LPI 202 содержит семь тем, и соответствующие им учебники от developerWorks это:

Таблица 1. Экзамен LPI 202: Учебники и темы

Тема экзамена	Учебник от developerWorks	Краткое описание учебника
LPI 202	<a href="#">Учебник для экзамена LPI 202 (тема 205): Настройка сети</a>	Узнайте как настроить простую сеть на основе TCP/IP, от уровня аппаратного обеспечения (обычно Ethernet, модем, ISDN или 802.11) до распределения сетевых адресов.
Тема 206	Учебник для экзамена LPI 202 (тема 206): Почта и новости	(Настоящий учебник) Узнайте как использовать Linux в качестве почтового сервера и сервера новостей. Прочтите о почтовых протоколах передачи данных, локальной фильтрации почты, программном обеспечении для поддержки почтовых рассылок и серверном программном обеспечении для протокола

		NNTP. Более детально смотри <a href="#">требования</a> ниже.
Тема 207	Учебник для экзамена LPI 202 (тема 207): DNS	Скоро выйдет
Тема 208	Учебник для экзамена LPI 202 (тема 208): Web-службы	Скоро выйдет
Тема 210	Учебник для экзамена LPI 202 (тема 210): Управление клиентами сети	Скоро выйдет
Тема 212	Учебник для экзамена LPI 202 (тема 212): Безопасность Системы	Скоро выйдет
Тема 214	Учебник для экзамена LPI 202 (тема 214): Устранение проблем сети	Скоро выйдет

Для подготовки к первому уровню сертификации смотри [учебники от developerWorks для экзамена LPI 101](#). Для подготовки ко второму уровню сертификации смотри [учебники от developerWorks для экзамена LPI 201](#). Прочти больше о [полном наборе учебников LPI от developerWorks](#).

Профессиональный Институт Linux (The Linux Professional Institute) не одобряет любых учебных материалов или технологий для подготовки к экзаменам от третьих лиц. За разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## Об этом учебнике

Добро пожаловать в "почта и новости", второй учебник из семи, посвященных администрированию сетей в Linux для специалистов. В этом учебнике вы узнаете, как использовать Linux в качестве почтового сервера и сервера новостей. В этом учебнике рассматриваются почтовые протоколы передачи данных, локальная фильтрация почты, и программное обеспечение для поддержки почтовых рассылок. Также кратко описывается серверное ПО для протокола NNTP.

Этот учебник скомпонован в соответствии с требованиями LPI по данной теме. Грубо говоря, на экзамене ожидайте больше вопросов по темам, имеющим больший рейтинг.

*Таблица 2. Почта и новости: Экзаменационные темы охватываемые в этом учебнике*

Экзаменационная тема LPI	Рейтинг темы	Краткое описание темы
2.206.1 <a href="#">Настройка списков рассылки</a>	Рейтинг 1	Установка и поддержка списков рассылки с использованием Majordomo. Отслеживание проблем Majordomo, путем просмотра журналов [logs]

		Majordomo.
2.206.2 <a href="#"><u>Использование Sendmail</u></a>	Рейтинг 4	Управление конфигурацией Sendmail, включая e-mail псевдонимы, почтовые квоты и виртуальные почтовые домены. Эта тема включает в себя настройку внутренних почтовых станций и мониторинг SMTP серверов.
2.206.3 <a href="#"><u>Управление почтовым трафиком</u></a>	Рейтинг 3	Использование программного обеспечения управления почтой клиентов для фильтрации, сортировки и отслеживания входящей почты пользователей. Эта тема включает описание использования такого ПО, как Procmail на сервере и на стороне клиента.
2.206.4 <a href="#"><u>Предоставление новостей</u></a>	Рейтинг 1	Установка и настройка почтовых серверов с использованием INN. Эта тема включает в себя настройку и мониторинг обслуживаемых групп новостей.

### Предварительные замечания

Для того, чтобы извлечь из этого учебника максимум, вы должны обладать базовыми знаниями о Linux, а также иметь рабочую версию системы Linux, в которой вы будете практиковаться в использовании описываемых в учебнике команд.

## Учебник для экзамена LPI: Почта и новости

*Администрирование для специалистов (LPIC-2) тема 206*

[Дэвид \(David\) Мертз \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** Это второй из семи учебников, посвященных сетевому администрированию Linux для специалистов®. В этом учебнике Дэвид Мэртз [David Mertz] рассматривает использование Linux в качестве почтового сервера или сервера новостей. Вообще, электронная почта (e-mail) это, вероятно, основное предназначение Internet, и, возможно, что Linux является наилучшей платформой для работы сервисов электронной почты. В этом учебнике рассматриваются почтовые протоколы передачи данных (mail transport), локальная фильтрация почты, и программное обеспечение для обслуживания списков рассылки. А также кратко обсуждается серверное программное обеспечение для NNTP протокола.

### О почте и новостях

Широкое использование Linux для почтовых и новостных серверов с течением времени привело к разработке множества инструментов. В момент разработки сертификационных экзаменов LPI наиболее популярными инструментами были: *Sendmail* для транспортировки почты, *Procmail* для работы с локальной почтой, *Majordomo* для списков рассылки и *innd* (InterNetNews daemon) для NNTP. Последний из них, вероятно, все еще является стандартным выбором для новостных групп, однако, не смотря на его техническую мощь, протокол NNTP слегка теряется на фоне почтовых рассылок и Web-форумов.

Что касается других инструментов, *Sendmail* и *Procmail* все еще широко используются, хотя уже не столь повсеместно как ранее. Наиболее популярным обновлением или заменой

Sendmail является postfix, содержащий средства обратной совместимости с Sendmail. Область управления локальной почтой насчитывает множество альтернатив, но Procmail все еще популярен. С другой стороны, Majordomo в настоящее время выглядит как анахронизм. Также как Majordomo во многом заменил более раннее ПО listserv, mailman в последнее время вытесняет Majordomo. Однако, для соответствия текущим темам LPI, в этом учебнике рассматривается Majordomo.

## Другие ресурсы

Как и для большинства Linux-инструментов, очень полезно обратиться к [так страницам](#) любой из рассматриваемых утилит. Версии и ключи могут изменяться для утилиты, ядра или различных дистрибутивов. За более полной информацией обратитесь к проекту Linux Documentation Project, содержащему много разнообразных полезных документов, особенно HOWTO. Смотри ссылку в разделе [Ресурсы](#). Опубликовано множество книг о сетях в Linux; Я обнаружил, что книга издательства O'Reilly *TCP/IP Network Administration* (*Администрирование сетей TCP/IP*) Крейга Ханта (Craig Hunt) может быть весьма полезной (найдите самое последнее издание для того времени, когда вы читаете эти строки; смотри ссылку в разделе [Ресурсы](#)).

## Настройка списков рассылки

### Что делает Majordomo?

Программа управления списками рассылки [менеджер списков рассылки] -- это, в основном, локальное расширение программы транспортировки почты (mail transport program -- MTA), например Sendmail. В основном, MTA работают в системах, предоставляющих менеджеру списков рассылки набор адресов, и менеджер изменяет, обрабатывает и, возможно, переотправляет получаемые им сообщения. Некоторые сообщения, получаемые менеджером списков рассылки предназначены для размещения в самом списке рассылки (возможно, ему потребуется проверить разрешение на запись в список(-ки)). Другими сообщениями являются управляющие сообщения, меняющие свойства списка рассылки, такие как настройки конкретного подписчика. Менеджер списков рассылки не выполняет доставку почты самостоятельно, а перекладывает эту функцию на поддерживающую его MTA.

Как указывалось во введении к данному учебнику, Majordomo в настоящий момент не является промышленным стандартом для списков рассылки. Более того, наилучшим выбором для вновь создаваемых списков рассылки вероятно является Mailman. Однако, Majordomo все еще достаточно функционален и установлен на множестве старых систем, которые продолжают работать без проблем (иногда они поддерживают списки, работающие уже много лет).

При выборе версии Majordomo нужно учесть одну тонкость. Несколько лет назад было начато переписывание приложения Majordomo 1.x, под названием Majordomo2. К сожалению, оно провалилось, даже не достигнув стадии релиза. Поскольку Majordomo2 (бета-версия) не рекомендуется к использованию на большом количестве систем, то Majordomo 1.9.5 -- это самый последний стабильный выпуск и потому именно он рассматривается в этом учебнике.

### Установка Majordomo

Вы можете получить архив с программным обеспечением Majordomo на сайте Majordomo (смотри ссылку в [Ресурсах](#)).

После распаковки файла с именем вроде majordomo-1.94.5.tgz обязательно внимательно прочитайте файл INSTALL. Вы должны следовать всем инструкциям, описанным в нем, чтобы получить хорошо работающую систему Majordomo. Сборка системы обычно производится следующим образом `make; make install` -- шаги для большинства

установок из исходных текстов, также как и `make install-wrapper`. Установка может и должна осуществить самопроверку при помощи команды вроде `cd /usr/local/majordomo-1.94.5; ./wrapper config-test` (`make install` выводит детали в виде сообщения).

Перед сборкой Majordomo измените Makefile, а также создайте и/или измените majordomo.cf. В качестве отправной точки для второго из указанных файлов вы можете скопировать из исходного дистрибутива файл sample.cf. В файле Makefile, прописано несколько переменных среды, но наиболее критична и важна из них, вероятно, `W_GROUP`. Это *числовой* gid группы к которой будет относиться Majordomo, почти всегда это группа "daemon". На большинстве систем gid для daemon равен 1, но все же следует проверить это, использовав следующее:

```
$ id daemon  
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

Среди других переменных файла MakefilePERL содержащая путь к интерпретатору, и `W_HOME` содержащая путь установки Majordomo.

Ваш новый файл majordomo.cf также следует отредактировать перед тем как выполнить `make install`. Переменные Perl, которые необходимо будет изменить находятся в основном ниже заголовка файла. Обязательно измените `$whereami` и `$homedir`, а также проверьте другие, чтобы убедиться в верности их значений.

### Готовим Sendmail использовать Majordomo

На последнем шаге установки нужно убедить Sendmail общаться с Majordomo. Внутри файла /etc/sendmail.cf имеются строки похожие на:

```
OA/path/to/majordomo/majordomo.aliases
```

Если для создания конфигурационных файлов Sendmail вы используете процессор M4, то можете использовать строки вроде этой:

```
define(`ALIAS_FILE', `/etc/aliases,/path/to/majordomo/majordomo.aliases')
```

Пример majordomo.aliases содержит несколько образцов переменных:

### Листинг 1. Пример majordomo.aliases

```
majordomo: "|/usr/test/majordomo-1.94.5/wrapper majordomo"  
majordomo-owner: you  
owner-majordomo: you  
test: "|/usr/test/majordomo-1.94.5/wrapper resend -l test test-list"  
test-list: :include:/usr/test/majordomo-1.94.5/lists/test  
owner-test: you  
test-owner: you  
test-request: you
```

Это, конечно же, необходимо отредактировать в соответствии с вашими настройками. В частности "you" означает имя администратора списков (который не обязательно является администратором всей системы).

### Создание нового списка Majordomo

В примере настройки, приведенном выше создается список с именем "test" для адресатов "test-owner", "test-request" и так далее, управляющих списком. При реальном использовании, вы, возможно, захотите иметь списки с другими именами. Для того, чтобы сделать это,

выполните следующее:

1. Переместитесь в каталог \$listdir, как определено в majordomo.cf.
2. Создайте файлы my-list-name и my-list-name.info (отредактируйте соответственно); выполните для них `chmod`, установив права доступа равными 664. Последний файл содержит начало списка.
3. Создайте несколько псевдонимов в вашем файле majordomo.aliases, следуя шаблону примера "test" -- например, "foo-owner", "foo", "foo-request" и тому подобное.
4. Выполните запрос к `subscribe`, `unsubscribe`, `signoff`, и так далее, для членов списка.
5. Создайте архивный каталог в месте, указанном переменными `$filedir` и `$filedir_suffix`.
6. Создайте подкаталог дайджестов [digest subdirectory] внутри `$digest_work_dir`. Используйте то же имя для списка дайджестов [digest list] (например: test-digest).
7. Убедитесь, что владельцем всего является пользователь majordomo, группа majordomo, и может изменяться владельцем и группой (другими словами, права доступа равны 664 для файлов и 775 для каталогов).
8. Выполните команды `config <listname> <listname>admin` для Majordomo. Это заставит его создать стандартные конфигурационные файлы для списка, и отправит их вам обратно.

## Использование Sendmail

### Что делает Sendmail?

Sendmail это Агент Транспортировки почты (Mail Transport Agent -- MTA). Он отправляет, изменяет и доставляет почтовые сообщения в гетерогенных почтовых системах. Если проводить исторические параллели с приложениями для списков рассылки, то Sendmail имеет "постоянную бета" версию с названием Sendmail X которая, как предполагается, будет обновлением/заменой стабильной серии Sendmail 8.x; однако, как Mailman в значительной мере вытеснил Majordomo, так и некоторые другие МТА частично заслонили Sendmail. Выделяется среди новых МТА Postfix, но Qmail и Exim также используются довольно широко. Тем не менее, Sendmail все еще остается (хотя его преимущество практически сошло на нет) наиболее широко используемым МТА для Linux систем. На 16 сентября 2005 года последней стабильной версией Sendmail была 8.13.5.

Не одна, а несколько книг, посвящены Sendmail. Смотри в [Ресурсах](#) список доступных книг. Наиболее всеобъемлющая из них это *Sendmail, Third Edition* (O'Reilly, 2002) от Bryan Costales и Eric Allman. На 1,232 страницах, эта книга описывает намного больше того, чего мы можем только коснуться в этом учебнике.

Хотя Sendmail стандартно поддерживает большое количество почтовых протоколов передачи данных, такие как UUCP, наиболее широко используется Simple Mail Transport Protocol (SMTP -- Простой Почтовый Протокол Передачи данных), включающий также Extended [Расширенный] SMTP (ESMTP) для тел сообщений закодированных при помощи расширенного MIME. Почта, которая не предназначена для передачи другим SMTP хостам доставляется в локальную систему путем размещения сообщений в локальных файлах. Локальные Почтовые Агенты Пользователя (Mail User Agent -- MUA) читают сообщения, которые Sendmail (или другой МТА) размещает в локальных файлах (и часто также забирают почту, используя POP3 или IMAP), и обычно запрашивают у Sendmail доставку исходящих сообщений. Однако некоторые MUA взаимодействуют с SMTP серверами (такими как

Sendmail, локальный или удаленный) напрямую, вместо размещения сообщений в очереди Sendmail для последующей обработки. Стандартная очередь Sendmail расположена в /var/spool/mqueue/.

## Установка Sendmail

Первое, что вам следует сделать -- это получить копию ПО Sendmail с sendmail.org (смотри ссылку в [Ресурсах](#)), например, sendmail.8.13.5.tar.gz. Как обычно распаковать его. В отличие от многих приложений, использующих шаблон **make; make install**, сборка Sendmail осуществляется командой **sh Build**. После начальной сборки, перейдите при помощи **cd** в подкаталог cf/cf/; создайте копию с именем sendmail.mc соответствующего файла \*.mc; измените sendmail.mc; и выполните следующее для создания файла sendmail.cf:

```
$ m4 ./m4/cf.m4 sendmail.mc > sendmail.cf
```

Вы можете также использовать ярлык **sh Build sendmail.cf**. Это может показаться загадочным, но обе указанные команды создают реальную конфигурацию Sendmail из формата более удобного для чтения, с использованием макропроцессора M4. Подлинные файлы sendmail.cf, которые, тем не менее также можно редактировать в ASCII, тяжелы для понимания и должны изменяться вручную минимально.

И наконец, скопируйте бинарный файл sendmail из каталога, в котором он был собран (обычно что-то вроде obj.Linux.2.6.10-5-386.i686/sendmail/sendmail на его основное место, как правило это /usr/sbin/ (сохраните резервную копию старого файла, если они существует), а также скопируйте ваш вариант sendmail.cf в /etc/mail/sendmail.cf. Его так же можно перенести и в подкаталог cf/cf/ командой **sh Build install -cf**. Возможно вам понадобится перейти в режим суперпользователя **su** или **sudo**, чтобы получить разрешение на добавление файлов в соответствующие каталоги.

С Sendmail поставляется несколько утилит: makemap, mailstats и т. д. В каждом соответствующем каталоге имеется файл README и их можно установить командой **sh Build install** выполненной в соответствующем подкаталоге.

## Файл sendmail.cf

Основные сложности и основная функциональность Sendmail заключены в его файле sendmail.cf. Это конфигурационный файл содержит некоторые настройки среды Sendmail, шаблоны адресов для перенаправления и/или доставки по имеющимся алгоритмам.

Два алгоритма перенаправления, которые могут быть использованы -- это **genericstable** и **virtusertable**, позволяющие вам указывать соответствия локальных пользователей и внешних адресов. Для обоих случаев преобразования сначала вы создаете файл псевдонимов в обычном текстовом формате. Например:

### Листинг 2. Исходящее преобразование

david	david.mertz@gmail.com
root	root@gnosis.cx
dqm@gnosis.lan	david.mertz@gmail.com

или, для входящей почты преобразование в локальные учетные записи:

### Листинг 3. Входящее преобразование

david@mail.gnosis.cx	david
david@smtp.gnosis.cx	david

```
david@otherdomain.net      david
@mail.gnosis.cx            %1@external-host.com
owner@list.gnosis.cx       owner%3
jax@bar.com                error:5.7.0:550 Address invalid
```

Для компиляции этих псевдонимов, используйте утилиту `makemap`:

```
$ makemap dbm /etc/mail/virtusertable < inbound
$ makemap hash /etc/mail/genericstable < outbound
```

Разрешение использования этих преобразований может быть настроено с использованием M4 макросов `sendmail.cf` (или в каком-нибудь другом используемом вами конфигурационном файле).

#### Листинг 4. Разрешение преобразований в `sendmail.cf`

```
DOMAIN(gnosis.cx)dnl
FEATURE(`virtusertable', `dbm /etc/mail/virtusertable')dnl
FEATURE(`genericstable', `hash /etc/mail/genericstable')dnl
GENERIC_DOMAIN_FILE(`/etc/mail/generics-domains')dnl
```

Здесь указано несколько элементов. `DOMAIN` макрос, показывающий, что файл вроде `cf/domain/gnosis.cx.m4` используется дополнительными макросами. `FEATURE` макрос, разрешающий использование `virtusertable` и `genericstable`.

`GENERIC_DOMAIN_FILE` макрос, определяющий домены, которые соответствуют преобразованию имен со сжатием в `genericstable`.

Переадресация будет происходить по указанным правилам. В режиме теста (`sendmail -bt`), вы можете проверить выполнение перенаправления для конкретных адресов.

Например, используя `genericstable`, сообщение локальному пользователю "david" будет доставлено на `david.mertz@gmail.com` внешним образом. Предположим, что `localhost` определено в `/etc/mail/generics-domains`, тогда почта для `david@localhost` будет отправлена на тот же адрес.

В другом направлении входящая почта для `david@mail.gnosis.cx` будет доставлена локальному пользователю "david". При помощи `Sendmail` можно управлять несколькими доменами одновременно, так что `david@otherdomain.net` будет доставлено также локально.

Главная мощь этих преобразований заключается в символах подстановки. Любая почта, отправленная на `mail.gnosis.cx`, кроме предназначеннной для специального локального пользователя, будет переадресована на то же имя пользователя на `external-host.com`. Но это простой шаблон. Более интересно то, что `%3` может использоваться для извлечения дополнительной информации об имени, то есть `owner-foo@list.gnosis.cx` и `owner-bar@list.gnosis.cx` будут доставлены локальным пользователям "owner-foo" и "owner-bar" ответственно. Эти локальные пользователи могут быть псевдонимами системами обработки списков рассылки или других автоматических манипуляторов сообщений. В качестве специального случая вы можете вызвать ошибку для указанных адресов вместо того, чтобы перенаправить предназначенную им почту куда бы то ни было.

Все что мы рассмотрели -- это просто рябь на поверхности правил переадресации, которые вы можете добавить в `Sendmail`, но этого достаточно чтобы уловить суть. Приобретите одну из больших книг по данной теме, если хотите изучить это более детально.

## Запуск Sendmail

Sendmail может работать в нескольких режимах. Наиболее распространенный режим -- это работа в качестве демона, работа в фоне с периодической обработкой очереди. Например, выполнение:

```
$ /usr/sbin/sendmail -bd -q10m
```

скажет Sendmail запуститься в качестве демона и проверять свою очередь каждые десять минут. Вы можете также запустить Sendmail единократно для обработки очереди один раз, но не как демон:

```
$ /usr/sbin/sendmail -q
```

Как указывалось выше, Sendmail имеет тестовый режим для проверки правил переадресации. Например (взято из *Linux Network Administrators Guide*; смотри ссылку в [Ресурсах](#)):

## Листинг 5. Тестовый режим Sendmail

```
$ /usr/sbin/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 isaac@vstdout.vbrew.com
rewrite: ruleset 3 input: isaac @ vstdout . vbrew . com
rewrite: ruleset 96 input: isaac < @ vstdout . vbrew . com >
rewrite: ruleset 96 returns: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 3 returns: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 0 input: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 199 input: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 199 returns: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 98 input: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 98 returns: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 198 input: isaac < @ vstdout . vbrew . com . >
rewrite: ruleset 198 returns: $# local $: isaac
rewrite: ruleset 0 returns: $# local $: isaac
```

## Управление почтовым трафиком

### Что делает Procmail?

Procmail это небольшой почтовый процессор. В сущности, как только Sendmail или другой MTA доставит почту в локальные почтовые ящики, вы можете использовать MUA для обработки почты в вашей паке inbox [Входящие]. Вы сохраняете некоторые сообщения в различные папки; другие вы удаляете; вы перенаправляете какие-то сообщения другим интересующимся абонентам; на какие-то отвечаете; и так далее. Выполнение этих задач в MUA является ручным и интерактивным процессом, и в своей массе отнимает много времени.

Procmail это программа, которая может выполнять эти задачи автоматически, как только вы укажете ей правила работы. Естественно, когда вы отвечаете на письмо своей мамы, то требуется оказать некое персональное внимание, но для большого класса других сообщений можно заранее точно сказать что нужно сделать при получении данного сообщения.

Правила, которые управляют автоматической обработкой писем, могут реагировать на специфические заголовочные поля, на определенное содержимое в теле сообщения или даже основываться на результатах обращения к более специфичным и специализированным внешним программам, таким как статистический спам-фильтр.

## Включение Procmail

Procmail вероятно уже установлен вместе с вашим дистрибутивом Linux. Если нет, то вы можете получить архив с исходными текстами на [procmail.org](http://procmail.org) (смотри [Ресурсы](#)). В момент написания данного текста, последней версией была 3.22. Вы можете также установить Procmail как бинарный файл, используя систему установки пакетов вашего дистрибутива Linux (например, в Debian: `apt-get install procmail`). Сборка из исходных текстов является стандартной: `make install`. Все, что необходимо Procmail для работы -- это бинарный файл procmail и конфигурационный файл `~/.procmailrc` (или, возможно, глобальный файл `/etc/procmailrc`).

Кроме установки Procmail, в первую очередь вам необходимо настроить вашу локальную почтовую систему для использования Procmail. Старый алгоритм обработки почты посредством Procmail заключается в использовании файла `.forward`; это все еще часто работает на уровне единичного пользователя. Обычно пользователь создает файл `~/.forward`, содержащий нечто вроде этого:

```
|/usr/local/bin/procmail
```

Этот файл перенаправляет все входящие сообщения к Procmail. Однако, более удобный и распространенный способ использования Procmail - это настроить ваш MTA передавать ваши письма напрямую Procmail. В Sendmail, это выполняется путем включения функции `local_procmail`, поместив в ваш файл `sendmail.mc` следующее:

```
FEATURE(`local_procmail', `/usr/bin/procmail', `procmail -Y -a $h -d $u')
```

При включении Procmail он считывает файл `~/.procmailrc`, содержащий набор правил, которые затем используются при обработке данного сообщения. Procmail -- это не демон, а инструмент обработки текста, принимающий электронные сообщения по одному через STDIN.

## Правила в `~/.procmailrc`

По сути файл настроек Procmail -- это просто набор правил, заданных в виде регулярных выражений. Вы можете определить в нем переменные среды в том же стиле, как это делается в скриптах командной оболочки. Правила выполняются строго в указанном порядке, но могут использоваться флаги для выполнения каких-то действий, только если предыдущее вернуло (A) или (E). Некоторые правила Procmail являются правилами доставки, а другие таковыми не являются; после выполнения подходящего правила обработка данного сообщения завершается, если только не указан флаг C для явного продолжения обработки. Вероятно, наиболее частым действием в списке правил является сохранение сообщений в именных почтовых ящиках, но вы можете также перенаправлять сообщения в другую программу или пересыпать сообщения целому списку адресатов.

Правило обычно начинается с указания блокировки (возможно с указанием конкретного файла блокировки, в его отсутствие файл выбирается автоматически) и нескольких флагов, затем следует некоторое количество условий и наконец ровно одна команда на выполнение. Например:

```
:0 [flags] [ : [locallockfile] ]
<zero or more conditions (one per line)>
<exactly one action line>
```

Отдельного упоминания заслуживают флаги H для соответствия заголовку B для соответствия телу. Шаблоны обычно не чувствительны к регистру, но флаг D позволяет включить чувствительность к регистру принудительно.

Если условие начинается с \*, то все после этого символа является регулярным выражением

**egrep**. В противном случае, если строка начинается с < или >, то проверяется размер сообщения больше оно или меньше указанного числа байт. Префикс \$ разрешает подстановки командной оболочки.

Команда, являющаяся просто именем файла, сохраняет сообщение в этом почтовом ящике. Используйте специальные псевдо-файлы `/dev/null` для удаления сообщения. Символ "вертикальная черта" ( | ) отправляет сообщение в другую программу, такую как почтовая утилита digest-splitting, распространяемая вместе с Procmail. Префикс в виде восклицательного знака (!), используемый внутри команды, пересыпает сообщение (внутри условия он играет роль отрицания). Несколько примеров:

### Листинг 6. Пример файла `~/.procmailrc`

```
:0:
* ^Subject:.*Digest          # split digests and save parts
* ^From:.*foo-digest
|formail +1 -ds cat >>mailing_lists_mailbox

:0:
* !(To|Cc).*mertz@gnosis.cx      # my main account here
* !(To|Cc).*david.mertz@gmail.com  # I get mail from here
* !From.*gnosis\.cx               # I trust gnosis not to spam
* !From.*list.*@                  # don't trash mailing lists
* !From.*good-buddy                # sometimes Bcc's me mail
spam

:0:
* ^Subject.*[MY-LIST]           # redistribute MY-LIST messages
! member@example.com, member2@example.net, member3@example.edu

:0:
* ^Cc.*joe@somewhere.org       # save to both inbox and JOE mbox
{
    :0 c
    $DEFAULT

    :0
    JOE
}
```

## Обслуживание NNTP новостей

### Что делает InterNetNews?

NNTP это прекрасный протокол для доставки сообщений по требованию любому пользователю, интересующемуся данной темой. Usenet -- это большая коллекция новостных групп по тысячам различных тем, которые доставляют сообщения через NNTP. Вследствие того, что протокол работает "по запросу", NNTP сервер собирает текущие сообщения доступные в децентрализованной сети серверов, выбирая только те группы новостей, которые администратор сайта выбрал для включения. При появлении нового сообщения в данной новостной группе оно распространяется без всякой иерархии с этого сервера на все остальные в сети Internet, заинтересованные в подписке на эту конкретную новостную группу.

С точки зрения конечного пользователя, список рассылки может показаться очень похожим на группу новостей. В обоих случаях пользователь создает и размещает сообщения, а также читает сообщения, написанные другими людьми. В давние времена Usenet и Internet, списки рассылки были не способны представить темы обсуждения в виде "цепочек", так как группы

новостей делают это сейчас автоматически. Но за прошедшие годы почтовые клиенты проделали хорошую работу по выделению цепочек обсуждения внутри списка рассылки.

Главным отличием новостных групп от списков рассылки заключается в используемых ими сетевых протоколах. Список рассылки все еще опирается на один централизованный почтовый сервер, который принимает все сообщения, предназначенные для отдельного списка, и распространяет эти сообщения по электронной почте всем пользователям, проявившим интерес (одобренные автоматически или человеком модератором). В противоположность этому, NNTP соединяет каждый узел со всеми остальными узлами не полагаясь на центральный сервер; каждый NNTP сервер просто общается с другими ближайшими серверами, и сообщения очень быстро облетают весь мир.

InterNetNews (INN) это NNTP сервер, впервые созданный в 1992, и активно использующийся до сих пор. С момента создания INN достиг уже версии 2.4.1. Домашняя страница INN содержит релизы и документацию (смотри ссылку в [Ресурсах](#)).

## Настройка INN

После получения и распаковки исходных текстов текущей версии, сборка INN осуществляется по шаблону `./configure; make; make install`. Для сборки INN, у вас должны быть установлены Perl и yacc (или bison). Программа установки создает множество файлов, преимущественно в каталоге `/usr/local/news/` (который вероятно у вас отсутствует, если ранее INN не был установлен).

Перед запуском демона `innd` (от имени пользователя "news"), вам следует изменить несколько конфигурационных файлов. Все подробности о полном наборе файлов, которым необходимо уделить внимание, находятся за рамками нашего обзора, в более длинном учебнике, озаглавленном *Установка и Запуск Сервера новостей Usenet с помощью INN и FreeBSD [Installing and Running a Usenet News Server with INN and FreeBSD]* доступном в сети (смотри ссылку в [Ресурсах](#)). Многие разрешения и квоты устанавливаются системой make, но вы скорее всего захотите дважды проверить эти настройки.

Файл, которому необходимо уделить особое внимание, это настройки квот -- `/usr/local/news/etc/storage.conf`. Он определяет на какие группы осуществляется подписка и насколько большая история будет поддерживаться для каждой группы новостей. Когда квота будет превышена, старые сообщения будут удалены из данной группы новостей (на локальном сервере, а не из Usenet вообще). Например, `storage.conf` может содержать нечто вроде этого:

### Листинг 7. Пример конфигурации `storage.conf`

```
method cnfs {
    newsgroups: alt.binaries./*
    class: 1
    size: 0,1000000
    options: BINARIES
}

method cnfs {
    newsgroups: *
    class: 2
    size: 0,100000
    options: NOTBINRY
}
```

Значение `class` определяет порядок в котором применяются различные правила.

После рассмотрения всех конфигурационных файлов просто запустите `innd` в качестве демона (возможен запуск из инициализационного скрипта), отслеживающего все вышележащие сервера, указанные в `/usr/local/news/etc/innfeed.conf`, `/usr/local/news/etc/incoming.conf` и `/usr/local/news/etc/newsfeeds`.

## Ресурсы

### Научиться

- Просмотрите всю [Серию учебников для подготовки к экзаменам LPI](#) на developerWorks чтобы освоить основы Linux и подготовиться к сертификации по системному администрированию.
- В [Программе LPIC](#), можно найти список заданий, примеры вопросов, и детальные требования для трех уровней сертификации по системному администрированию Linux от Linux Professional Institute [Профессиональный Институт Linux].
- Просмотрите [700 Linux User Groups \[Групп Пользователей Linux\] по всему миру](#) -- многие LUG имеют группы локального и дистанционного обучения для сдачи экзаменов LPI.
- Проект [Linux Documentation Project](#) содержит много разнообразной полезной документации, особенно HOWTO.
- [TCP/IP Network Administration, Third Edition \[Администрирование сетей TCP/IP. Третье издание\]](#) Крейга Ханта [Craig Hunt] (O'Reilly, April 2002) это прекрасная книга о сетях Linux.
- [Linux Network Administrators Guide \[Руководство по администрированию сетей в Linux\]](#) это большая сетевая книга охватывающая многие аспекты работы с сетями в Linux. О специфичных темах интересных для тестирования Sendmail пригодится раздел 18.9, [Testing Your Configuration \[Тестирование вашей конфигурации\]](#).
- [Сайт Sendmail](#) предоставляет [список книг о Sendmail](#) к которым можно обратиться, включая [Sendmail, Third Edition \[Sendmail, Третье издание\]](#) (O'Reilly, December 2002).
- Смотри [Installing and Running a Usenet News Server with INN and FreeBSD \[Установка и Запуск сервера Новостей при помощи INN и FreeBSD\]](#) для более полного учебного материала по установке и запуску INN сервера.
- Найдите больше [учебников для Linux разработчиков](#) в [Linux разделе developerWorks](#).

### Получить продукты и технологии

- На [Сайте Majordomo](#), вы можете найти файлы для загрузки и информацию о Majordomo.
- На сайте [procmail.org](#), можно загрузить последнюю версию Procmail.
- На сайте [sendmail.org](#), можно загрузить последнюю версию Sendmail.
- Посетите [домашнюю страницу INN](#) для получения информации InterNetNews и загрузки файлов.
- Создайте свой следующий Linux проект с [trial программным обеспечением IBM](#), доступным для загрузки напрямую с developerWorks.

# Экзамен LPI: Domain Name System (DNS, Доменная система имен)

*Администрирование Linux, средний уровень (LPIC-2) тема 207*

[Дэвид Мерц](#), автор, Gnosis Software, Inc.

**Описание:** Это третье из [семи руководств](#), описывающих базовое сетевое администрирование Linux®. Это руководство Дэвида Мерца представляет из себя введение в DNS и использование Linux как сервера DNS, прежде всего с использованием BIND 9. В нем рассказывается об установке и конфигурировании этого сервиса, обсуждается создание прямых и обратных зон, защита сервера от атак.

[Больше статей из этой серии](#)

**Дата:** 01.12.2005

**Уровень сложности:** средний

## Прежде чем начать

Узнайте, чему может научить вас это учебное пособие и как извлечь из него максимум.

## Об этой серии учебных пособий

[Linux Professional Institute](#) (LPI) осуществляет сертификацию системных администраторов Linux по двум уровням: *для начинающих* (также называемый "уровень сертификации 1") и *средний уровень* (также называемый "уровень сертификации 2"). Для достижения уровня сертификации 1 вы должны сдать экзамены 101 и 102; для достижения уровня сертификации 2 вы должны сдать экзамены 201 и 202.

developerWorks предоставляет учебные пособия, которые помогут вам в подготовке к каждому из четырех экзаменов. Каждый экзамен охватывает несколько тем, и для каждой темы на developerWorks существует соответствующее пособие для самостоятельного изучения. Экзамен LPI 202 содержит семь тем, которым соответствуют учебные пособия от developerWorks:

*Таблица 1. Экзамен LPI 202: Учебные пособия и темы*

Тема экзамена	Учебное пособие от LPI 202	Краткое содержание пособия
Тема 205	<a href="#">Материалы к экзамену LPI 202 (тема 205): Конфигурирование сети</a>	Описывает базовое конфигурирование сети TCP/IP, начиная с уровня оборудования (обычно это Ethernet, modem, ISDN, или 802.11) до настройки маршрутизации сетевых адресов.
Тема 206	<a href="#">Материалы к экзамену LPI 202 (тема 206): Почта и новости</a>	Описывает использование Linux-системы в качестве почтового сервера или сервера новостей. Описываются механизмы пересылки почты, локальная фильтрация почты, программное обеспечение для поддержки списков рассылки и серверное ПО для поддержки NNTP протокола.
Тема 207	Материалы к	(Это руководство) рассказывает об использовании

	экзамену LPI 202 (тема 207): DNS	Linux как DNS сервера, главным образом с использованием BIND. Описывается базовое конфигурирование BIND, поддержка зон DNS и защита DNS сервера. Детальнее смотрите <a href="#">цели</a> , изложенные ниже.
Тема 208	Материалы к экзамену LPI 202 (тема 208): Web сервисы	Скоро ожидается
Тема 210	Материалы к экзамену LPI 202 (тема 210): Поддержка сетевого клиента	Скоро ожидается
Тема 212	Материалы к экзамену LPI 202 (тема 212): Безопасность системы	Скоро ожидается
Тема 214	Материалы к экзамену LPI 202 (тема 214): Проблемы при работе с сетью	Скоро ожидается

Для продолжения подготовки к сертификации уровня 1 см. [учебные пособия от developerWorks для экзаменов LPI 101](#). Для подготовки к другим экзаменам для сертификации на 2-ой уровень, см. [учебные пособия от developerWorks для экзаменов LPI 201](#). Дополнительные информацию можно найти в [полном наборе учебных пособий LPI от developerWorks](#).

Linux Professional Institute не одобряет использование при подготовке к экзаменам любых учебных материалов или технологий, разработанных третьими лицами. За разъяснениями обращайтесь по адресу [info@lpi.org](mailto:info@lpi.org).

## Об этом руководстве

Добро пожаловать в "Domain Name System", третье из семи руководств, посвященных начальному сетевому администрированию Linux. В этом руководстве вы найдете систематическое изложение основ системы доменных имен DNS и научитесь использовать Linux как DNS сервер. Вы получите представление об установке и конфигурировании BIND-сервера, включая работу с named.conf и другими конфигурационными файлами; прямых и обратных зонах DNS, а так же об основах безопасности DNS, включая запуск BIND в среде chroot jail и DNS Security Extensions.

Данное руководство организовано в соответствии с рабочей программой LPI по этой теме. Грубо говоря, экзамены с большим количеством вопросов имеют больший рейтинг.

*Таблица 2. Domain Name System: Цели экзамена, описанные в этом учебном пособии*

Цель экзамена LPI	Рейтинг	Описание цели
2.207.1 <a href="#">Базовое</a>	2	Конфигурирование BIND как кэширующего DNS сервера. Этот раздел включает описание возможности

## конфигурирование BIND 8

конвертирования файла named.boot для BIND 4.9 в формат named.conf BIND 8.x и перегрузки DNS с использованием `kill` или `ndc`. Сюда также включается конфигурирование журналирования и опций, например, каталога, где располагаются файлы зон.

### 2.207.2 Создание и поддержка DNS зон

3

Создание файла зон для прямой и обратной зоны или сервера корневого уровня. Описываются возможные значения для записей ресурсов SOA, записей NS и MX. Излагается процедура добавления хостов через записи ресурсов A и записи CNAME, добавление хостов в обратные зоны в записи PTR, добавление зон в файл /etc/named.conf с использованием директивы `zone` с указанием типа, файла и мастера зоны. Вы должны также научиться делегированию зоны другому DNS серверу.

### 2.207.3 Защита DNS сервера

3

Конфигурирование BIND для запуска под непrivилегированным пользователем и в окружении chroot jail. Этот раздел также включает конфигурирование записей DNSSEC, таких как key и trusted-keys, для предотвращения атак путем подмены домена (domain spoofing). Кроме того, описывается возможность создания "split DNS" конфигурации при помощи директивы `forwarders` и указания нестандартного номера версии в ответе на DNS-запросы.

## **Необходимые условия**

Чтобы извлечь максимум из этого учебного пособия, вы должны иметь базовые знания о Linux и рабочую версию системы Linux, где вы сможете упражняться в выполнении команд, приведенных в этом пособии.

# **Экзамен LPI: Domain Name System (DNS, Доменная система имен)**

*Администрирование Linux, средний уровень (LPIC-2) тема 207*

Дэвид Мерц, автор, Gnosis Software, Inc.

**Описание:** Это третье из семи руководств, описывающих базовое сетевое администрирование Linux®. Это руководство Дэвида Мерца представляет из себя введение в DNS и использование Linux как сервера DNS, прежде всего с использованием BIND 9. В нем рассказывается об установке и конфигурировании этого сервиса, обсуждается создание прямых и обратных зон, защита сервера от атак.

## **DNS**

Доменная система имен (Domain Name System) позволяет достаточно удобным образом обращаться к серверам по имени, а не IP-адресу, всем пользователям TCP/IP приложений.

Berkeley Internet Name Domain (BIND) представляет из себя сервер *named*, отвечающий на запросы о связи IP-адреса с символическим именем (или наоборот, а также другую информацию). Со стороны клиента системы DNS имеется набор библиотек *resolver*, позволяющих приложениям связываться с DNS серверами. В пакет BIND также входит ряд клиентских приложений для конфигурирования, осуществления запросов и отладки BIND 9 сервера: `dig`, `nslookup`, `host`, and `rndc` (ранее `ndc`). В сущности, эти приложения вызывают те же самые библиотеки, что и другие клиентские приложения, непосредственно реагируя на запросы DNS серверов.

## BIND

На время написания данного руководства текущей версией BIND была 9.3.1. Первая стабильная версия серии BIND 9 была выпущена в октябре 2000. На некоторых старых инсталляциях вы можете найти BIND 8, который продолжает поддерживаться путем выпуска security patches (текущая версия 8.4.6), но, как правило, обновляются они до BIND 9 везде, где это возможно. Самые архаичные системы могут иметь BIND 4, но их рекомендуется обновлять как можно скорее, поскольку поддержка BIND 4 прекращена.

Все версии BIND могут быть получены от Internet Systems Consortium (ISC; см. ссылку в разделе [Ресурсы](#)). Документацию и другие ресурсы, связанные с BIND, также можно найти на этом сайте.

Поскольку основным для данной темы является знание конфигурации BIND 8 и мы далее сосредотачиваемся именно на BIND 8, мы рекомендуем ознакомиться с информацией по BIND 8 на сайте ISC, перед сдачей экзамена LPI 202.

## Другие ресурсы

Как и для большинства Linux-приложений, крайне полезным является изучение страниц руководств `man` по всем обсуждаемым утилитам. Версии и опции могут различаться от версии к версии утилиты или ядра, или между различными дистрибутивами Linux. Более глубокую информацию можно найти в большом количестве очень полезных HOWTO из Linux Documentation Project (см. ссылку в разделе [Ресурсы](#)). Был опубликован ряд полезных книг, в особенности хотелось бы отметить *TCP/IP Network Administration* издательства O'Reilly (постарайтесь найти последнее издание; см. ссылку в разделе [Ресурсы](#)).

Для получения информации именно о DNS и BIND очень хорошим источником может являться *DNS and BIND, Fourth Edition* издательства O'Reilly (см. ссылку в разделе [Ресурсы](#)); на ее 622 страницах можно найти более детальную информацию, чем в данном руководстве. Другие издательства также выпустили ряд изданий, посвященных BIND.

## Запросы доменной системы имен

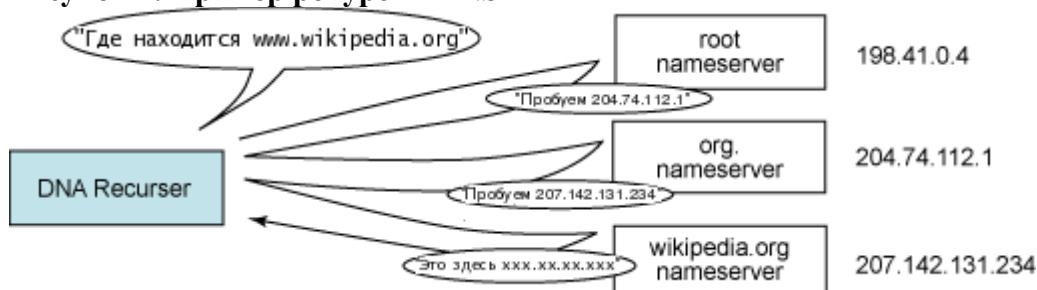
### Топология DNS

DNS представляет из себя иерархическую систему доменных зон. Каждая зона предоставляет ограниченный набор отображений в доменные имена, входящие в ее поддомен. Запрошенный сервер пошлет запрос более высокому в иерархии серверу в случае, если он не в состоянии найти требуемое отображение и, в случае необходимости, будет следовать указаниям о перенаправлении, пока не найдет правильного соответствия (или не определит, что данный запрос не может быть разрешен, выдав ошибку). Когда локальный сервер *named* получает ответ на посланный DNS запрос, он кэширует его на период времени, указанный в конфигурации (это скорее часы, чем секунды или дни). Вследствие кэширования DNS запросов совокупная сетевая загрузка значительно снижается, особенно для серверов верхнего уровня домена (top-level-domain -- TLD). Статья о DNS в Wikipedia (см. ссылку в разделе [Ресурсы](#)) -- отличная стартовая точка для понимания архитектуры в целом. В этом

руководстве мы приводим диаграмму из этого источника, предоставленную в свободный доступ (см. рис. 1 далее).

Диаграмма прохождения гипотетического DNS запроса позволит взглянуть на процесс его обслуживания в целом. Положим, ваша локальная машина хочет получить доступ к машине с доменным именем [www.wikipedia.org](http://www.wikipedia.org). Для поиска соответствующего IP адреса ваша машина должна обратиться к локальному серверу имен, указанному в конфигурации клиентской машины. Этот сервер может работать на той же самой машине, может располагаться на DNS сервере вашей локальной сети (LAN) или же предоставляться вашим интернет-провайдером (ISP). Во всех этих случаях это будет какой-то экземпляр BIND named. Этот локальный сервер имен в первую очередь проверит свой кэш и в случае, если там такой информации нет, выполнит следующие шаги, отображенные на диаграмме:

**Рисунок 1. Пример рекурсии DNS**



Следует понимать, что "DNS Recurser" -- это текущий DNS сервер (named), а не клиентское приложение, которое с ним общается.

DNS использует TCP и UDP на порту 53 для обслуживания запросов. Практически все DNS запросы представляют из себя одиночные UDP запросы от клиента, порождающие одиночные UDP ответы от сервера.

### Откуда приложение знает, где найти DNS сервер

Конфигурирование клиентского приложения для доступа к DNS серверу (серверам) достаточно просто. Вся конфигурация содержится в файле /etc/resolv.conf, в котором указаны один или более "локальных" DNS серверов. Вы можете вручную сконфигурировать /etc/resolv.conf на подключение к известным вам DNS серверам; однако, если вы используете DHCP для конфигурирования клиента, в ходе DHCP handshaking'a эта информация добавляется в /etc/resolv.conf автоматически (вы по-прежнему можете читать его и даже модифицировать установки, сделанные DHCP, но они будут вновь восстановлены после перезагрузки). Код библиотеки с установками, сделанными /etc/resolv.conf, называется "DNS resolver."

Если в /etc/resolv.conf указано более одного DNS сервера, вторичный и третичный DNS сервера будут использоваться в случае, если ответ от первичного сервера не поступает в течение указанного периода времени. Максимальное количество серверов, которое можно указать в конфигурации -- три.

Прежде всего, файл /etc/resolv.conf содержит записи следующего вида **nameserver <IP-addr>**. Некоторые другие записи могут изменять ответы на посланные запросы. Например, директивы **domain** и **search** расширяют имена, не содержащие точек (машины в локальной сети). Директива **options** позволяет вам изменять время ожидания ответа от DNS сервера, включать режим отладки при расширении до полного доменного имени и изменять другие аспекты работы DNS resolver'a. Например, на одной из моих машин конфигурация такова:

## **Листинг 1. Модификация опций в файле конфигурации доступа к DNS серверам**

```
# cat /etc/resolv.conf
search gnosis.lan
nameserver 0.0.0.0
nameserver 192.168.2.1
nameserver 151.203.0.84
options timeout:3
```

Первая директива указывает, что машины в локальной сети входят во внутренний домен *gnosis.lan*, таким образом, короткое имя *bacchus* может быть расширено в *bacchus.gnosis.lan*. Несколько доменов, разделенных пробелами могут быть перечислены в директиве **search**.

Затем я перечислил несколько DNS серверов. Первый из них -- локальная машина, на которую можно ссылаться как на *0.0.0.0* или через официальный IP адрес, но не *loopback*-адрес. Следующая директива, **nameserver**, указывает на мой домашний роутер, соединяющий мою локальную сеть с Интернет (и поддерживает DHCP и DNS серверы). Третий **nameserver** предоставляется мне сервис-провайдером. Кроме того, я установил 3-х секундное время ожидания (**timeout**) для каждого сервера имен, вместо 5 секунд, установленных по умолчанию.

## **Клиентские утилиты DNS**

BIND 9 поставляется с четырьмя основными клиентскими утилитами. Три из них -- **dig**, **nslookup** и **host** -- выполняют сходные функции, более или менее отличающиеся деталями. Все три утилиты представляют из себя приложения, выполняемые в командной строке и посылающие запросы в DNS resolver. В сущности они делают то, что другие клиентские приложения делают на внутреннем уровне, но выводя результаты своей деятельности на STDOUT. Наиболее мощным средством из описанных выше является **dig**, поскольку предоставляет наиболее широкий набор опций для задания запросов и конфигурирования формата вывода полученной информации.

Эти утилиты чаще всего используются для определения IP адреса из символьного доменного имени, но вы также можете сделать и обратный запрос или запросить другие типы записей, кроме записей типа "A". Например, команда **host -t MX gnosis.cx** покажет вам почтовые сервера домена gnosis.cx. Несколько полезных примеров:

## **Листинг 2. Запрос при помощи host о google.com**

```
$ host google.com
google.com has address 72.14.207.99
google.com has address 64.233.187.99
```

## **Листинг 3. Запрос при помощи host о записи MX для gnosis.cx**

```
$ host -t MX gnosis.cx
gnosis.cx mail is handled by 10 mail.gnosis.cx.
```

Для утилиты **nslookup**:

## **Листинг 4. nslookup использует сервер, установленный по умолчанию (на локальной**

**машине)**

```
$ nslookup gnosis.cx
Server:      0.0.0.0
Address:     0.0.0.0#53

Non-authoritative answer:
Name:   gnosis.cx
Address: 64.41.64.172
```

Обратный запрос с использованием утилиты **dig** и с указанием другого сервера имен:

**Листинг 5. Обратный запрос через dig на иной сервер, чем установленный по умолчанию**

```
$ dig @192.168.2.2 -x 64.233.187.99

; <>> DiG 9.2.4 <>> @192.168.2.2 -x 64.233.187.99
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 3950
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;99.187.233.64.in-addr.arpa.    IN  PTR

;; AUTHORITY SECTION:
187.233.64.in-addr.arpa. 2613  IN  SOA  ns1.google.com. dns-admin.google.com.
2004041601 21600 3600 1038800 86400

;; Query time: 1 msec
;; SERVER: 192.168.2.2#53(192.168.2.2)
;; WHEN: Thu Nov 10 02:00:27 2005
;; MSG SIZE  rcvd: 104
```

Последняя утилита BIND 9, о которой надо иметь представление, -- это **rndc**, утилита для управления сервером имен. Она является расширением утилиты **ndc**, поставлявшейся с предыдущими версиями BIND. Если **rndc** вызывается из командной строки без опций и аргументов, она выводит короткую справку о поддерживаемых командах. См. страницы системного руководства **man** для получения полной информации об использовании **rndc**.

**Задание базовой конфигурации BIND и запуск сервера имен**

**Конфигурирование BIND**

Запуск демона **named** в качестве DNS сервера возможен в одном из трех режимов: *master*, *slave* и *caching-only*. Демон **named** управляет自身的 конфигурационными файлами, в первую очередь **/etc/bind/named.conf**, для определения режима запуска.

В *master* mode сервер **named** работает как авторитетный источник всей информации об этой зоне. Информация о домене, предоставляемая сервером, размещается в файле на локальном диске, который можно модифицировать или обновить вручную. Каждая зона DNS должна иметь единственный мастер-сервер.

В *slave* mode сервер **named** передает информацию, предоставленную мастер-сервером зоны.

Технически, сервер, поддерживающий несколько зон, может играть роль master'а для одной зоны и slave для другой, но обычно в локальной сети присутствует единственная иерархия из master, slave и caching-only серверов. Сервер slave переносит полную информацию о зоне с мастер-сервера в локальные файлы, так что информация, предоставляемая slave сервером, по-прежнему остается авторитетной.

В режиме caching-only сервер named не поддерживает файлы зон. Каждый запрос перенаправляется на какой-то другой сервер имен для получения первичной информации, но затем полученная информация кешируется. Однако новые запросы требуют передачи запроса далее по сети. Caching-only сервера чаще всего используются на локальных машинах, где клиентские приложения часто посыпают запросы в службу имен без генерации при этом сетевого трафика.

В конфигурации /etc/resolv.conf, показанной мной ранее в [Листинге 1](#), 0.0.0.0 -- это caching-only сервер, 192.168.2.1 -- slave сервер и 151.203.0.84 -- как master сервер. Вы не можете определить этого, просто глядя на порядок их следования или на их IP адреса, но использование псевдо-IP адреса локальной машины дает основание полагать, что это caching-only сервер.

### Создание named.conf

Есть несколько стандартных моментов, которые должны быть включены в каждый файл /etc/bind/named.conf. Это начальная директива **options**, задающая некоторую базовую информацию. Затем несколько директив **zone**, задающих конфигурацию для обработки различных зонных запросов. Домены, поименованные в директивах **Zone** так, что их название начинается с IP адресов, задающих начальные цифры диапазона IP адресов, обозначают "обратные" зоны. Символьные имена определяют зоны, а кроме того, позволяют дальнейшее определение поддоменов.

Файлы named.conf (и другие конфигурационные файлы BIND) следуют соглашениям по форматированию языка C для больших фрагментов текста. Могут использоваться как C-подобные блочные комментарии (`/* comment */`) и принятые в C++ строчные (`// comment`), так и строчные комментарии shell (`# comment`). Директивы завершаются точками с запятой и заключаются в фигурные скобки.

Для начала несколько обычных опций. Мой локальный файл /etc/bind/named.conf начинается с:

### Листинг 6. Мой локальный named.conf начинается так

```
include "/etc/bind/named.conf.options";
```

Но вы можете также вставлять непосредственно директиву **options**:

### Листинг 7. Задание опций в named.conf

```
options {
    directory "/var/bind";
    forwarders { 192.168.2.1; 192.168.3.1};
    // forward only;
}
```

Эти установки указывают на то, что файлы, указанные без полного пути, будут искааться в указанном каталоге; кроме того, BIND в первую очередь обращается к 192.168.2.1 и

192.168.3.1 для незакэшированных результатов. Директива `forward only` (здесь закомментированная) говорит о том, что запросы выполняются на эти сервера имен, а не запрашиваются корневые сервера в Интернете.

Специальная директива `zone` должна помещаться ранее других в файлах named.conf:

### Листинг 8. Особая зона (Hint zone) корневых серверов

```
zone "." {
    type hint;
    file "/etc/bind/db.root";
};
```

Содержимое `db.root` (иногда называемого `named.ca` для "certifying authority") носит достаточно специальный характер. Оно представляет из себя описание набора канонических корневых серверов, собственно регистрирующих домены. Этот файл меняется достаточно редко, но вы всегда можете получить его последнюю официальную версию с `ftp.rs.internic.net`. Это не тот файл, который будет править обычный администратор.

За корневой особой зоной в named.conf должны размещаться master и/или slave зоны. Например, для локального loopback'a:

### Листинг 9. Конфигурация loopback зоны

```
zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};
```

Что более интересно, named может выступать как master домена (в том числе предоставляя и обратные записи):

### Листинг 10. Конфигурация внешней зоны

```
zone "example.com" {
    type master;
    file "example.com.hosts"; // file relative to /var/bind
};
// Reverse lookup for 64.41.* IP addresses (backward IP address)
zone "41.64.in-addr.arpa" {
    type master;
    file "41.64.rev";
};
```

Для slave конфигурации вместо этого вы можете использовать:

### Листинг 11. Конфигурация внешней зоны (slave)

```
zone "example.com" {
    type slave;
    file "example.com.hosts"; // file relative to /var/bind
```

```

        masters { 192.168.2.1; };
};

// Reverse lookup for 64.41.* IP addresses (backward IP address)
zone "41.64.in-addr.arpa" {
    type slave;
    file "41.64.rev";
    masters { 192.168.2.1; };
};

```

## Другие конфигурационные файлы

Файл named.conf ссылается на ряд других конфигурационных файлов через директиву `file`. Они могут различаться в зависимости от конкретной установки, но обычно они будут содержать какие-то записи, определенные в RFC 1033 (*Domain Administrators Operations Guide*; см. в разделе [Ресурсы](#)). Стандартные записи:

### SOA

Start of authority (Начало полномочий). Параметры, действующие на всю зону.

### NS

Nameserver (Сервер имен). Доменный сервер имен.

### A

Address (Адрес). Имя хоста или IP адрес.

### PTR

Pointer (Указатель). IP адрес или имя хоста.

### MX

Mail exchange (Почтовая станция). Где обрабатывается почта домена.

### CNAME

Canonical name (Каноническое имя). Псевдоним хоста.

### TXT

Text (Текст). Хранит произвольные значения.

Формат записи следующий: `<name> <time-to-live> IN <type> <data>`.

Имя и "время жизни" (time-to-live) опциональны, по умолчанию используются последние значения. Символьная строка `IN` обозначает Internet и частенько используется в практике. Файлы с записями ресурсов могут также содержать директивы, начинающиеся со знака доллара. Наверное, наиболее часто в `$TTL`, устанавливающем умолчание для времени жизни. Например, какой нибудь тривиальный файл записей для `127.* localhost` выглядит так:

## Листинг 12. Простейший файл записей

```

# cat /etc/bind/db.127
; BIND reverse data file for local loopback interface
;
$TTL    604800
@      IN      SOA     localhost. root.localhost. (
                      1          ; Serial
                      604800    ; Refresh
                      86400     ; Retry
                      2419200   ; Expire
                      604800 )  ; Negative Cache TTL
;
@      IN      NS      localhost.

```

```
1.0.0 IN PTR localhost.
```

Другие директивы -- это **\$ORIGIN**, устанавливающая имя домена, используемое для дописывания любого относительного имени домена; **\$INCLUDE**, которая подгружает внешний файл; и **\$GENERATE**, создающая серию записей, лежащих в определенном диапазоне IP адресов.

## Создание и поддержание DNS зон

### Файлы обратных зон

Файлы обратных зон (часто имеющих расширение .rev) -- содержащие отображение IP адресов данной зоны на их символьные имена. Например, у вас может быть файл /var/bind/41.64.rev, который содержит:

#### Листинг 13. Файл обратной зоны для 64.41.\*

```
$TTL 86400
; IP address to hostname
@ IN SOA example.com. mail.example.com. (
    2001061401 ; Serial
    21600       ; Refresh
    1800        ; Retry
    604800      ; Expire
    900 )       ; Negative cach TTL

        IN NS ns1.example.com.
        IN NS ns2.example.com.
; Define names for 64.41.2.1, 64.41.2.2, etc.
1.2     IN PTR foo.example.com.
2.2     IN PTR bar.example.com.
3.2     IN PTR baz.example.com.
```

### Файлы прямых зон

Файлы прямых зон (обычно называемые как *domain.hosts*) -- содержащие основные записи "A" отображающие символьные имена в IP адреса. Например, у вас может быть файл /var/bind/example.com.hosts, содержащий следующее:

#### Листинг 14. Прямая зона для example.com

```
$TTL 86400
; Hostname to IP address
@ IN SOA example.com. mail.example.com. (
    2001061401 ; Serial
    21600       ; Refresh
    1800        ; Retry
    604800      ; Expire
    900 )       ; Negative cach TTL

        IN NS ns1.example.com.
        IN NS ns2.example.com.
localhost IN A 127.0.0.1
foo      IN A 64.41.2.1
```

```
www           IN      CNAME  foo.example.com
bar           IN      A       64.41.2.2
bar           IN      A       64.41.2.3
```

## Защита DNS сервера

Как и для многих других сервисов, запуск BIND в так называемом *chroot jail* окружении является хорошей идеей. Это ограничивает доступ BIND к другим файлам или системным ресурсам, при взломе BIND или наличии в нем ошибок. Более детальную информацию о запуске BIND в chroot окружении можно найти в "Chroot-BIND HOWTO" (см. в разделе [Ресурсы](#)).

Суть этой процедуры заключается в том, что запуск BIND под суперпользователем root или даже под традиционным спец. пользователем типа "nobody". Обычно для запуска BIND создается пользователь "named". Файлы, используемые этим спец. пользователем, помещаются в локальный каталог, например /chroot/named/, и соответствующие подкаталоги.

BIND 9 предоставляет более четкую поддержку для chroot, чем BIND 8; для ее включения достаточно обычной сборки без специальных опций или установок в Makefile.

## DNSSEC

Кроме проведения работ по общему повышению защищенности машины, на которой работает BIND, также возможно обеспечить гарантированную защиту в рамках самого протокола DNS. DNS Security Extensions (DNSSEC) представляет из себя набор расширений DNS, обеспечивающих аутентификацию и целостность.

DNS базируется на UDP в большей степени, чем на TCP, а стало быть, не имеет механизма верификации источника пакета. Другими словами, посылающие запрос к DNS могут получить в ответ ложные данные, например перенаправляющие соединение на хост взломщика. Добавлением криптографических Transactional Signatures (TSIG) в DNS запросы DNSSEC может предупредить подмену (spoofing) DNS ответов. На каждом сервере BIND 9, который должен работать в защищенном режиме, должен быть включен DNSSEC.

Расширенный протокол, с другой стороны, обладает обратной совместимостью. В первую очередь DNS сервера, которые желают обмениваться данными в защищенном режиме, должны каким-то образом сгенерировать *пару ключей*. Это работает так же, как и SSH ключи для хоста и сервера. Например:

## Листинг 15. Генерация ключей DNSSEC

```
dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 128 -n HOST \
primary-secondary.my.dom
# ls Kprimary-secondary.my.dom.*
Kprimary-secondary.my.dom.+157+46713.key
Kprimary-secondary.my.dom.+157+46713.private
```

Как подсказывают нам имена файлов, генерируются открытый (public) и закрытый (private) ключи для конфигурируемого хоста и публичный ключ для распространения по другим серверам. Хорошим введением в DNSSEC является "The Basics of DNSSEC" из O'Reilly Network (см. в разделе [Ресурсы](#)).

## Ресурсы

### Научиться

- [Оригинал этого учебного пособия](#) на developerWorks.

- Обзор всей [серии учебных пособий для экзамена LPI](#) на developerWorks для изучения основ операционной системы Linux и подготовки к сертификации по системному администрированию.
- В [Программе LPIC](#) вы найдете список заданий, типовые вопросы и подробные программы для трех уровней сертификации Linux Professional Institute по системному администрированию Linux.
- The [Статья о DNS в Wikipedia](#) -- отличная стартовая точка для понимания архитектуры в целом.
- Книга Пола Альбица (Paul Albitz) и Крикета Лю (Cricket Liu) [\*DNS and BIND, Fourth Edition\*](#) (O'Reilly, 2001), посвященная DNS and BIND.
- [\*TCP/IP Network Administration, Third Edition\*](#) Грэга Ханта (Craig Hunt) (O'Reilly, апрель 2002) -- отличная книга о сетевом администрировании Linux.
- Обратитесь к RFC 1033, the [\*Domain Administrators Operations Guide\*](#) для того, чтобы ознакомиться с определениями стандартных записей ресурсов.
- В [Chroot-BIND HOWTO](#) показано, как настроить BIND и запустить его в среде chroot jail.
- В [Basics of DNSSEC](#) дается хорошее введение в DNSSEC.
- Посмотрите на [700 Linux User Groups around the world](#) -- многие LUG проводят локальное и дистанционное обучение по подготовке к экзаменам LPI.
- На [Linux Documentation Project](#) содержится ряд полезных документов, главным образом HOWTO.
- Найдите другие [учебные пособия для Linux-разработчиков](#) на [developerWorks Linux zone](#).

## Получить продукты и технологии

- [Загрузите BIND](#) с [Internet Systems Consortium](#).
- Постройте ваш следующий проект разработки для Linux с использованием [IBM trial software](#), загрузив его непосредственно с developerWorks.

# Подготовка к экзамену LPI: Web-сервисы

Средний уровень администрирования (LPIC-2) тема 208

[Дэвид Мертц \(David Mertz\)](#), Разработчик, Gnosis Software, Inc.

**Описание:** В этом учебном пособии, четвертом из [серии, состоящей из семи пособий](#), охватывающих сетевое администрирование под Linux, Дэвид Мертц продолжает готовить вас к сдаче экзамена 208 Администрирования Среднего Уровня (LPIC-2) Профессионального Института Linux. На этот раз Дэвид Мертц рассказывает, как настраивать и запускать Apache HTTP сервер и Squid прокси-сервер.

[Больше статей из этой серии](#)

**Дата:** 25.04.2006

**Уровень сложности:** средний

## Перед тем, как начать

Узнайте, чему могут научить эти пособия, и как получить максимум от них.

## Об этих курсах

[Профессиональный Институт Linux](#) аттестовывает системных администраторов Linux по двум уровням: *младший уровень* (или "аттестация уровня 1") и *средний уровень* (или "аттестация уровня 2"). Для получения аттестация уровня 1, нужно сдать экзамены 101 и 102; для получения аттестация уровня 2, следует сдать экзамены 201 и 202.

developerWorks предоставляет учебные пособия, чтобы помочь вам подготовиться ко всем четырем экзаменам. Каждый экзамен охватывает несколько тем, и каждая тема имеет соответствующее учебное пособия на developerWorks. Для экзамена LPI 202, семь тем и соответствующие developerWorks пособия таковы:

*Таблица 1. Экзамен LPI 202: Темы и учебные пособия*

Тема экзамена LPI 202	Учебные пособие developerWorks	Краткое содержание
Тема 205	<a href="#">Подготовка к LPI экзамену 202 (тема 205): Конфигурация сети</a>	Узнайте, как настроить базовую сеть TCP/IP, от аппаратного уровня (как правило, это Ethernet, или, ISDN, или 802.11) до маршрутизации сетевых адресов.
Тема 206	<a href="#">Подготовка к LPI экзамену 202 (тема 206): Почта и новости</a>	Узнайте, как использовать Linux в качестве почтового сервера и новостного сервера. Узнайте о почтовом транспорте, фильтре локальной почты, фильтре локальной почты, программах по управлению списком рассылки и серверных приложениях для протокола NNTP.
Тема 207	<a href="#">Подготовка к LPI экзамену 202 (тема 207): DNS</a>	Узнайте, как использовать Linux в качестве DNS-сервера, главным образом, с использованием BIND. Узнайте, как осуществить базовую настройку BIND, управлять зонами DNS, и обеспечивать безопасность DNS-сервера.
Тема 208	Подготовка к LPI	(Это учебное пособие) Узнайте, как установить и

	экзамену 202 (тема 208): Web-сервисы	настроить web-сервер Apache, и как использовать прокси-сервер Squid. Подробности ищите <a href="#">ниже</a> .
Тема 210	Подготовка к LPI экзамену 202 (тема 2010): Управление сетевым клиентом	Скоро будет
Тема 212	Подготовка к LPI экзамену 202 (тема 212): Системная безопасность	Скоро будет
Тема 214	Подготовка к LPI экзамену 202 (тема 214): Устранение неполадок работы сети	Скоро будет

Чтобы начать готовиться к аттестации уровня 1, посмотрите [учебные пособия developerWorks для экзамена LPI 101](#). Для подготовки к другим экзаменам аттестации уровня 2, посмотрите [учебные пособия developerWorks для экзамена LPI 201](#). Read больше о [полном курсе LPI](#) [учебных пособий от developerWorks](#).

Профessionальный институт Linux не поддерживает никаких материалов по подготовке к экзаменам, разработанных третьими лицами. За подробностями обращайтесь на [info@lpi.org](mailto:info@lpi.org).

## Об этом пособии

Добро пожаловать на "Web-сервисы", четвертый из семи учебных пособий, посвященных среднему уровню администрирования Linux. Здесь вы узнаете, как настраивать и запускать HTTP сервер Apache и прокси-сервер (Web Proxy Cache) Squid.

Наряду с другими учебными пособиями из developerWorks курсов 201 и 202, представленный материал предназначен скорее служить руководством к изучению и стартовой точкой при подготовке к экзаменам, и не является полной документацией по данному предмету.

Поощряется, если читатель обращается к [подробному списку целей и задач](#) LPI и, по необходимости, дополняет информацию, представленную здесь, другими материалами.

Это учебное пособие проранжировано согласно LPI задачам для этой темы. Грубо говоря, ожидайте на экзамене больше вопросов по темам с большим весом.

*Таблица 2. Web-службы: Задачи экзамена, рассмотренные в этом пособии*

Тема	Вес	Краткое содержание
2.208.1 <a href="#">Внедрение web-сервера</a>	2	Установка и настройка web-сервера. Это цель включает наблюдение за загрузкой и производительностью сервера, ограничение доступа пользователям клиента, настройка поддержки скриптовых языков посредством модулей, и установка идентификации пользователя клиента. Сюда также включается способность настроить параметры сервера так, чтобы ограничить использование ресурсов.
2.208.2	2	Настройка web-сервера на использование виртуальных хостов,

[Поддержка](#)  
[работы](#)  
[web-сервера](#)

2.208.3

[Внедрение](#)  
[прокси-сервера](#)

[В начало](#)

Secure Sockets Layer (SSL), и настройка доступа к файлам.

2 Установка и настройка прокси-сервера, включая политику доступа, идентификацию, и использование ресурсов.

## **Предпосылки**

Для того, чтобы извлечь из этого учебника максимум, вы должны уже иметь основные знания о Linux и рабочую систему Linux, где вы можете опробовать команды, описанные в этом пособии.

# **Подготовка к экзамену LPI: Web-сервисы**

*Средний уровень администрирования (LPIC-2) тема 208*

[Дэвид Мертц \(David Mertz\)](#), Разработчик, Gnosis Software, Inc.

**Описание:** В этом учебном пособии, четвертом из [серии, состоящей из семи пособий](#), охватывающих сетевое администрирование под Linux, Дэвид Мертц продолжает готовить вас к сдаче экзамена 208 Администрирования Среднего Уровня (LPIC-2) Профессионального Института Linux. На этот раз Дэвид Мертц рассказывает, как настраивать и запускать Apache HTTP сервер и Squid прокси-сервер.

**Дата:** 25.04.2006

**Уровень сложности:** средний

## **Об Apache и Squid**

### **Web-сервер Apache**

Apache -- это самый распространенный web-сервер во всем Интернете, и среди серверов на основе Linux его преимущество выглядит подавляющим. Кроме него существуют и другие специализированные web-сервера (некоторые из них показывают более высокую производительность для определенных задач), но именно Apache устанавливается по умолчанию.

В большинстве дистрибутивов Linux Apache предустановлен и часто уже работает, будучи запущенным во время установки, даже если вы и не настраивали его. Если Apache не установлен, его можно установить, используя обычную систему установки вашего дистрибутива, или же вы можете [скачать последнюю версию](#) с Apache HTTP Server Project. Большое количество дополнительных возможностей может быть обеспечено модулями, многие из которых поставляются вместе с Apache, а остальные доступны от третьих лиц.

Хотя с 2001 года последней версией Apache является ветка 2.x, Apache 1.3.x все еще широко используется, и ветка 1.3.x продолжает поддерживаться в виде bug fixes и обновлений безопасности. Между 1.3 и 2.x существуют незначительные различия в конфигурации; некоторые модули доступны для 1.3, и не доступны для 2.x. Последние версии на момент написания этого пособия -- 1.3.34 (стабильная), 2.0.55 (стабильная), и 2.1.9 (бета).

Как правило, новый сервер использует последнюю стабильную версию семейства 2.x. Если у вас нет нужды использовать именно более старый модуль, 2.x дает хорошую стабильность, больше возможностей, и совокупную лучшую производительность (в некоторых задачах, таких как поддержка PHP в 1.3 все еще работает лучше). Заглядывая в будущее, учтите, что

новые возможности будут, безусловно, лучше поддерживаться версией 2.x, а не 1.3.x.

### Squid прокси-сервер

Squid -- это прокси-кэширующий сервер для web-клиентов, который поддерживает протоколы HTTP, FTP, TLS, SSL, и HTTPS. Скорость передачи может быть увеличена, а ширина канала сети -- снижена благодаря работе прокси-сервера в локальной сети, или, по меньшей мере, где-нибудь ближе к вашей сети, чем запрашиваемые ресурсы. Когда один и тот же ресурс запрашивается машинами, обслуживамыми одним и тем же сервером Squid несколько раз, этот ресурс доставляется из локальной копии на сервере, и не запросу не требуется проходить через множество сетевых маршрутизаторов и потенциально замедлять или нагружать сервера назначения.

Можно настроить Squid в качестве явного прокси, которого следует настраивать для каждого web-клиента (браузера), или же можно сделать так, чтобы он перехватывал все web-запросы, выходящие за пределы LAN и кэшировал весь такой трафик. Можно также указать Squid при помощи его многочисленных опций, как долго и при каких условиях следует хранить web-страницы в кэше.

### другие источники

Как и при изучении других приложений Linux, всегда полезно обращаться к man-страницам любых рассматриваемых здесь утилит. Версии и опции могут отличаться для различных версий утилиты или ядра, или же у различных дистрибутивов Linux. За дополнительной информацией обращайтесь на Linux Documentation Project, он содержит большое количество различных полезных документов, в особенности HOWTO. Также опубликовано множество книг по сетям Linux; я считаю, что книга O'Reilly's *TCP/IP Network Administration*, Крейга Ханта, вполне может помочь. (Ссылки [Источники](#) располагаются ниже.)

По работе с Apache тоже написано много хороших книг. Некоторые касаются общих вопросов администрирования, тогда как другие охватывают конкретные модули или специальные настройки Apache. Посетите ваш любимый книжный магазин и поищите там книги с подходящими названиями.

## Внедрение web-сервера

### Туча демонов

Запуск Apache похож на запуск любого другого демона. Обычно хочется поместить запуск в скрипт инициализации системы, однако, в принципе, можно запускать Apache в любое время. В большей части систем сервер Apache называется *httpd*, хотя он может вместо этого называться и *apache2*. Вероятнее всего, сервер установлен в */usr/sbin/*, но возможны и другие расположения, в зависимости от вашего дистрибутива и от того, как вы установили сервер.

Чаще всего Apache запускается без опций, хотя об опциях **-d serverroot** и **-f config** следует помнить. Первая позволяет указывать расположение локального каталога, откуда поставляется содержимое; вторая позволяет указывать конфигурационный файл, отличный от используемого по умолчанию. Файл конфигурации может отменять опцию **-f** с помощью директивы **ServerRoot**. По умолчанию, конфигурационными файлами являются либо *apache2.conf*, либо *httpd.conf*, в зависимости от установок при компиляции. Эти файлы, скорее всего, располагаются в */etc/apache2/*, */etc/apache/*, */etc/httpd/conf/*, */etc/httpd/apache.conf*, или в каких-нибудь других местах, в зависимости от версии, дистрибутива Linux, и от того, как вы установили и скомпилировали Apache. Вызов **man apache2** или **man httpd** должен выдать вам системно-зависимые подробности.

Демон Apache отличается от других серверов тем, что он обычно создает несколько выполняющихся копий самого себя. Первичная копия просто порождает остальных, в то время как эти вторичные копии и обслуживают входящие запросы. Целью наличия

множества запущенных копий является создание набора обработчиков на случай нескольких одновременных запросов к серверу; при необходимости могут запускаться дополнительные копии демона в соответствии с параметрами конфигурации. Первичная копия обычно запускается от root, но остальные копии по соображениям безопасности запускаются как более ограниченный пользователь. Например:

### **Listing 1. The многоликость выполняющихся копий Apache**

```
# ps axu | grep apache2
root      6620      Ss    Nov12    0:00 /usr/sbin/apache2 -k start -DSSL
www-data  6621      S     Nov12    0:00 /usr/sbin/apache2 -k start -DSSL
www-data  6622      Sl    Nov12    0:00 /usr/sbin/apache2 -k start -DSSL
www-data  6624      Sl    Nov12    0:00 /usr/sbin/apache2 -k start -DSSL
dqm       313      S+    03:44   0:00 man apache2
root      637      S+    03:59   0:00 grep apache2
```

На большом числе систем ограниченным пользователем является *nobody*. В Listing 1 это -- пользователь **www-data**.

### **Включение конфигурационных файлов**

Как уже упоминалось, поведение Apache определяется директивами в его конфигурационном файле. Для систем Apache2, главный конфигурационный файл, скорее всего, находится в /etc/apache2/apache2.conf, но часто этот файл содержит многочисленные **Include** statements для добавления информации о конфигурации из других файлов, возможно, даже с шаблонами. В общем случае, конфигурация Apache, вероятно, содержит сотни директив и опций (большая часть которых не описывается в этом пособии).

В частности, несколько файлов, вероятно должны быть включены. Можно взглянуть в настройки "пользователей" файла httpd.conf, для использования прежних Apache 1.3 файлов конфигурации, использующих то же имя. Виртуальные хосты обычно задаются в отдельных конфигурационных файлах, соответствующих шаблону, например, вот так:

### **Listing 2. Задание виртуальных хостов**

```
# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/[^.#]*
```

При использовании Apache 2.x, модули обычно тоже определяются в отдельных конфигурационных файлах (более часто в том же файле в 1.3.x). Например, в моей системе включения таковы:

### **Listing 3. From /etc/apache2/apache2.conf**

```
# Include module configuration:
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
```

Фактически использование модуля в запущенном сервере Apache требует два шага в файле конфигурации, оба загружающих и активирующих его:

#### **Listing 4. Загрузка дополнительного модуля Apache**

```
# cat /etc/apache2/mods-enabled/userdir.load
LoadModule userdir_module /usr/lib/apache2/modules/mod_userdir.so
# cat /etc/apache2/mods-enabled/userdir.conf
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    </Directory>
</IfModule>
```

Шаблоны в строках **Include** вставлят все файлы .load и .conf в каталоге /etc/apache2/mods-enabled/

Обратите внимание на общую мысль: Основные директивы -- это команды в одну строку с некоторыми параметрами; более сложные директивы используют XML-льный тег открыть/закрыть для вложенных команд. Является ли директива одностroчной или же стиль открыть/закрыть, следует знать -- по своему усмотрению стили выбирать нельзя.

#### **Файлы журнала**

Важный класс директив конфигурации касается журналирования действий Apache. Можно задавать различные типы информации и степени детализации для операций Apache. Ведение журнала ошибок всегда приветствуется; это можно задать одной директивой:

#### **Listing 5. Задание журнала ошибок**

```
# Global error log.
ErrorLog /var/log/apache2/error.log
```

Можно добавить другие журналы для записи обращений к серверу, ссылающего сайта, и другой информации, удовлетворяющей вашим индивидуальным целям. Операция журналирования настраивается двумя директивами. Сначала директива **LogFormat** использует набор специальных переменных для задания, того, что помещать в файл журнала; затем, директива **CustomLog** говорит Apache actually записывать события в указанном формате. Можно задать бесчисленное число форматов невзирая на то, используются ли они на самом деле. Это позволяет включать и выключать журналирование подробностей, в зависимости от меняющихся потребностей.

Переменные в **LogFormat** похожи на переменные оболочки, но имеют в начале %. Некоторые переменные состоят из одной буквы, в то время как другие имеют длинные имена, окруженные скобками, как показано в Listing 6.

#### **Listing 6. Переменные LogFormat**

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
CustomLog /var/log/apache2/referer_log combined
```

Обратитесь к книге с полной документацией Apache за списком всех переменных. Широко используются следующие: `%h` для IP-адреса клиента, выполняющего запрос, `%t` для даты и времени запроса, `%>S` для HTTP статуса кода, и написанный с ошибкой `%{Referer}` для адреса ссылающего сайта, на котором есть ссылка на обрабатываемую страницу.

Имена, используемые в директивах `LogFormat` и `CustomLog` являются произвольными. В Listing 6 использовалось имя `combined`, но вместо нее могла стоять, например, `myfoobarlog`. Однако несколько имен являются общепринятыми и даются в образцах файлах конфигурации, такие как `combined`, `common`, `referer`, и `agent`. Эти специальные форматы зачастую поддерживаются утилитами, анализирующими log-файлы.

## Поддержка работы web-сервера

### Виртуальные хосты, multi-homing и отдельные настройки различных каталогов

Индивидуальные каталоги, обслуживаемые сервером Apache, могут иметь свои собственные настройки конфигурации. Однако в главном конфигурационном файле может указать ограничения на то, какие опции могут быть настроены локально. Если настройка каждого каталога в отдельности разрешается, то используется директива `AccessFileName`, и обычно прописывается локальное конфигурационное имя файла `.htaccess`. Ограничения на возможности локальной настройки каталога определяются в директиве `<Directory>`. Например:

### Listing 7. Пример директивы `directory`

```
#Let's have some Icons, shall we?  
Alias /icons/ "/usr/share/apache2/icons/"  
<Directory "/usr/share/apache2/icons">  
    Options Indexes MultiViews  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>
```

Часто одновременно с индивидуально настраиваемыми каталогами Apache может обслуживать *виртуальные хосты*. Многочисленные доменные имена могут быть обслужены одним и тем же процессом Apache, каждый имея доступ к определенному каталогу. Виртуальные хосты можно определить директивой `<VirtualHost>`; разместите файлы конфигурации во внутреннем каталоге, например, `/etc/apache2/sites-enabled/`, или в главном конфигурационном файле. Например, можно их задать вот так:

### Listing 8. Конфигурация виртуальных хостов

```
<VirtualHost "foo.example.com">  
    ServerAdmin webmaster@foo.example.com  
    DocumentRoot /var/www/foo  
    ServerName foo.example.com  
    <Directory /var/www/foo>  
        Options Indexes FollowSymLinks MultiViews  
        AllowOverride None  
        Order allow,deny  
        allow from all  
    </Directory>  
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  
    <Directory "/usr/lib/cgi-bin">  
        AllowOverride None
```

```

        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
    CustomLog /var/log/apache2/foo_access.log combined
</VirtualHost>
<VirtualHost "bar.example.org">
    DocumentRoot /var/www/bar
    ServerName bar.example.org
</VirtualHost>
<VirtualHost *>
    DocumentRoot /var/www
</VirtualHost>

```

Последняя опция \* принимает все HTTP запросы, которые направлены не на один из явно определенных имен (как те, адресуемые IP-адресом или адресуемые неуказанным символьным доменом, что также приводят к машине-серверу). Чтобы виртуальные хосты работали, DNS должна определять каждый псевдоним с записью CNAME.

Возможность multi-homing по названию похожа на виртуальный хостинг, однако идея здесь совсем другая. Используя *multi-homing*, можно указывать IP-адреса, с которыми машина соединяется для того, чтобы разрешить web-запросы. Например, можно дать HTTP-доступ только локальной сети LAN, но не для остального мира. Если указывается адрес, от которого ждут сигнала, также можно указать неумолчательный порт. Значение для **BindAddress**, выставленное по умолчанию -- это \*, что означает принимать запросы на все IP-адреса, с которыми связан этот сервер. Пример смешанного использования выглядит примерно вот так:

### **Listing 9. Конфигурация multi-homing**

```

BindAddress 192.168.2.2
Listen 192.168.2.2:8000
Listen 64.41.64.172:8080

```

В этом случае, все клиентские запросы из локальной LAN (использующие адрес 192.168.2.2) будут приняты на порт 80 и специальный порт 8000. Этот дистрибутив Apache также будет отслеживать клиентские HTTP запросы из WAN адреса, но только на порт 8080.

### **Ограничение доступа к Сети**

Командами **Order**, **Allow from**, и **Deny from** в директиве **<Directory>** можно управлять доступом к отдельным каталогам сервера. Запрещенные или разрешенные адреса можно задавать полными или частичными именами хостов или IP-адресами. **Order** позволяет задавать приоритет между списком разрешения и списком запрета.

Часто требуется более тонкий контроль, нежели тот, что задается простым разрешением определенным хостам обращаться к web-серверу. Для подключения требований входа пользователя, используется семейство команд **Auth\***, опять-таки, внутри директивы **<Directory>**. Например, для установки базовой аутентификации можно использовать директиву, как показано в Listing 10.

### **Listing 10. Конфигурация базовой аутентификации**

```

<Directory "/var/www/baz">
    AuthName "Baz"

```

```
AuthType Basic
AuthUserFile /etc/apache2/http.passwrods
AuthGroupFile /etc/apache2/http.groups
Require john jill sally bob
</Directory>
```

Можно также указать базовую аутентификацию внутри .htaccess-файла. Аутентификация дайджестом более безопасна, нежели базовая, но не так широко реализована в браузерах. Однако слабость базовой схемы (когда пароль передается открытым текстом) в любом случае лучше решается с помощью уровня SSL.

Поддержка SSL-шифрования web-трафика обеспечивается модулем `mod_ssl`. Если используется SSL, данные, передаваемые между сервером и клиентом, шифруются с динамически изменяемым паролем, что является стойким по отношению к перехвату. Все основные браузеры поддерживают SSL. За большей информацией по настройке Apache 2.x с `mod_ssl`, обратитесь к описанию на web-сайте Apache (ссылка указана в [ресурсах](#)).

## Внедрение прокси-сервера

### Установка и запуск Squid

В большей части дистрибутивов, Squid может быть установлен при помощи стандартных установочных процедур. Возьмите исходные файлы Squid с web-сайта Squid Web Proxy Cache. Сборка из исходников требует обычной последовательности действий  
`./configure; make; make install`.

После установки можно просто запустить от имени root `/usr/sbin/squid` (но `squid` может оказаться в каком-нибудь другом каталоге, который использует ваш дистрибутив, возможно `/usr/local/sbin/`). Конечно, чтобы заставить его делать что-то полезное, вам придётся отредактировать файл конфигурации Squid `/etc/squid/squid.conf`, `/usr/local/squid/etc/squid.conf`, или где там у вас в системе находится `squid.conf`. Как и для многих других демонов, можно использовать различные файлы конфигураций, с помощью опции `-f`.

### Порты, IP-адресы, http\_access, и ACLи

Наиболее важными опциями конфигурации Squid являются `http_port`. Если захотеть, то можно наблюдать за этими портами, закрепляя за каждым из них определенный IP-адрес или имя хоста. Порт для Squid, выставленный по умолчанию, 3128, при этом все IP-адреса, что подключаются к серверу Squid, имеют права на его использование. Чтобы кэшировать только для LAN, укажите вместо этого локальный IP-адрес, как показано ниже:

#### Listing 11. Кэширование Squid только для LAN

```
# default (disabled)
# http_port 3128
# LAN only
http_port 192.168.2.2:3128
```

Также можно включить кэширование через другие Squid-серверы, с помощью `icp_port` и `http_port`. Протоколы IPC и HTTP используются, чтобы кэши взаимодействовали друг с другом, а не с web-серверами или клиентами. Для группового кэширования используется `mcast_groups`.

Чтобы клиенты могли подключиться к серверу Squid, нужно раздать им соответствующие разрешения. В отличие от web-сервера, Squid не сильно щепетилен со своими ресурсами. В

простой ситуации для контроля за правами доступа можно просто использовать пару подсеть/маска сети или шаблоны CIDR (Classless Internet Domain Routing):

### Listing 12. Простые права доступа Squid

```
http_access deny 10.0.1.0/255.255.255.0
http_access allow 10.0.0.0/8
icp_access allow 10.0.0.0/8
```

Директива `acl` используется для обозначения списков контроля доступа (ACLs). Можно назвать `src` ACL, что просто указывают диапазоны адресов, как в Listing 12, но можно также создавать и свои типы ACLей. Например:

### Listing 13. Тонкая настройка права доступа

```
acl mynetwork    src          192.168/16
acl asp          urlpath_regex \.asp$
acl bad_ports    port         70 873
acl javascript   rep_mime_type -i ^application/x-javascript$
# what HTTP access to allow classes
http_access deny asp      # don't cache active server pages
http_access deny bad_ports # don't cache gopher or rsync
http_access deny javascript # don't cache javascript content
http_access allow mynetwork # allow LAN everything not denied
```

Listing 13 показывает только малый набор доступных типов ACL. Образец файла `squid.conf` содержит много других примеров. Или же взгляните в документацию по контролю доступа (Глава 6) в Руководстве Пользователя Squid (ссылку можно увидеть в [Resources](#)).

В Listing 13, принято решение не кэшировать URL, заканчивающиеся на `.asp` (возможно, они имеют динамическое содержимое), на кэшировать порты 70 и 873, и не кэшировать возвращаемые объекты JavaScript. И, если это не запрещено другими правилами, машины внутри LAN (в диапазоне /16) будут пользоваться кэшированием своих запросов. Обратите внимание, что каждый заданный ACL имеет уникальное, хотя и произвольное, имя (используйте осмысленные имена; Squid их не резервирует).

### Режимы кэширования

Простейший способ запуска Squid -- это режим прокси. Если это так, клиентов нужно явно настраивать на использование кэша. Клиенты web-браузеры имеют специальные окна настройки, позволяющие указывать адрес прокси и порт, вместо прямого HTTP-соединения. При таком способе настройка Squid очень проста, однако клиентам, если они хотят использовать выгоду от кэша Squid, следует выполнить немного настроек самим.

Также можно настроить Squid на выполнение в качестве прозрачного кэша. Для этого нужно либо настроить маршрутизацию, основанную на политике безопасности (извне Squid, с помощью `ipchains` или `ipfilter`), либо использовать сервер Squid в качестве шлюза. Полагая, что вы можете, направлять внешние запросы через сервер Squid, Squid следует настраивать следующим образом. Может случиться, что вам придется перекомпилировать Squid с опцией `--enable-ipf-transparent`; однако для большинства дистрибутивов Linux этого не потребуется. Для настройки сервера на прозрачное кэширование (по мере того, как он получит перенаправленные пакеты), нужно добавить в файл `squid.conf` что-нибудь вроде представленного в Listing 14:

#### **Listing 14. Настройка Squid на прозрачное кэширование**

```
httpd_accel_host virtual  
httpd_accel_port 80  
httpd_accel_with_proxy on  
httpd_accel_uses_host_header on
```

#### **Ресурсы**

#### **Научиться**

- Ознакомьтесь с полным [учебным курсом подготовки к экзамену LPI](#) на developerWorks, дабы изучить основы Linux и подготовиться к аттестации системного администратора.
- Вам так же доступна [оригинальная версия](#) этого учебника (на английском языке).
- На [LPIC Program](#), найдите список задач, примерных вопросов и подробных целей для трех уровней аттестации системных администраторов Linux Профессионального Института Linux.
- Узнайте больше о [настройке Apache 2.x при помощи mod\\_ssl](#).
- Прочтайте книгу "[Customizing Apache for maximum performance](#)" (developerWorks, June 2002) и узнайте, как подогнать Apache под определенные среды и нужды.
- Глава [Access Control and Access Control Operators](#) в Руководстве Пользователя Squid описывает доступные типы ACL.
- [TCP/IP Network Administration, Третье издание](#), Крейг Хант (O'Reilly, April 2002) является отличным ресурсом по сетям Linux.
- Домашняя страница [Linux Users Groups WorldWide](#) дает список 700 пользовательских групп Linux со всего света. Множество LUGов имеют локальные и дистанционные группы обучения для экзаменов LPI.
- [Linux Documentation Project](#) располагает множеством различных полезных документов, в особенности HOWTO.
- На [developerWorks Linux-зоне](#), найдите больше ресурсов для разработчиков Linux.
- Оставайтесь в курсе [технических событий developerWorks и Webcasts](#).

#### **Получить продукты и технологии**

- Скачайте последний [web-сервер Apache](#).
- Скачайте [Squid](#) и дополнительную документацию по Squid.
- Вместе с [тестовое ПО IBM](#), доступном для скачивания непосредственно с developerWorks, постройте ваш следующий проект на Linux.

# Подготовка к экзамену LPI: Управление клиентом сети

Средний уровень администрирования (LPIC-2) тема 210

[Дэвид \(David\) Мертц \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** В этом учебном пособии, пятом из [серии, состоящей из семи пособий](#), охватывающих сетевое администрирование под Linux, Дэвид Мертц продолжает готовить вас к сдаче экзамена 202 Администрирования Среднего Уровня (LPIC-2) Профессионального Института Linux. Следуя этому пособию, вы изучите централизованные конфигурации настроек сети клиентов для нескольких протоколов. DHCP широко используется для установления основного подтверждения связи между клиентскими машинами, такими как присвоение IP-адресов. На более высоком уровне, для обмена любой информацией между машинами сети используется NIS и (более часто) LDAP. В этом пособии также обсуждается PAM, который является гибкой сетевой системой идентификации пользователей.

[Больше статей из этой серии](#)

**Дата:** 24.05.2006

**Уровень сложности:** средний

## Прежде чем начать

Узнайте, чему могут научить эти пособия, и как получить максимум от них.

## Об этих курсах

[Профессиональный Институт Linux](#) аттестовывает системных администраторов Linux по двум уровням: *младший уровень* (или "аттестация уровня 1") и *средний уровень* (или "аттестация уровня 2"). Для получения аттестация уровня 1, нужно сдать экзамены 101 и 102; для получения аттестация уровня 2, следует сдать экзамены 201 и 202.

developerWorks предоставляет учебные пособия, чтобы помочь вам подготовиться ко всем четырем экзаменам. Каждый экзамен охватывает несколько тем, и каждая тема имеет соответствующее учебное пособие на developerWorks. Для экзамена LPI 202, семь тем и соответствующие developerWorks пособия таковы:

Таблица 1. Экзамен LPI 202: Темы и учебные пособия

Тема экзамена	Учебные пособие developerWorks	Краткое содержание
LPI 202		
Тема 205	<a href="#">LPI exam 202 prep (topic 205): Конфигурация сети</a>	Узнайте, как настроить базовую сеть TCP/IP, от аппаратного уровня (как правило, это Ethernet, или, ISDN, или 802.11) до маршрутизации сетевых адресов.
Тема 206	<a href="#">LPI exam 202 prep (topic 206): Почта и новости</a>	Узнайте, как использовать Linux в качестве почтового сервера и новостного сервера. Узнайте о почтовом транспорте, фильтре локальной почты, фильтре локальной почты, программах по управлению списком рассылки и серверных приложениях для протокола NNTP.

Тема 207	<a href="#">Подготовка к LPI экзамену 202 (тема 207): DNS</a>	Узнайте, как использовать Linux в качестве DNS-сервера, главным образом, с использованием BIND. Узнайте, как осуществить базовую настройку BIND, управлять зонами DNS, и обеспечивать безопасность DNS-сервера.
Тема 208	<a href="#">Подготовка к LPI экзамену 202 (тема 208): Web-сервисы</a>	Узнайте, как установить и настроить web-сервер Apache, и как использовать прокси-сервер Squid.
Тема 210	Подготовка к LPI экзамену 202 (тема 201): Управление клиентом сети	(Это пособие) Узнайте, как настроить сервер DHCP, клиента NIS и NISD сервер, сервер LDAP, и поддержку идентификации PAM. На подробностисмотрите <a href="#">ниже</a> .
Тема 212	Подготовка к LPI экзамену 202 (тема 212): Системная безопасность	Скоро будет
Тема 214	Подготовка к LPI экзамену 202 (тема 214): Устранение неполадок работы сети	Скоро будет

Чтобы начать готовиться к аттестации уровня 1, посмотрите [учебные пособия developerWorks для экзамена LPI 101](#). Для подготовки к другим экзаменам аттестации уровня 2, посмотрите [учебные пособия developerWorks для экзамена LPI 201](#). Здесь вы можете прочитать больше о [полном курсе LPI учебных пособий от developerWorks](#).

Профессиональный институт Linux не поддерживает никаких материалов по подготовке к экзаменам, разработанных третьими лицами. За подробностями обращайтесь на [info@lpi.org](mailto:info@lpi.org).

## Об этом пособии

Добро пожаловать на "Управление клиентом сети", пятое из семи учебных пособий, посвященных промежуточному сетевому администрированию под Linux. Здесь вы узнаете, о многопротокольной централизованной конфигурации настроек сети на клиентах внутри сети, увидите, как широко применяют DHCP для установления основного подтверждения связи между клиентскими машинами (такими, как присвоение IP-адресов), и увидите как, на более высоком уровне, для обмена любой информацией между машинами сети используются NIS и (более часто) LDAP для произвольного разделения информации между машинами в сети. В этом пособии также обсуждается PAM (Pluggable Authentication Module), который является гибкой сетевой системой идентификации пользователей.

Наряду с другими учебными пособиями из developerWorks курсов 201 и 202, представленный материал предназначен скорее служить руководством к изучению и стартовой точкой при подготовке к экзаменам, и не является полной документацией по данному предмету. Поощряется, если читатель обращается к [подробному списку целей и задач](#) LPI и, по необходимости, дополняет информацию, представленную здесь, другими материалами.

Это учебное пособие проранжировано согласно LPI задачам для этой темы. Грубо говоря,

ожидайте на экзамене больше вопросов по темам с большим весом.

*Таблица 2. Веб-службы: Задачи экзамена, рассмотренные в этом пособии*

Тема	Вес	Краткое содержание
2.210.1 <a href="#"><u>Конфигурация DHCP</u></a>	2	Настройте сервер DHCP. Это включает установку опций по умолчанию и опций на клиенте, добавление статических хостов и хостов БООТР. Также сюда входит настройка агента-ретранслятора DHCP и поддержка DHCP-сервера.
2.210.2 <a href="#"><u>Конфигурация NIS</u></a>	1	Настройте сервер NIS. Здесь рассматривается настройка системы как клиента NIS.
2.210.3 <a href="#"><u>Конфигурация LDAP</u></a>	1	Настройте сервер LDAP. Это включает работу с иерархией каталога, группами, хостами, службами, и добавление новых данных в иерархию. Также сюда входит импорт и добавление пунктов, равно как и добавление пользователей и управление ими.
2.210.4 <a href="#"><u>Идентификация PAM</u></a>	2	Настройте PAM на поддержку идентификации, используя различные доступные методы.

## Предпосылки

Для полной отдачи от этого пособия, вы должны уже иметь основные знания о Linux и рабочую систему Linux, где вы можете опробовать команды, описанные в этом пособии.

# Подготовка к экзамену LPI: Управление клиентом сети

*Средний уровень администрирования (LPIC-2) тема 210*

[Дэвид \(David\) Мертц \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** В этом учебном пособии, пятом из [серии, состоящей из семи пособий](#), охватывающих сетевое администрирование под Linux, Дэвид Мертц продолжает готовить вас к сдаче экзамена 202 Администрирования Среднего Уровня (LPIC-2) Профессионального Института Linux. Следуя этому пособию, вы изучите централизованные конфигурации настроек сети клиентов для нескольких протоколов. DHCP широко используется для установления основного подтверждения связи между клиентскими машинами, такими как присвоение IP-адресов. На более высоком уровне, для обмена любой информацией между машинами сети используется NIS и (более часто) LDAP. В этом пособии также обсуждается PAM, который является гибкой сетевой системой идентификации пользователей.

**Дата:** 24.05.2006

**Уровень сложности:** средний

## Введение

### DHCP

Dynamic Host Configuration Protocol (DHCP) -- это преемник более старого протокола БООТР. Главная роль DHCP-сервера состоит в присвоении клиентским машинам, которые могут

соединяться или рассоединяться с сетью, IP-адресов. Большинство IP сетей, даже при наличии стабильной топологии и клиентских списков, используют DHCP, чтобы исключить конфликты при распределении IP-адресов.

Кроме этого, DHCP-сервер обеспечивает клиентов информацией о маршрутизации и о подсети, адресами DNS, и иногда и другой информацией. DHCP присваивания могут иметь различную длительность, как малую, так и бесконечно большую, в зависимости от конфигурации сервера и деталей запроса клиента. На самом деле DHCP совместим и с присваиванием фиксированных IP-адресов некоторым машинам (посредством их аппаратных MAC адресов), но в любом случае конфликты между машинами предотвращаются.

Формальной спецификацией DHCP является RFC 2131

## **O NIS**

Network Information Service (NIS) -- это протокол клиент-серверной службы каталога от Sun Microsystems' "Yellow Pages" (YP) для распространения данных системных конфигураций, таких как пользователи и имена хостов в компьютерной сети.

NIS/YP использует центральный каталог для хранения пользователей, имен хостов, и многих других полезных в компьютерной сети вещей. Например, в большинстве сред UNIX, список пользователей (для идентификации) размещается в /etc/passwd. При использовании NIS добавляется другой "глобальный" список пользователей, который используется для идентификации пользователей на каждой машине.

В большинстве своем NIS вытесняется более общей и более безопасной LDAP, пригодной для более широкого использования.

Хорошим началом для добывания большей информации о NIS -- это "The Linux NIS(YP)/NYS/NIS+ HOWTO"

## **O LDAP**

Lightweight Directory Access Protocol (LDAP) -- клиент-серверный протокол для доступа к службам каталога, в обобщенности основанных на X.500.

Каталог LDAP похож на базу данных, но склоняется к содержанию более подробной, атрибутивной информации. Поэтому LDAP обеспечивает достаточную гибкость для хранения любого вида информации, разделяемое по сети. Информация в каталоге считывается намного чаще, нежели записывается туда, так что она настроена на быстрый отклик при операции поиска или при запросах большого объема данных.

LDAP имеет способность широко реплицировать информацию, чтобы повысить ее доступность и надежность, при этом уменьшая время доступа. При репликации информации из каталога всякие временные несоответствия между репликами будут синхронизоваться.

Формальной спецификацией LDAP является RFC 2251.

## **O PAM**

Linux-PAM (Pluggable Authentication Modules for Linux) -- это комплект разделяемых библиотек, которые позволяют локальному системному администратору выбирать, как приложения идентифицируют пользователей.

Использующее PAM приложение может в процессе выполнения переключаться между различными механизмами идентификации. Кроме того, можно полностью обновить локальную систему идентификации, не перекомпилируя приложения. Эта библиотека PAM сконфигурирована локально при помощи системного файла, /etc/pam.conf (или совокупностью конфигурационных файлов, расположенных в /etc/pam.d/) для идентификации запроса пользователя через локально доступные модули идентификации. Модули сами по себе обычно располагаются в каталоге /lib/security и имеют вид динамически

загружаемых объектных файлов.

Руководство Linux-PAM System Administrators -- хорошее начало для получения дополнительной информации

## Другие ресурсы

Как и другие приложения Linux, всегда полезно обращаться к man-страницам любых рассматриваемых здесь утилит. Linux Documentation Project содержит большое количество различных полезных документов, в особенности HOWTO. Также опубликовано множество книг по сетям Linux; я считаю, что книга O'Reilly's *TCP/IP Network Administration*, Крейга Ханта, вполне может помочь.

## Конфигурация DHCP

### Общее о протоколе

Подобно большинству сетевых протоколов, Dynamic Host Configuration Protocol (DHCP) -- это клиент/серверный интерфейс. Клиент DHCP является намного более простой программой, как по устройству, так и для настройки, нежели сервер DHCP. В сущности, вся работа клиента DHCP состоит в рассылке сообщения DHCPDISCOVER по локальной физической подсети, и ожидания ответа.

Сообщение DHCPDISCOVER может содержать параметры, которые содержат желаемое значение для сетевого адреса и продолжительности аренды. Если серверы получает сообщение DHCPDISCOVER, то он должен ответить соответствующему MAC адресу сообщением DHCPOFFER. Клиент, в свою очередь, отвечает сообщением DHCPREQUEST одному из предлагающих серверов, обычно первому (и единственному) серверу, приславшему ответ.

Главные параметры конфигурации, которые использует клиент, он получает в сообщении DHCPACK. При этом клиент получает собственный IP-адрес и с этого момента может взаимодействовать не только на уровне Data Link Layer (Ethernet), но и на уровне Network Layer (IP) как полноценный участник IP-сети.

### Процесс на клиенте

Клиент DHCP обычно требует только настройки информации, которую он желает получать. Например, дистрибутив на основе Debian обычно использует клиента DHCP, dhclient, который сконфигурирован файлом /etc/dhcp3/dhclient.conf. Образец файла, распространяемые с пакетом dhcp3-client, имеет все опции, кроме одной, закомментированными. Единственная задействованная опция, скорее всего, имеет вид:

### Listing 1. Опции для dhclient.conf

```
request subnet-mask, broadcast-address, time-offset, routers,
      domain-name, domain-name-servers, host-name,
      netbios-name-servers, netbios-scope;
```

В этом примере, конфигурации по умолчанию, клиент, по существу, говорит: "спрашивайте все, что возможно". В переговорных сообщениях, сообщение DHCPACK от сервера будет содержать информацию для всех таких запрошенных значений, которые клиент сможет использовать по мере получения. IP-адрес клиента содержится в списке, поскольку эта конфигурация всегда есть предмет переговоров.

Наряду с указанием максимального времени и параметрами времени аренды (а также некоторыми другими), клиент *может*, но, в большинстве случаев, не обязан, наложить

некоторые ограничения на IP-адрес, который он хочет использовать. Для того, чтобы исключить определенный адрес, можно записать `reject 192.33.137.209;`. Для указания явного адреса, который хочет использовать клиент, следует записать `fixed-address 192.5.5.213;`.

Клиент может отклонить предложение аренды сообщением DHCPDECLINE, но сервера попытаются удовлетворить запросы везде, где это возможно. DHCP-сервер может также сделать фиксированное присваивание определенного IP-адреса к запрашиваемому MAC-адресу; настройка IP-адресов по машинам чаще производится конфигурацией сервера, чем конфигурацией клиента.

Чтобы отслеживать полученные аренды, dhclient хранит список аренд, которые были присвоены в файле `/var/lib/dhcp3/dhclient.leases` (путь может отличаться в различных дистрибутивах); таким образом, не просроченная DHCP аренда не теряется, даже если система отсоединяется от физической сети и/или перезагружается.

## Процесс на сервере

Сервер DHCP должен знать немного больше своих опций, поскольку он обеспечивает клиентов различной информацией во время DHCP аренды, и также должен быть уверен, что клиенты получили уникальные IP-адреса. Сервер DHCРобычно запускается как демон, dhcpd, и черпает информацию о своей конфигурации из `/etc/dhcpd.conf` (этот путь может варьироваться, в зависимости от дистрибутива). Один демон dhcpd может, тем не менее, поддерживать несколько подсетей, обычно когда к серверу подсоединяются многие физические сети; однако зачастую все-таки один сервер управляет одной подсетью. Listing 2 -- фактически полный пример конфигурации сервера.

### Listing 2. Конфигурационные опции dhcpd.conf

```
# default/max lease in seconds: day/week
default-lease-time 86400;
max-lease-time 604800;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
# DNS locally, fallback to outside local domain
option domain-name-servers 192.168.2.1, 151.203.0.84;
option domain-name "example.com";
# Set the subnet in which IP address are pooled
subnet 192.168.2.0 netmask 255.255.255.0 {
    range 192.168.2.100 192.168.2.199;
}
# Assign a fixed IP to a couple machines by MAC address
group {
    use-host-decl-names true;
    host first.example.com {
        hardware ethernet 00:00:c1:a2:de:10;
        fixed-address 192.168.2.200;
    }
    host second.example.com {
        hardware ethernet 00:dd:cc:a1:78:34;
        fixed-address 192.168.2.201;
    }
}
```

Когда клиент высылает широковещательное сообщение серверу с такой конфигурацией, он либо получит в аренду 192.168.2.200, или 192.168.2.201, если он имеет указанный MAC-адрес, либо получит в аренду доступный адрес в диапазоне с 192.168.2.100 по

192.168.2.199.

Клиент также может использовать сообщение DHCPINFORM, чтобы сообщить серверу, что он уже имеет назначенный (вручную) IP-адрес, но хотел бы получить другую конфигурационную информацию. Обратите внимание, что сообщать серверу о том, что клиент *использует* определенный IP-адрес -- совсем не то же, что запрашивание определенного IP-адреса; в последнем случае сервер может дать согласие на запрос, а может и не дать, в зависимости от существующих арендах. А в первом случае сервер не имеет голоса в принятии решения, и, по сути, аренда не предоставляется (однако сервер попытается избежать присваивания новым клиентам тех IP-адресов, которые используются).

Когда истекает срок действия аренды, клиенты и сервера должны договориться о новых арендах, а параметры конфигурации остаются в силе. Более короткие аренды могут использоваться там, где конфигурационная информация на сервере вероятно изменяется (например, на динамическом DNS посредством внешнего WAN). Клиент может элегантно прекратить аренду путем послания сообщения DHCPRELEASE, но для корректной работы не требуется и этого (клиенты иногда сбоят, перезагружаются, или, в конце концов, разъединяются и не могут послать такое сообщение).

При отсутствии высвобождающего сообщения аренда поддерживается сервером, пока истекут условия, на которых она была получена, так что перегруженная машина зачастую продолжает использовать свою предшествующую аренду (которая хранится в dhclient.leases как на сервере, так и на клиенте).

## Конфигурация NIS

### Когда надо использовать NIS

Большая часть утилит, связанных с NIS, все еще имеют префикс *yp*, из-за того, что она была первоначально известна как "Sun Yellow Pages"; проблемы с использованием торговой марки вынудили сменить имя на NIS. NIS используется, когда нужно совместно использовать информацию, такую как пользователи и группы (содержимым /etc/passwd и /etc/group, соответственно), внутри сети, предлагая пользователям права доступа на любой машине внутри домена NIS.

NIS действует способом, аналогичным тому, как DNS определяет домены, где распределена информация, и разрешает master и slave серверам иерархически распределять информацию внутри домена. Фактически, NIS можно использовать вместо DNS, путем распределения информации о доменном имени, обнаруженной в /etc/hosts, но на практике так делают редко. NIS присуща определенная гибкость, так что любой тип информации может быть, в принципе, помещен в базу данных NIS (которая имеет формат DBM, и хотя утилита makedbm из пакета NIS сервера преобразовывает простые файлы в этот формат, как правило, "заценой").

Также существует служба, которая называется NIS+, предназначенная заменить NIS, включая шифрование данных и идентификацию; однако NIS+ не имеет обратной совместимости с NIS и не является широко используемым.

### Перед тем, как начать

Для запуска любой из утилит NIS, необходимо запустить демона /sbin/portmap, который переводит номера программ RPC в номера портов протокола TCP/IP (или UDP/IP), поскольку клиенты NIS делают вызова RPC. Большинство дистрибутивов Linux запускают /sbin/portmap в своих скриптах инициализации системы, однако следует проверить, что этот демон запущен, с помощью % ps -ax | grep portmap.

Если он еще не выполняется, установите /sbin/portmap и внесите его в скрипты запуска вашей системы.

## Утилиты клиента NIS (демон `ypbind`)

Клиент NIS содержит утилиты `ypbind`, `ypwhich`, `ypcat`, `yppoll` и `ypmatch`. Демон `ypbind` должен быть запущен от root, и обычно он запускается одним из скриптов запуска системы (хотя это и не обязательно).

Другие утилиты зависят от служб `ypbind`, но выполняются на пользовательском уровне. Старые версии `ypbind` рассылают по локальной сети запрос на соединение, но это позволяет злонамеренному NIS-серверу отвечать на запрос и давать клиентам неправильную информацию о пользователях и группах. Предпочтительнее с помощью файла `/etc/yp.conf` настраивать определенные серверы, к которым может подключиться `ypbind`. Когда настроено несколько серверов (или когда, несмотря на опасность, используется рассылка), `ypbind` может переключаться между серверами каждые 15 минут, в зависимости от того, кто из них откликается быстрее всего. Эти разные сервера должны быть связаны друг с другом как `master/slave`, клиенту не нужно ни знать ни заботиться об этом. Например, конфигурация `ypbind` может иметь такой вид:

### Listing 3. `/etc/yp.conf`

```
ypserver 192.168.2.1
ypserver 192.168.2.2
ypserver 192.168.1.100
ypserver 192.168.2.200
```

Перед запуском `/usr/sbin/ypbind`, следует задать вашей сети доменное имя NIS. Оно может быть любым именем, на использование которого настроен NIS, и, вообще говоря, должно отличаться от доменного имени. Это доменное имя NIS задается примерно так: `%  
ypdomainname my.nis.domain`.

## Утилиты для клиента NIS (другие конфигурации)

Если вы желаете использовать NIS для поиска доменного имени, следует изменить `/etc/host.conf` так, чтобы включить NIS в порядок поиска; например, чтобы проверить имя сначала в `/etc/hosts`, затем в NIS, а затем в DNS:

### Listing 4. Изменение порядка `lookup`

```
% cat /etc/host.conf
order hosts,nis,bind
```

Для подключения NIS-распределенных пользователей, следует изменить клиентский файл `/etc/passwd`, добавив туда `+:::>:::`.

Информация базы данных NIS ведет себя как шаблон для попыток входа с такой "незаполненной" содержимым. При желании можно настроить информацию о пользователе, например, так:

### Listing 5. Подробный `/etc/passwd`

```
+user1:::::
+user2:::::
+user3:::::
+@sysadmins::::::
```

```
-ftp  
+*::::::/etc/NoShell
```

Это дает разрешение на вход для `user1`, `user2` и `user3`, а также всем членам сетевой группы `sysadmin`, и обеспечивает хранение учетных записей всех пользователей в базе данных NIS.

Источники NIS настраиваются в `/etc/nsswitch.conf`. Имя этого файла предполагает, что он предназначен строго для сервера имен, однако там описываются разные типы данных. В основном эта конфигурация описывает порядок, в котором происходит поиск источников информации. Имя `nis` в этой последовательности означает, что информацию получают с NIS-сервера; имя `files` значит, что нужно использовать соответствующий локальный файл конфигурации. Имя `dns` используется для опции `hosts`.

Кроме того, можно указать, что делать, если начальные источники не содержат требуемой информации: `return` означает отказ (а если NIS еще доступен, `continue` значит, что нужно попробовать обратиться за данными к следующему источнику). Например:

#### **Listing 6. /etc/nsswitch.conf**

```
passwd:      compat  
group:       compat  
shadow:      compat  
hosts:       dns  [!UNAVAIL=return] files  
networks:    nis  [NOTFOUND=return] files  
ethers:      nis  [NOTFOUND=continue] files  
protocols:   nis  [NOTFOUND=return] files  
rpc:         nis  [NOTFOUND=return] files  
services:    nis  [NOTFOUND=return] files
```

### **Утилиты пользователя NIS-клиента**

Утилиты `ypwhich`, `ypcat`, `ypoll`, и `ypmatch` используются на пользовательском уровне для запросов информации NIS:

- `ypwhich` выдает имя NIS-сервера.
- `ypcat` выдает значения всех ключей базы данных NIS.
- `ypoll` выдает версию и `master server` карты NIS.
- `ypmatch` выдает значения одного или более ключей из карты NIS.

За более подробной информацией по каждой утилите обратитесь к соответствующим `man`-страницам.

### **Утилиты сервера NIS (`ypinit`, `ypserv`)**

Для предоставления клиентам баз данных NIS NIS-сервер использует демона `ypserv`, который сконфигурирован файлом `/etc/ypserv.conf`. Как уже упоминалось, внутри домена можно запускать два NIS-сервера, `master` и `slave`. Набор баз данных NIS инициализируется на `master` сервере (только во время первого запуска; после этого используется `make -C /var/yp`) с помощью команды: `% /usr/lib/yp/ypinit -m`.

`slave` сервер в действительности -- просто NIS-клиент, который получает базы данных от `master` сервера и выполняет `ypserv`. Чтобы скопировать информацию с `master` сервера на локально запущенный `slave` сервер, выполняют `% /usr/lib/yp/ypinit -s my.nist.domain`.

На master сервере, базы данных NIS составляется из данных, размещенных в (некоторых из) следующих знакомых конфигурационных файлов:

- /etc/passwd,
- /etc/group,
- /etc/hosts,
- /etc/networks,
- /etc/services,
- /etc/protocols,
- /etc/netgroup,
- /etc/grc.

То, какие базы данных экспортируются, задается в /var/yp/Makefile, который также распространяет изменения при новой сборке.

Slave серверы оповещаются (посредством программы yppush) о любых изменениях в картах NIS, если они пересобираются, и автоматически возвращают необходимые изменения, чтобы синхронизировать свои базы данных. NIS-клиентам не требуется делать этого, поскольку они непрерывно общаются с NIS-сервером и считывают информацию, хранящуюся в его DBM базах данных.

## Конфигурация LDAP

### Когда используется LDAP

В принципе, по своим целям Lightweight Directory Access Protocol похож на NIS. Оба распределяют некоторую структурированную информацию о настройках сети от клиента к серверу; однако LDAP идет дальше в иерархическом структурировании того, каким клиентам какую информацию нужно предоставлять, перенаправляя запросы другим LDAP-серверам, куда необходимо, и выстраивая механизмы безопасности. Более того, LDAP дает клиентам механизмы и утилиты для обновления информации, содержащейся на LDAP-серверах, которые, в свою очередь, распространяют эти данные тем другим клиентам, что будут запрашивать их (конечно, это зависит от прав доступа).

### Установка

Перед запуском OpenLDAP (the Free Software реализация, широко используемая на Linux, хотя существуют и коммерческие разработки), вам следует установить или проверить, установлены ли следующие необходимые библиотеки:

- OpenSSL Transport Layer Security (TLS) можно получить на [OpenSSL Project](#) (или через способы установки вашего дистрибутива Linux).
- Поддержка Kerberos Authentication Services необязательна, но их наличие очень желательно. Пойдет либо [MIT Kerberos](#), либо [Heimdal Kerberos](#).
- Simple Authentication и Security Layer может быть установлена как часть основного дистрибутива, но также может быть получено [Cyrus SASL](#) as well.
- Рекомендуется иметь [Sleepycat Software Berkeley DB](#), хотя, вероятно, другие реализации DBM тоже подойдут.
- Posix нити и TCP wrappers если и не обязательны, то желательны.

После того, как эти предварительные условия соблюдены, скачайте [библиотеку OpenLDAP](#) и совершите более-менее привычный танец:

### Listing 7. Обычное камлание для установки OpenLDAP

```
% ./configure  
% make depend
```

```
% make  
% make test # make sure things went OK  
% su root -c 'make install'
```

После базовой установки, следует настроить конфигурацию slapd, обычно расположенную в /usr/local/etc/openldap/slapd.conf. Установка должна содержать компоненты вашего домена:

#### **Listing 8. Компоненты домена, включаемые в slapd.conf**

```
database bdb  
suffix "dc=eng,dc=uni,dc=edu,dc=eu"  
rootdn "cn=Manager,dc=eng,dc=uni,dc=edu,dc=eu"  
rootpw <secret>  
directory /usr/local/var/openldap-data
```

Для того, чтобы найти значение <secret> используйте утилита slappasswd, а затем берите эту base64-зашифрованную строку в качестве вашего "<secret>":

#### **Listing 9. Раскрытие вашего "секрета"**

```
% slappasswd  
New password: *****  
Re-enter new password: *****  
{SSHA}YzPqL5Jio2+17NFIy/pAz8pqS5Ko13fH
```

За большей информацией о правах доступа, репликации и других опциях, которые задаются в slapd.conf, обратитесь к manpages.

Запуск демона slapd daemon весьма схож с запуском любого другого демона; проверить, запущен ли он, можно с помощью ldapsearch:

#### **Listing 10. Проверка на то, запущен ли slapd**

```
su root -c /usr/local/libexec/slapd  
ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

Если все прошло успешно, вы увидите что-нибудь вроде этого:

#### **Listing 11. Отклик работающего slapd**

```
dn:  
namingContexts: dc=eng,dc=uni,dc=edu,dc=eu
```

### **Добавление данных в файл LDIF**

Формат данных, используемых в LDAP, является двоичным, а для экспорта и импорта данных в базу данных LDAP используется ASCII сериализация, которая называется LDAP

Data Interchange Format (LDIF). Двоичные данные в LDIF представляются base64 кодированном виде. OpenLDAP содержит утилиты, экспортирующие данные от LDAP-сервером в LDIF (**ldapsearch**), импортирующие данные из LDIF на LDAP-серверы (**ldapadd**), и применяющие набор изменений, описанных в LDIF, к LDAP-серверам (**ldapmodify** и **ldapdelete**).

Более того, LDIF -- один из форматов, использующихся в импортировании и экспортации данных адресной книги для Mozilla Application Suite и других пользовательских программ прикладного уровня. Даже Microsoft Windows 2000 Server и Windows Server 2003 содержат утилиту LDIF, **LDIFDE**, для передачи данных из Active Directory и обратно.

Чтобы вручную добавить информацию на LDAP-сервер, сначала создайте LDIF-файл:

### **Listing 12. Создание примерного файла LDIF, example.ldif**

```
dn: dc=example,dc=com
objectclass: dcObject
objectclass: organization
o: Example Company
dc: example

dn: cn=Manager,dc=example,dc=com
objectclass: organizationalRole
cn: Manager
```

Затем следует выполнить `% ldapadd -x -D "cn=Manager,dc=example,dc=com" -W -f example.ldif`, чтобы добавить его.

Очевидно, вам нужно поменять имя домена example.com на ваше. Как правило, структуры и имена доменов LDAP совпадают с обычными DNS именами. Вам понадобится указать значение `rootpw`, которое вы указали в `slapd.conf`.

### **Выполнение запросов в базах данных LDAP**

Имеется утилита **slurpd** (Standalone LDAP Update Replication Daemon), реплицирующая всю информацию базы данных; однако для отдельных данных используется либо программа типа `ldapsearch`, либо, что более вероятно, поддержка LDAP, встраивается в некоторые приложения, которые запускает пользователь. Утилита **slapcat** также пригодна для сброса базы данных LDAP в LDIF. Например, многие Mail User Agents (MUAs) могут использовать LDAP для выделения адреса и контактной информации.

Внутри приложений, включая и те, что вы напишете самостоятельно, используя компилируемые или скриптовые языки, к ресурсам LDAP можно обращаться при помощи специальных URL для LDAP. Они имеют вид `ldap://host:port/dn?attr ibutes?scope?filter?extens ions`.

Большая часть этих полей является необязательными. Имя хоста по умолчанию -- это `localhost`; порт, использующийся по умолчанию, 389. Умолчательное значение имени, распознаваемого как `root`, -- пустая строка. Если требуется информация для идентификации, она указывается в добавочных частях URL.

В добавок к LDAP URL, многие серверы и клиенты LDAP также поддерживают нестандартные, но все-же широко используемые LDAPS URL. LDAPS URL используют соединения SSL вместо обычных текстовых соединений, и их портом по умолчанию является 636: `ldaps://host:port/dn?attributes?scope?filter?extensions`.

## Идентификация PAM

### Когда используется PAM

Первое, что следует уяснить о Pluggable Authentication Modules (PAM) -- это, что это не есть ни приложение, ни протокол. Правильнее говорить, что это -- коллекция библиотек, которые могут включаться в приложения во время компиляции, чтобы использовать функции этих модулей. Если в приложении включены PAM, политика безопасности этого приложения может задаваться системным администратором, без изменения или обновления самого приложения. Многие утилиты Linux, в особенности демоны и сервера могут использовать PAM.

Быстрый способ выяснить, является ли данное приложение *возможно* использующим PAM, состоит в использовании `ldd`, чтобы выяснить, какие библиотеки используются. Например, мне интересно, является ли использующей PAM программа, выполняющая вход в систему:

### Listing 13. Вход PAM-задействован?

```
% ldd /bin/login | grep libpam
libpam.so.0 => /lib/libpam.so.0 (0xb7fa8000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb7fa5000)
```

Использование `libpam.so` и `libpam_misc.so` программой `login` не полностью гарантирует, что средства PAM действительно используются этими утилитами, и используются правильно, но это -- хороший знак. Подобным образом, мне, возможно, интересно узнать то же про мои сервера Apache и FTP:

### Listing 14. А как насчет серверов Apache и FTP?

```
% ldd /usr/sbin/apache2 | grep libpam
% ldd /bin/login | grep libpam
libpam.so.0 => /lib/libpam.so.0 (0xb7fa8000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb7fa5000)
```

Теперь я точно знаю, что моя сборка Apache не поддерживает PAM (хотя есть версии, для которых это не так).

Для более тщательной проверки, работают ли PAM с данным приложением, для данной программы можно создать конфигурационный файл PAM. Например, для проверки утилиты входа следует создать файл `/etc/pam.d/login` (однако обратите внимание, что он, возможно, уже имеется в вашей системе, и содержит более важные настройки, так что не удаляйте имеющийся файл):

### Listing 15. Проверка входа на PAM с помощью etc/pam.d/login

```
auth      required      pam_permit.so
auth      required      pam_warn.so
```

Теперь выполнение правильного поддерживающего PAM `login` позволит осуществить вход, но при этом занесет действия в системный журнал. Если `syslog` покажет вход, значит для этого приложения PAM задействован. Читатель, наверно, заметит, что это -- наверно, худшая

конфигурация, которую можно придумать для `login`, так как теперь он даст доступ к оболочке *каждому*. Подметив это, уясните, что настраивать PAM следует с определенной осторожностью.

## Настройка PAM

### PAM работает с двумя видами конфигурационных файлов.

Предпочтительный способ настройки PAM состоит в использовании файлов из каталога `/etc/pam.d/`, которые имеют те же имена, что и службы, чья безопасность является желаемой. Более старый и менее рекомендуемый способ состоит в использовании одного файла, `/etc/pam.conf`, для указания политики безопасности для всех приложений. С точки зрения возможности поддержки работы, создание по конфигурационному файлу на приложение проще, вместо файлов можно создать символические ссылки для "копирования" политика безопасности от одного приложения на другое. Оба стиля конфигурации в основном совпадают. Единый файл `/etc/pam.conf` содержит строки вида:

#### Listing 16. Оба конфигурационных файла содержат

```
<service> <module-type> <control-flag> <module-path> <args>
```

В конфигурационных файлах отдельных приложений первое поле опускается, поскольку оно такое же, как и имя файла. Тестовая конфигурация входа, которую мы видели, в старом стиле выглядит как:

#### Listing 17. /etc/pam.conf

```
login      auth      required    pam_permit.so
login      auth      required    pam_warn.so
```

Поле `<module-type>` может принимать одно из четырех значений:

- `auth` (автентификация),
- `account` (неидентификационные права доступа, основанные на состоянии пользователя в системе),
- `session` (совершать действия до/после запуска службы), and
- `password` (обновление tokenов идентификации пользователя).

Поле `<control-flag>` используется для "stack" модулей, которые позволяют вам вести довольно утонченный контроль того, когда этот метод требуется, требуется ли он вообще, и когда принимается какой-нибудь другой исход. Эти опции таковы:

- `required`,
- `requisite`,
- `sufficient`,
- `optional`, and
- `include`.

Я расскажу об этом ниже.

Поле `<module-path>` уже появлялось в примерах. Оно обозначает разделяемую библиотеку - либо указывая явный путь к ней (если значение начинается с "/", либо только имя (при этом сама библиотека ищется в каталогах по умолчанию). Например, в Listing 17, можно было бы

написать `/lib/security/pam_warn.so`. Полем `<args>` может быть все, что угодно, в зависимости от того, какой именно модуль требуется для настройки этой операции, хотя несколько аргументов общего типа должны поддерживаться большинством модулей PAM. Обратите внимание, что модули PAM являются расширяемыми. Если кто-нибудь пишет новый модуль PAM, его можно просто поместить в `/lib/security` и все ваши приложения смогут использовать его, как только их конфигурационный файл будет обновлен.

### Пример прав доступа PAM.

Чтобы увидеть, как работает `<control-flag>`, давайте сконструируем довольно сложный пример. Первое, что следует сделать -- это создать специальную службу *OTHER*. Если это сделано, и для этой службы не определена никакая политика PAM, используется политика *OTHER*. Безопасное *default* будет пожалуй на Listing 18:

#### Listing 18. `/etc/pam.d/other`

```
auth      required    pam_warn.so
auth      required    pam_deny.so
@include safe-account
@include safe-password
@include safe-session
```

В этом примере, попытка идентификации сначала журналируется, а затем получает отказ. `@include` просто включает содержимое какого-нибудь файла, например `/etc/pam.d/safe-account` и подобных, в то время как эти "безопасные" определения должны содержать похожие предупредить-и-отклонить инструкции для других `<module-type>`-ов.

Теперь давайте настроим доступ к нашему гипотетическому секретному-db приложению. Будучи довольно озабоченным доступом, пользователь должен представить либо соответствующий отпечаток сетчатки глаза, либо отпечаток пальца, либо же ввести пароль. Пароль, однако, должен храниться либо в локальных конфигурациях `/etc/passwd` и `/etc/shadow`, или быть доступными посредством одного или двух внешних серверов баз данных.

Ни один из модулей безопасности, которые приводятся в этом примере, в действительности не существуют (насколько мне известно), кроме `pam_unix.so`, который есть доступ к паролю в устаревшем UNIX-стиле.

#### Listing 19. `/etc/pam.d/classified-db`

```
auth      optional   pam_unix.so
auth      optional   pam_database1.so  try_first_pass
auth      optional   pam_database2.so  use_first_pass
auth      requisite  pam_somewhatso
auth      sufficient pam_fingerprint.so master=file1 7points
auth      required   pam_retinaprint.so
```

Путь через эту конфигурацию весьма сложен. Вначале мы пытаемся идентифицировать пароль тремя `optional` типами модулей. Поскольку они `optional` (необязательны), отказ одного ни останавливает процесса идентификации, ни удовлетворяет его. Сначала пробуется стандартный пароль UNIX (пользователя просят ввести пароль). После этого пароль проверяется в `database1`, но вначале используется общий аргумент модуля `try_first_pass`, чтобы убедиться, совпадает ли пароль UNIX с паролем в базе данных;

дополнительный пароль вводится только в случае несовпадения. Для базы данных `database2`, однако, мы лишь пытаемся идентифицировать, используя пароль UNIX, введенный пользователем (общий аргумент `use_first_pass`).

После проверки пароля в нескольких опциональных (`optional`) системах паролей, у нас есть гипотетический `param_somerpasswd.so`, которому нужно определить, увенчалась ли успехом какая-нибудь из ранее проведенных проверок паролей (возможно, с использованием семафоров; но вопрос, как это делать остается открытым для гипотетических модулей). Но поскольку эта проверка является `requisite`(необходимый), то если он не срабатывает, дальнейшая идентификация не производится, и докладывается о неудаче.

Последние две идентификационные проверки (если дело доходит до них) являются `sufficient`. То есть, удовлетворение одной из них возвращает вызываемому приложению состояние успеха. То есть вначале мы делаем проверку отпечатка пальца, используя `param_fingerprint.so` (обратите внимание, что гипотетические аргументы передаются модулю). И только если здесь будет неудача -- может быть, из-за отсутствия сканера отпечатков пальцев, или же из-за плохого отпечатка -- предпринимается попытка сканировать сетчатку глаза. Далее, если сканирование сетчатки имело успех, этого `sufficient`(достаточно). Однако, чтобы продемонстрировать все возможные опции, мы использовали также `required`, что означает, что даже если сканирование сетчатки дало успех, следует продолжить проверку другими методами (но в примере других нет, так что `sufficient` будет делать то же самое).

Также существует способ указать еще более тонко настроенные составные флаги для `<control-flag>` при помощи взятых в скобки `[value1=action1 ...]` множеств, но обычно хватает основных ключевых слов.

## Ресурсы

### Научиться

- Ознакомьтесь с полным [учебным курсом подготовки к экзамену LPI](#) на developerWorks, дабы изучить основы Linux и подготовиться и аттестации системного администратора.
- Вам так же доступна [оригинальная версия](#) этого учебника (на английском языке).
- На [LPIC Program](#), найдите список задач, примерных вопросов и подробных целей для трех уровней аттестации системных администраторов Linux Профессионального Института Linux.
- Узнайте о [DHCP](#) (RFC 2131) его из формальной спецификации.
- Почтайте "[The Linux NIS\(YP\)/NYS/NIS+ HOWTO](#)" для лучшего знакомства с NIS.
- Формальная спецификация [LDAP -- это RFC 2251](#).
- [TCP/IP Network Administration, Третье издание](#) by Craig Hunt (O'Reilly, April 2002), Крейг Хант (O'Reilly, April 2002) является отличным ресурсом по сетям Linux.
- Здесь перечислено более [700 Linux User Groups по всему свету](#), которые могут помочь найти локальные и дистанционные группы обучения для экзаменов LPI.
- Для углубленного изучения, [Linux Documentation Project](#) располагает множеством различных полезных документов, в особенности HOWTO.
- На [developerWorks Linux-зоне](#), найдите больше ресурсов для разработчиков Linux.
- Оставайтесь в курсе [технических событий developerWorks и Webcasts](#).

## **Получить продукты и технологии**

- Вместе с [тестовое ПО IBM](#), доступном для скачивания непосредственно с developerWorks, постройте ваш следующий проект на Linux.

# Подготовка к экзамену LPI: Системная безопасность

Средний уровень администрирования (LPIC-2) тема 212

[Дэвид \(David\) Мертц \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** В этом учебном пособии, шестом из [серии, состоящей из семи пособий](#) covering intermediate network administration on Linux®, Дэвид Мертц продолжает готовить вас к сдаче экзамена 202 Администрирования Среднего Уровня (LPIC-2) Профессионального Института Linux. По необходимости, это пособие кратко затрагивает широкий круг вопросов, связанных с Linux с точки зрения безопасности сервера сети, включая общие вопросы маршрутизации, межсетевых экранов, преобразования NAT и соответствующие утилиты. Здесь нашлось место для задания политики безопасности для FTP и SSH; также рассматривается общий контроль доступа с помощью tcpd, hosts.allow, и их товарищем; представлены некоторые основные утилиты отслеживания безопасности, рассказано, где найти ресурсы по безопасности.

[Больше статей из этой серии](#)

**Дата:** 13.06.2006

**Уровень сложности:** средний

## Перед тем, как начать

Узнайте, чему могут научить эти пособия, и как получить максимум от них.

## Об этих курсах

[Профессиональный Институт Linux](#) аттестовывает системных администраторов Linux по двум уровням: *младший уровень* (или "аттестация уровня 1") и *средний уровень* (или "аттестация уровня 2"). Для получения аттестация уровня 1, нужно сдать экзамены 101 и 102; для получения аттестация уровня 2, следует сдать экзамены 201 и 202.

developerWorks предоставляет учебные пособия, чтобы помочь вам подготовиться ко всем четырем экзаменам. Каждый экзамен охватывает несколько тем, и каждая тема имеет соответствующее учебное пособия на developerWorks. Для экзамена LPI 202, семь тем и соответствующие developerWorks пособия таковы:

Таблица 1. Экзамен LPI 202: Темы и учебные пособия

Тема экзамена	Учебные пособие developerWorks	Краткое содержание
LPI 202	<a href="#">LPI exam 202 prep (topic 205): Конфигурация сети</a>	Узнайте, как настроить базовую сеть TCP/IP, от аппаратного уровня (как правило, это Ethernet, или, ISDN, или 802.11) до маршрутизации сетевых адресов.
Тема 205	<a href="#">LPI exam 202 prep (topic 206): Почта и новости</a>	Узнайте, как использовать Linux в качестве почтового сервера и новостного сервера. Узнайте о почтовом транспорте, фильтре локальной почты, фильтре локальной почты, программах по управлению списком рассылки и серверных приложениях для протокола NNTP.

Тема 207	<a href="#">Подготовка к LPI экзамену 202 (тема 207): DNS</a>	Узнайте, как использовать Linux в качестве DNS-сервера, главным образом, с использованием BIND. Узнайте, как осуществить базовую настройку BIND, управлять зонами DNS, и обеспечивать безопасность DNS-сервера.
Тема 208	<a href="#">Подготовка к LPI экзамену 202 (тема 208): Web-сервисы</a>	Узнайте, как установить и настроить web-сервер Apache, и как использовать прокси-сервер Squid.
Тема 210	<a href="#">Подготовка к LPI экзамену 202 (тема 201): Управление клиентом сети</a>	Узнайте, как настроить сервер DHCP, клиента NIS и NISD сервер, сервер LDAP, и поддержку идентификации PAM. За подробностямисмотрите <a href="#">цели</a> ниже.
Тема 212	Подготовка к LPI экзамену 202 (тема 212): Системная безопасность	(Это пособие) Узнайте, как настроить маршрутизатор, безопасные FTP-серверы, сконфигурировать SSH, и производить различные другие задачи администрирования безопасности. За подробностямисмотрите <a href="#">цели</a> ниже.
Тема 214	Подготовка к LPI экзамену 202 (тема 214): Устранение неполадок работы сети	Скоро будет

Чтобы начать готовиться к certification уровня 1, посмотрите [учебные пособия developerWorks для экзамена LPI 101](#). Для подготовки к другим экзаменам certification уровня 2, посмотрите [учебные пособия developerWorks для экзамена LPI 201](#). Read больше о [полном курсе LPI](#) учебных пособий от developerWorks.

Профессиональный институт Linux не поддерживает никаких материалов по подготовке к экзаменам, разработанных третьими лицами. За подробностями обращайтесь на [info@lpi.org](mailto:info@lpi.org).

## Об этом пособии

Добро пожаловать в "Системную безопасность", шестое пособие из серии, состоящей из семи пособий, охватывающих сетевое администрирование под Linux. В этом пособии вы познакомитесь с широким кругом вопросов, связанных с Linux с точки зрения безопасности сервера сети. Охвачены такие вопросы, как маршрутизация, межсетевые экраны, преобразования NAT (и утилиты, управляющие этим), равно как и задание политики безопасности для FTP и SSH. Также научитесь задавать общий контроль доступа с помощью tcpd, hosts.allow, и их товарищей (возвращаясь к вопросам, которые мы рассматривали в пособии [Подготовка к экзамену LPI 202 \(тема 209\): Совместное использование файлов и служб](#)). Наконец, здесь затронуты некоторые основные утилиты отслеживания безопасности, рассказано, где найти ресурсы по безопасности.

Наряду с другими учебными пособиями из developerWorks курсов 201 и 202, представленный материал предназначен скорее служить руководством к изучению и стартовой точкой при подготовке к экзаменам, и не является полной документацией по данному предмету. Поощряется, если читатель обращается к [подробному списку целей и задач](#) LPI и, по необходимости, дополняет информацию, представленную здесь, другими материалами.

Это учебное пособие организовано согласно LPI задачам для этой темы. Грубо говоря, ожидайте на экзамене задач с большим весом.

*Таблица 2. Системная безопасность: Экзаменационные темы, рассматриваемые в этом учебнике*

<b>Тема LPI экзамена</b>	<b>Вес</b>	<b>Краткое содержание</b>
2.212.2 <a href="#"><u>Настройка маршрутизатора</u></a>	2	Настройка системы, осуществляющей трансляцию сетевого адреса (NAT, IP маскарадинг), и обоснование важности этого для защиты сети. Сюда включается настройка переназначения портов, управление правилами фильтров, и отвод атак.
2.212.3 <a href="#"><u>Обеспечение безопасности FTP-серверов</u></a>	2	Настройка FTP-сервера на возможность анонимного скачивания и загрузку. Эта тема содержит предосторожности, о которых следует позаботиться, если разрешена анонимная загрузка файлов, а также настройка прав пользователей.
2.212.4 <a href="#"><u>Безопасная оболочка (SSH)</u></a>	2	Настройка демона SSH. Эта цель включает управление ключами, настройку SSH для пользователей, пересылка протокола приложения через SSH, и управление входом в SSH.
2.212.5 <a href="#"><u>TCP-wrappers</u></a>	1	Настроить TCP-wrappers, чтобы разрешить соединения только к определенным серверам только из определенных хостов или подсетей.
2.212.6 <a href="#"><u>Задачи по безопасности</u></a>	3	Установить и настроить систему безопасной идентификации; произвести базовый аудит безопасности исходного кода; получить сигналы тревоги от нескольких источников; провести аудит серверов на предмет открытых e-mail реле и анонимных FTP-серверов; установить, настроить, и запустить систему опознавания вторжений; применить патчи безопасности и исправления ошибок.

## Предпосылки

Для полной отдачи от этого пособия, вы должны уже иметь основные знания о Linux и рабочую систему Linux, где вы можете опробовать команды, описанные в этом пособии.

## Другие источники

Как и другие приложения Linux, всегда полезно обращаться к тап-страницам любых рассматриваемых здесь утилит. За большей информацией, Linux Documentation Project имеет большое количество различных полезных документов, в особенности HOWTO. Также опубликовано множество книг по сетям Linux; я считаю, что книга O'Reilly's *TCP/IP Network Administration*, Крейга Ханта, вполне может помочь. (Ссылки [Источники](#) располагаются ниже.)

# Подготовка к экзамену LPI: Системная безопасность

Средний уровень администрирования (LPIC-2) тема 212

Дэвид (David) Мертц (Mertz), Developer, Gnosis Software, Inc.

**Описание:** В этом учебном пособии, шестом из серии, состоящей из семи пособий covering intermediate network administration on Linux®, Дэвид Мертц продолжает готовить вас к сдаче экзамена 202 Администрирования Среднего Уровня (LPIC-2) Профессионального Института Linux. По необходимости, это пособие кратко затрагивает широкий круг вопросов, связанных с Linux с точки зрения безопасности сервера сети, включая общие вопросы маршрутизации, межсетевых экранов, преобразования NAT и соответствующие утилиты. Здесь нашлось место для задания политики безопасности для FTP и SSH; также рассматривается общий контроль доступа с помощью `tcpd`, `hosts.allow`, и их товарищем; представлены некоторые основные утилиты отслеживания безопасности, рассказано, где найти ресурсы по безопасности.

**Дата:** 13.06.2006

**Уровень сложности:** средний

## Настройка маршрутизатора

### О фильтре пакетов

Ядро Linux содержит инфраструктуру "netfilter", которая позволяет вам отфильтровывать сетевые пакеты. Обычно эта возможность компилируется как неотъемлемая часть ядра, но можно собрать в виде модуля. В любом случае, загрузка модуля должна быть прозрачной (например, запуск `iptables` приведет к загрузке `iptables_filter.o`, если это потребуется).

В более новых системах Linux фильтрация пакетов управляет утилитой `iptables`; более старые системы использовали `ipchains`. А до этого применялась `ipfwadm`. Хотя, если требуется обратная совместимость, вы все еще можете использовать `ipchains` вместе с недавними версиями ядра, вы все равно предпочтете более широкие возможности и улучшенный синтаксис в `iptables`. То есть, большая часть понятий и опций в `iptables` являются совместимыми улучшениями `ipchains`.

В зависимости от точного сценария фильтра (firewall, NAT, и тд), фильтрация и перевод адреса может произойти либо до, либо после маршрутизации как таковой. В обоих случаях используется одна и та же программа `ipchains`, но используют различные правила ("цепочки") -- в основе, `INPUT` и `OUTPUT`. Однако, фильтрация также может повлиять на решение маршрутизации, из-за фильтрации `FORWARD` цепочки; этот способ может привести к исчезновению пакетов, вместо их маршрутизирования.

### Маршрутизация

Наряду с фильтрацией при помощи `iptables` (или более ранней `ipchains`), ядро Linux производит *маршрутизацию* IP-пакетов, которых получает. Маршрутизация -- более простой процесс, нежели фильтрация, хотя они умозрительно являются связанными.

Во время маршрутизации, хост просто смотрит на IP-адрес назначения и решает, знает ли он, как доставить пакет непосредственно на этот адрес, или же ему доступен шлюз, который знает, как доставить на этот адрес. Если хост не может ни доставить пакет сам, ни знает шлюза, которому можно поручить доставку, пакет теряется. Однако типичная конфигурация содержит "шлюз по умолчанию", который обрабатывает любой адрес, не определенный

каким-нибудь способом.

Настройка и отображение информации по маршрутизации производится утилитой `route`. Однако маршрутизация может быть либо *статической*, либо *динамической*.

При статической маршрутизации, доставка определяется таблицей маршрутизации, которая явно настраивается вызовом команды `route` и ее командами `add` или `del`. Более полезной, однако, может оказаться настройка динамической маршрутизации с использованием демонов `routed` или `gated`, которые рассылают информацию о маршрутизации соседним демонам-маршрутизаторам.

Демон `routed` поддерживает Routing Information Protocol (RIP); демон `gated` вдобавок имеет поддержку других протоколов -- и может пользоваться многими протоколами одновременно -- таких как:

- Routing Information Protocol Next Generation (RIPng)
- Exterior Gateway Protocol (EGP)
- Border Gateway Protocol (BGP) и BGP4+
- Defense Communications Network Local-Network Protocol (HELLO)
- Open Shortest Path First (OSPF)
- Intermediate System to Intermediate System (IS-IS)
- Internet Control Message Protocol (ICMP и ICMPv6)/Router Discovery

Давайте взглянем на довольно типичную статическую таблицу маршрутизации:

### Listing 1. Типичная статическая таблица маршрутизации

```
% /sbin/route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
66.98.217.0     *              255.255.255.0   U      0      0      0 eth0
10.10.12.0      *              255.255.254.0   U      0      0      0 eth1
66.98.216.0     *              255.255.254.0   U      0      0      0 eth0
169.254.0.0     *              255.255.0.0    U      0      0      0 eth1
default         ev1s-66-98-216- 0.0.0.0       UG      0      0      0 eth0
```

Это означает, что адреса в диапазонах 66.98.217/24 и 66.98.216/23 напрямую доставляются через `eth0`. Диапазоны адресов 10.10.12/23 и 169.254/16 доставляются на `eth1`. Все, что осталось, посыпается на шлюз `ev1s-66-98-216-1.ev1servers.net` (имя обрезано в выводе `route`; также можно использовать `route -n`, чтобы убедиться, что это имя имеет IP-адрес 66.98.216.1). Если вы хотите добавить другой шлюз для некоторых диапазонов адресов, вам следует выполнить следующее:

### Listing 2. Добавление нового шлюза для других диапазонов адресов

```
% route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.2.1 dev eth0
```

Для машины, которая сама по себе является шлюзом, вообще говоря, используется динамическая маршрутизация, при помощи демонов `routed` или `gated`, которые могут давать меньшее число статических маршрутов. Демон `routed` настраивается содержимым `/etc/gateways`. Демон `gated` является более современным, и, как уже говорилось, имеет больше возможностей; он настраивается файлом `/etc/gated.conf`. Вообще говоря, при

использовании какого-нибудь из этих демонов, вы можете запускать через сценарии запуска. Но *нельзя* запускать демонов `routed` и `gated` на одной машине, поскольку результаты станут непредсказуемыми и, конечно же, нежелательными.

### Фильтрация с помощью `iptables`

Ядро Linux хранит таблицу правил фильтрации IP-пакетов, которая образует некоторую разновидность машины состояний. Наборы правил объединяются в последовательности, известные как "цепочки". Если одна цепочка встречает условие, одним из возможных действий является передача управление на обработку другой цепочки, как делается в машине состояний. Перед добавлением правил или состояний, автоматически присутствуют три цепочки: `INPUT`, `OUTPUT`, и `FORWARD`. Цепочка `INPUT` работает когда пакет, адресованный машине-хосту, передаётся процессу локального приложения. Цепочка `FORWARD` используется, когда приходит пакет, адресованный другой машине, полагая, что здесь активирована переадресация, и система маршрутизации знает, как переслать пакет дальше. Пакет, порожденный на локальном хосте, посыпается для фильтрации в цепочку `OUTPUT` -- если он проходит фильтры в цепочке `OUTPUT` (или любых связанных цепочек), он маршрутизуется за пределы сетевого интерфейса.

Одно действие, которое может предпринять правило, есть `DROP` (уничтожение) пакета; в этом случае для этого пакета больше никаких дальнейших обработок правил или переходов между состояниями не предпринимается. Но если пакет не уничтожается, то проверяется, соответствует ли пакету следующее правило в цепочке. В некоторых случаях соответствие правилу перенесёт процесс обработки на другую цепочку со своим набором правил. Создание, удаление или изменение правил и цепочек, где содержатся эти правила, производится утилитой `iptables`. В старых системах Linux, эти же функции выполняла утилита `ipchains`. Идеи, реализованные в этих утилитах, и даже в более древней `ipfwadm` похожи, но здесь мы обсудим синтакс `iptables`.

Правило определяет набор условий, которым пакет может соответствовать, и то, какое действие следует произвести, если пакет не соответствует условию. Как упоминалось, одно общее действие состоит в `DROP` (уничтожении) пакетов. Например, предположим, что вам нужно (по каким-то причинам) отключить `ping` в петлевом интерфейсе (интерфейсе ICMP). Это можно осуществить с помощью:

#### Listing 3. Отключение петлевого интерфейса

```
% iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
```

Конечно, это правило глупое, и его, скорее всего, следует убрать после тестирования, примерно вот так:

#### Listing 4. Как убрать это глупое правило

```
% iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP
```

Удаление правила с опцией `-D` требует либо в точности те же параметры, что были указаны при добавлении правила, либо описания с помощью номера правила (который *нужно* определять на первом месте) вот так:

#### Listing 5. Указание номера правила, так чтобы удаление сработало

```
% iptables -D INPUT 1
```

Более интересное правило может глядеть на адрес источника и назначения в пакетах. Например, представьте, что подозрительная удаленная сеть пытается использовать службы на определенной подсети вашей сети. Это можно заблокировать на машине вашего шлюза/межсетевого экрана таким образом:

#### **Listing 6. Блокировка машины шлюза/межсетевого экрана**

```
% iptables -A INPUT -s 66.98.216/24 -d 64.41.64/24 -j DROP
```

Это остановит все, что исходит из **66.98.216.\*** IP-диапазона на локальную подсеть **64.41.64.\***. Конечно, использование этого метода занесения определенного IP-диапазона в черный список в качестве защиты довольно ограничено. Более вероятным сценарием будет *разрешение* доступа к локальной подсети только от определенного диапазона IP:

#### **Listing 7. Разрешение указанному диапазону IPиметь доступ к локальной подсети**

```
% iptables -A INPUT -s ! 66.98.216/24 -d 64.41.64/24 -j DROP
```

В этом случае *только* адреса из IP диапазона **66.98.216.\*** могут иметь доступ к указанной подсети. Более того, для адреса можно использовать символическое имя, и можно указывать определенный протокол для фильтрации. Также можно выбрать для фильтрации определенный сетевой интерфейс (например, **eth0**), но это используется не так широко. Например, чтобы разрешить только отдельной удаленной сети обращаться к локальному web-серверу, следует использовать:

#### **Listing 8. Разрешение отдельной удаленной сети обращаться к локальному Веб-серверу**

```
% iptables -A INPUT -s ! example.com -d 64.41.64.124 -p TCP -sport 80 -j DROP
```

Можно указать еще и некоторые другие опции для **iptables**, например, включая ограничения на число допустимых пакетов или фильтрацию по флагам TCP. За подробностями обращайтесь к *man*-страницам **iptables**.

### **Цепочки, определенные пользователем**

Мы видели основы добавления правил в автоматическую цепочки. Но гибкая настраиваемость **iptables** возможна при добавлении цепочек, определенных пользователем и переход к ним при совпадении с шаблоном. Новые цепочки определяются с использованием опции **-N**; мы уже осуществляли переход с помощью специальной цели **DROP**. **ACCEPT** (разрешить) -- это тоже специальная цель с очевидным значением. Также доступны специальные цели **RETURN** (вернуть) и **QUEUE** (поставить в очередь). Первая означает остановку обработки данной цепочки и возврат к вызывавшему/породителю ее. Обработчик **QUEUE** позволяет передавать пакеты к процессам пользователя для дальнейшей обработки (это может быть журналирование, изменение пакета, или более сложная фильтрация, нежели та, что

поддерживается **iptables**). Простой пример из книги Русти Рассела "Linux 2.4 Packet Filtering HOWTO" -- хорошая иллюстрация добавления пользовательских цепочек:

### **Listing 9. Добавление пользовательской цепочки**

```
# Создание цепочки для блокировки всех соединений
# за исключением локальных или уже существующих
% iptables -N block
% iptables -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
% iptables -A block -m state --state NEW -i ! ppp0 -j ACCEPT
% iptables -A block -j DROP # Уничтожаем всё, что не было разрешено (ACCEPT)
# Переключение на эту цепочку с цепочек INPUT и FORWARD
% iptables -A INPUT -j block
% iptables -A FORWARD -j block
```

Обратите внимание, что цепочка **block** принимает (**ACCEPT**) в ограниченном числе случаев, тогда как последнее правило пропускает (**DROP**) все, что не было принято ранее.

По мере того, как были созданы новые цепочки, либо путем добавления правил к автоматическим, либо при помощи новых пользовательских цепочек, можно использовать опцию **-L** для просмотра текущих правил.

### **Преобразование сетевых адресов в сравнении с межсетевыми экранами**

Примеры, которые мы видели, в основном касались правил для межсетевых экранов. Но преобразование сетевых адресов (NAT) так же настраивается с помощью **iptables**.

Как правило, NAT -- это способ использования отслеживания соединений для маскаранинга пакетов, исходящих из адреса локальной подсети в качестве внешнего адреса WAN, перед тем, как отправлять из дальше "по проводам" (на цепочке **OUTPUT**). Шлюз/маршрутизатор, выполняющий NAT, должен запоминать, какой локальный хост соединен с каким удаленным хостом, и обращать преобразование адреса, если вдруг пакеты приходят обратно с удаленной машины.

с точки зрения системы фильтрации просто делается вид, что NAT не существует. Правила, определяемые нами, просто используют "настоящие" локальные адреса, безотносительно тому, как NAT замаскирует их для внешнего мира. Включение маскарадинга, такого как базовый NAT, просто использует описанную ниже команду **iptables**. Для этого следует сначала убедиться, что модуль ядра **iptables\_nat** загружен, а затем включить IP-пересылку:

### **Listing 10. Включение маскарадинга**

```
% modprobe iptables_nat # Загрузка модуля ядра
% iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
% echo 1 > /proc/sys/net/ipv4/ip_forward # Включение маршрутизации
```

Эта способность называется *NAT источника* -- адрес исходящего пакета изменен. Также существует *NAT назначения* (DNAT), позволяя осуществлять пересылку порта, разделение загрузки, и прозрачное проксирования. В этих случаях, входящие пакеты изменяются так, чтобы достигнуть требуемого хоста или подсети.

Но в большинстве случаев, когда пользователи или администраторы говорят о NAT, они имеют ввиду NAT источника. Если хотите настроить NAT назначения, следует указать **PREROUTING**, а не **POSTROUTING**. При DNAT, пакеты преобразовываются перед

маршрутизацией.

## Обеспечение безопасности FTP-серверов

### FTP-сервера

Для Linux существует множество различных FTP-серверов, и разные дистрибутивы предлагают разные сервера. Естественно, настройки различных серверов различаются, хотя большинство склоняется использовать похожие конфигурационные директивы.

Популярным FTP-сервером является vsftpd (Very Secure FTP daemon). ProFTP также широко используется, равно как wu-ftpd и ncftpd.

Для многих целей FTP, в сущности, и не нужен. Например, безопасную передачу файлов для пользователей, имеющих учетные записи на машине сервера, можно часто совершать при помощи `scp` (secure copy), которая основывается на SSH, но в использовании очень похожа на знакомую команду `cp`.

Файл конфигурации для vsftpd -- это `/etc/vsftpd.conf`. Другие FTP-сервера используют похожие файлы.

### Конфигурационные опции FTP

Существует несколько опций для `/etc/vsftpd.conf` (и, возможно, вашего сервера, если у вас другой), которые следует помнить:

- `anonymous_enabled`: Позволяет анонимным пользователям входить, используя имена пользователей "anonymous" или "ftp".
- `anon_mkdir_write_enable`: Позволяет анонимным пользователям создавать каталоги (внутри открытых всем для записи родительских каталогов).
- `anon_upload_enable`: Позволяет анонимным пользователям загружать файлы.
- `anon_world_readable_only`: По умолчанию "YES"; вряд ли менять его -- хорошая идея. Позволяет анонимному пользователю FTP считывать только к файлы, помеченные как общедоступные для чтения.
- `chroot_list_enable`: Помещает пользователей (перечисленных в `/etc/vsftpd.chroot-list`) в "тюрьму chroot" в их домашнем каталоге (они не смогут просмотреть файлы и каталоги выше в иерархии файловой системы, даже если имеют на них права доступа).
- `ssl_enable`: Поддерживает SSL-шифрованные соединения.

Ознакомьтесь с man-страницами для вашего FTP-сервера за более сложными опциями.

Вообще говоря, запуск FTP-сервера настолько прост, насколько таковым является выбор конфигурационного файла и запуск сервера из сценариев инициализации системы.

### Безопасная оболочка (SSH)

#### Клиент и сервер

Почти каждая Linux-машина (да и большинство других операционных систем) должны иметь клиента безопасной оболочки (SSH). Часто используется версия OpenSSH, но также иногда применяют и других совместимых SSH клиентов. SSH клиент является необходимым для соединения, большие проблемы с безопасностью возникают при неправильной настройке SSH-сервера.

Поскольку клиент запускает соединение с сервером, ему часто приходится доверять серверу. Просто наличие SSH-клиента не дает никакого доступа *внутрь* машины; следовательно, это не может представлять опасности.

Настройка сервера тоже не особенно сложная; демон сервера был разработан содержащим и реализующим хорошую схему безопасности. Но, очевидно, именно сервер разделяет ресурсы

с клиентами, основываясь на запросах от клиентов, которым сервер решает доверять.

Протокол SSH имеет две версии, версию 1 и версию 2. В современных системах всегда предпочтительным является использования протокола версии 2, но, вообще говоря, как клиент, так и сервер поддерживают обратную совместимость с версией 1 (если только если это не отключено конфигурационными опциями). Это позволяет соединяться с все реже и реже встречающимися системами версии 1.

Протоколам версий 1 и 2 соответствуют несколько различных конфигурационные файлы. В протоколе версии 1 клиент сначала создает с помощью ssh-keygen пару ключей RSA, и хранит закрытый ключ в \$HOME/.ssh/identity, а открытый ключ -- в \$HOME/.ssh/identity.pub. Точно такой же identity.pub должен быть добавлен к удаленным файлам \$HOME/.ssh/authorized\_keys.

Очевидно, здесь присутствует проблема яйца и курицы: как можно скопировать файл на удаленную систему, до получения доступа к ней? К счастью, SSH также поддерживает метод аварийной идентификации, состоящий в отправке зашифрованных на лету паролей, которые вычисляются посредством обычного контроля входа на удаленную систему (таких как существует ли учетная запись пользователя, и введен ли правильный пароль).

Протокол 2 поддерживает как ключи RSA, так и DSA, но RSA-идентификация немного улучшена по сравнению с протоколом 1. В протоколе 2, закрытые ключи хранятся в \$HOME/.ssh/id\_rsa и \$HOME/.ssh/id\_dsa. Протокол 2 также поддерживает некоторое количество алгоритмов экстра конфиденциальности и целостности: AES, 3DES, Blowfish, CAST128, HMAC-MD5, HMAC-SHA1, и так далее. Сервер можно настроить как на предпочтительные алгоритмы, так и задать порядок использования аварийных режимов.

Общие конфигурационные опции, в отличие от ключа, содержатся у клиента в /etc/ssh/ssh\_config (или, если это доступно, в \$HOME/.ssh/config). Параметры клиента можно также настроить с помощью опции -O; часто используемой опцией является -X или -X, для включения или выключения переадресации X11. Если она активирована, порт X11 туннелируется через SSH, чтобы осуществлять шифрованные X11 соединения.

Утилиты, подобные **Scp** также используют похожий порт для переадреации через по SSH. Пусть, например, я работаю за локальной машиной, и я могу запустить увидеть на дисплее X11 приложение, которое работает только удаленно (в этом случае -- на сервере в локальной подсети):

### **Listing 11. запуск удаленного приложения X11**

```
$ which gedit # на локальной системе отсутствует
$ ssh -X dqm@192.168.2.2
Password:
Linux averatec 2.6.10-5-386 #1 Mon Oct 10 11:15:41 UTC 2005 i686 GNU/Linux
No mail.
Last login: Thu Feb 23 03:51:15 2006 from 192.168.2.101
dqm@averatec:~$ gedit &
```

## **Настройка сервера**

Демон **sshd**, что особенно характерно для версии OpenSSH, включает безопасные шифрованные соединения между двумя ненадежными хостами по небезопасной сети. Основной **sshd** сервер обычно запускается во время инициализации системы и ждет сигнала от подключений пользователей, запуская новый экземпляр демона для каждого пользовательского соединения. Отделенные демоны осуществляют обмен ключами, шифрование, идентификацию, выполнение команд и обмен данными.

Сервер `sshd` поддерживает большое разнообразие опций командной строки, но обычно настраивается файлом `/etc/ssh/sshd_config`. Также используется некоторое число других конфигурационных файлов. Например, файлы контроля доступом `/etc/hosts.allow` и `/etc/hosts.deny` являются предпочтительными. Ключи хранятся похожим образом на стороне клиента, в `/etc/ssh/ssh_host_key` (протокол 1), `/etc/ssh/ssh_host_dsa_key`, `/etc/ssh/ssh_host_rsa_key`, а открытые ключи -- в `/etc/ssh/ssh_host_dsa_key.pub` и подобных. Также, на клиенте, для генерации ключей используется `ssh-keygen`. Обратитесь к `man`-страницам для `sshd` и `ssh-keygen` за деталями конфигурационных файлов и копирования сгенерированных ключей в соответствующие файлы.

Множество конфигурационных опций имеется в `/etc/ssh/sshd_config`, а значения по умолчанию, вообще говоря, чувствительны (и ощутимо безопаснее). Стоит упомянуть несколько опций:

- `AllowTcpForwarding` включает или выключает переадресацию портов, и по умолчанию выставлено "YES".
- `Ciphers` управляет списком и порядком шифровательных алгоритмов, которые будут использоваться.
- `AllowUsers` и `AllowGroups` принимают шаблоны ввода и позволяет контролировать, какие пользователи могут сделать попытку в дальнейшей идентификации.
- `DenyGroups` и `DenyUsers` действуют, как и следовало ожидать, симметрично.
- `PermitRootLogin` позволяет пользователю `root` подключаться через SSH.
- `Protocol` позволяет указать, принимаются ли оба типа протоколов (и, если нет, какой из них).
- `TCPKeepAlive` хороша, если у вас иногда обрываются SSH-соединения. Сообщение "keepalive" рассыпается, чтобы проверить соединения, если опция включена. Но это может вызвать рассоединение, если при маршрутизации происходят редкие ошибки.

## SSH туннелирование

OpenSSH позволяет создать туннель для инкапсулирования других протоколов в шифрованном SSH-канале. Эта возможность включается на сервере `sshd` по умолчанию, но может быть отключена опциями командной строки или конфигурационными файлами. Полагая, что эта возможность включена, клиенты могут легко эмулировать любой порт/протокол, который пожелаю, и использовать его для соединения. Например, для создания туннеля для `telnet`:

### Listing 12. Прокладка туннеля для telnet

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
% telnet localhost 5023
```

Конечно же, этот пример бессмысленный, поскольку командная оболочка SSH делает то же в качестве оболочки `shell`. Однако можно создавать соединения POP3, HTTP, SMTP, FTP, X11, и по любому другому протоколу аналогичным образом. Основная идея состоит в том, что определенные порты локального хоста ведут себя так будто это есть удаленная служба с реальными коммуникационными пакетами, гуляющими в SSH соединении в зашифрованном виде.

Опции, которые мы использовали в примере, таковы:

- `-2` (использовать протокол 2),

- -N (нет команды/только туннель),
- -f (SSH фоновым), и
- -L, (описать туннель как "localport:remotehost:remoteport").

Также указывается сервер и имя пользователя.

## TCP-wrappers

### Что такое "TCP-wrappers"?

Первое, что следует уяснить про TCP-wrappers -- это то, что их *не надо* использовать, и они не являются активно развивающимися. Однако вы можете обнаружить, что демон `tcpd` из TCP-wrappers все еще выполняется в вашей системе. В свое время это было хорошим приложением, но теперь его функциональность сильно упала по сравнению с `iptables` и другими приложениями. Главной целью TCP-wrappers осталось отслеживание и фильтрация входящих запросов от SYSTAT, FINGER, FTP, TELNET, RLOGIN, RSH, EXEC, TFTP, TALK, и других сетевых служб.

TCP-wrappers можно настроить двумя путями. Один состоит в замене других служб на `tcpd`, обеспечивая аргументы, для передачи контроля определённому приложению после того, как `tcpd` выполнил свое журналирование и фильтрацию. Другой метод оставляет сетевых демонов в покое, но изменяет конфигурационный файл `inetd`. Например, такая строка:

```
tftp dgram udp wait root /usr/etc/tcpd in.tftpd -s /tftpboot
```

приведет к тому, что входящий запрос `tftp` запустится через оберточную программу (`tcpd`) под именем процесса `in.tftpd`.

## Задачи по безопасности

Чтобы выполнить поставленную цель - изучить методы обеспечения безопасности - надо изучить огромное количество вопросов (и каждый из них может оказаться принципиально важным). Поэтому я не надеюсь полностью раскрыть эту тему на страницах небольшого учебника. Всвязи с этим я рекомендую уделить время знакомству с ресурсами и утилитами, перечисленными в этом разделе.

web-сайты, на которые стоит заглянуть за статьями по безопасности и патчами:

- [Security focus news](#): The Security Focus web-сайт -- один из лучших сайтов по докладам и обсуждению проблем безопасности и конкретных уязвимостей. Сайт содержит несколько новостных групп и объявлений, на которые можно подписаться, равно как и статьи общего характера и сообщения об ошибках, по которым можно выполнять поиск.
- [The Bugtraq mailing list](#): Обширный модерируемый список рассылки для *подробных* обсуждений и объявлений уязвимостей в системах компьютерной безопасности: кто они, как с ними бороться, как их обнаружить.
- [CERT Coordination Center](#): размещенный на сервере Carnegie Mellon University, CERT имеет совещательный коллектив, схожий с Security Focus site, но имеет больший уклон в сторону пособий и руководств. Обозревать несколько таких сайтов -- хороший способ убедиться, что вы в курсе всех проишествий, влияющих на вашу ОС, дистрибутив или определенные программы или серверы.
- [Computer Incident Advisory Capability](#): Информационные бюллетени CIAC распространяются сообществом Department of Energy для извещения сайтов об уязвимостях в компьютерной безопасности и рекомендуемых действиях. Пободно тому, консультативные записки CIAC служат для предупреждения сайтов об опасной, уязвимости, требующей срочного решения, и разрешения этих проблем. Технические бюллетени CIAC охватывают вопросы технической безопасности и анализы, не столь чувствительные ко времени.

Утилиты для слежения за безопасностью, которые стоит запускать, таковы:

- [Open Source Tripwire](#): Утилита, касающаяся безопасности и целостности данных, отслеживающая определенные изменения в файлах.
- [scanlogd](#): Утилита, обнаруживающая сканирование TCP портов.
- [Snort](#): Обнаружение и предотвращение вторжения в сеть, с помощью языка правил; использует методы, основанные на подписях, протоколах и аномалиях.

[Перейти к тексту](#)

- [Войти \(или Регистрация\)](#)

- [Русский](#)



- [Технические материалы](#)

- [Пробное ПО](#)

- [Сообщество](#)



- [developerWorks Россия](#)

- [Linux](#)

- [Статьи](#)

## Подготовка к экзамену LPI: Системная безопасность

Средний уровень администрирования (LPIC-2) тема 212

[Дэвид \(David\) Мертц \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** В этом учебном пособии, шестом из [серии, состоящей из семи пособий](#) covering intermediate network administration on Linux®, Дэвид Мертц продолжает готовить вас к сдаче экзамена 202 Администрирования Среднего Уровня (LPIC-2) Профессионального Института Linux. По необходимости, это пособие кратко затрагивает широкий круг вопросов, связанных с Linux с точки зрения безопасности сервера сети, включая общие вопросы маршрутизации, межсетевых экранов, преобразования NAT и соответствующие утилиты. Здесь нашлось место для задания политики безопасности для FTP и SSH; также рассматривается общий контроль доступа с помощью tcpd, hosts.allow, и их товарищем; представлены некоторые основные утилиты отслеживания безопасности, рассказано, где найти ресурсы по безопасности.

### Ресурсы

### Научиться

- Ознакомьтесь с полным [учебным курсом подготовки к экзамену LPI](#) на developerWorks, чтобы изучить основы Linux и подготовиться к аттестации

системного администратора.

- [\*TCP/IP Network Administration, Третье издание\*](#) by Craig Hunt (O'Reilly, April 2002), Крейг Хант (O'Reilly, April 2002) является отличным ресурсом по сетям Linux.
- Для углубленного изучения, [Linux Documentation Project](#) располагает множеством различных полезных документов, в особенности HOWTO.
- На [developerWorks Linux zone](#), найдите больше resources для разработчиков Linux.
- Оставайтесь в курсе [технических событий developerWorks и Webcasts](#).

#### **Получить продукты и технологии**

- Вместе с [тестовое ПО IBM](#), доступном для скачивания непосредственно с developerWorks, постройте ваш следующий проект на Linux.

# Учебник для экзамена LPI 202: Устранение проблем в сети

*Администрирование для специалистов (LPIC-2) тема 214*

[Дэвид \(David\) Мертз \(Mertz\)](#), Developer, Gnosis Software, Inc.

**Описание:** В этом учебнике, последнем в [серии семи учебников](#) посвященных администрированию сетей в Linux® для специалистов, Дэвид Мертз (David Mertz) заканчивает вашу подготовку к экзамену LPI 202 Профессионального Института Linux (Linux Professional Institute) Администрирование для специалистов (LPIC-2). Этот учебник вновь обращается к ранним учебникам серии LPI 202, концентрируясь на том как использовать уже известные вам основные инструменты для устранения проблем в сети. Рассматриваемые инструменты поделены на две категории: средства настройки и диагностические средства.

[Больше статей из этой серии](#)

**Дата:** 28.06.2006

**Уровень сложности:** средний

## Файлы настройки сети

### /etc/network/ и /etc/sysconfig/network-scripts/

Каталог /etc/network/ во многих дистрибутивах Linux содержит различные данные для текущей сети, особенно в файле /etc/network/interfaces. Различные утилиты, особенно `ifup` и `ifdown` (или `iwup` и `iwdown` для беспроводных интерфейсов) в некоторых дистрибутивах находятся в /etc/sysconfig/network-scripts/ (но те же скрипты в вашем дистрибутиве могут находиться где-то в другом месте).

### /var/log/syslog и /var/log/messages

Сообщения от ядра или средства `syslogd` записываются в файлы журналов /var/log/syslog и /var/log/messages. [Учебник для экзамена LPI 201 \(тема 211\): Обслуживание системы](#) детально рассматривает журналирование системы. Обычно для проверки журналов (log-файлов) используется утилита `dmesg`.

### /etc/resolv.conf

[Учебник для экзамена LPI 202 \(тема 207\): Служба имен доменов \(Domain Name System\)](#) детально рассматривает /etc/resolv.conf. Вообще говоря, этот файл просто содержит необходимую для нахождения доменных имен серверов. Он может быть настроен вручную или посредством динамического метода типа RIP, DHCP или NIS.

### /etc/hosts

Файл /etc/hosts обычно является тем, что система Linux просматривает в первую очередь при попытке распознать символьное имя хоста. Вы можете добавить в него записи или для пропуска поиска DNS (или иногда средств YP или NIS), или для задания имени хоста, которые не доступны через DNS, часто вследствии того, что они имеют прямые имена в локальной сети. Смотри пример в Листинге 4

### Листинг 4. /etc/hosts, место для разрешения символьных имен хостов

```
$ cat /etc/hosts
# Set some local addresses
```

```
127.0.0.1      localhost
255.255.255.255 broadcasthost
192.168.2.1    artemis.gnosis.lan
192.168.2.2    bacchus.gnosis.lan
# Set undesirable site patterns to loopback
127.0.0.1      *.doubleclick.com
127.0.0.1      *.advertising.com
127.0.0.1      *.valueclick.com
```

## /etc/hostname и /etc/HOSTNAME

Файл /etc/HOSTNAME (на некоторых системах не заглавными буквами) иногда используется для символьного имени локального хоста, известного в сети. Однако, использование этого файла в различных дистрибутивах различается; вообще говоря, /etc/hosts используется только в новых дистрибутивах.

## /etc/hosts.allow и /etc/hosts.deny

[Учебник для экзамена LPI 201 \(тема 209\): Совместное использование файлов и сервисов](#) и [учебник для экзамена LPI 202 \(тема 212\): Безопасность системы](#) рассматривают файлы /etc/hosts.allow и /etc/hosts.deny детально. Эти конфигурационные файлы используются для позитивных и негативных списков контроля доступа различными сетевыми инструментами. Читайте man-страницы этих конфигурационных файлов, чтобы получить больше информации о спецификациях шаблонов, диапазонов и специальных запретов.

Если после начальной настройки безопасности системы соединения нет, хотя кажется, что оно должно работать, стоит начать анализ с проверки содержимого этих файлов. Вообще, проверка элементов управления доступом при анализе причин проблем следует сразу за проверкой основных интерфейсов и информации о маршрутизации. То есть, если вы вообще не можете "достучаться" до некоторого хоста (или он не может обратиться к вам), то не имеет значения, имеет ли хост запрет на использование предоставляемого вами сервиса. Но причиной отдельных сбоев соединения и сервисов обслуживания часто могут быть элементы управления доступом.

## Инструменты конфигурирования сети

### Об исправлении проблем в сети

Для исправления проблем настройки сети, вы должны знать как использовать различные инструменты, описываемые в учебниках данной серии; вы также должны быть знакомы с конфигурационными файлами, определяющими состояние и поведение сети. В этом учебнике приводится краткая сводка основных инструментов и конфигурационных файлов, с которыми вы должны быть знакомы, для эффективных действий по исправлению проблем с сетью.

Для простоты в учебнике инструменты сгруппированы в соответствии с тем используются ли они больше для настройки сети или для диагностики проблем с сетью. Конечно же на практике эти элементы редко отделены друг от друга.

### ifconfig

В [Учебнике для экзамена LPI 202 \(тема 205\): Настройка сети ifconfig](#) рассматривается во всех деталях. Эта утилита может и сообщать о статусе сетевых интерфейсов, и позволяет вам изменять настройки этих интерфейсов. В большинстве случаев, когда сеть ведет себя *как-то* не так -- например, некоторая машина вообще не видна в сети -- первое что следует обычно предпринять это запустить **ifconfig** без параметров. Если она не сможет вывести отчет об

активных интерфейсах, то вы можете быть совершенно уверены, что эта локальная машина имеет проблемы в настройке. "Активные" в данном случае означает, что утилита показывает назначенный IP адрес; в большинстве случаев следует ожидать вывода числа пакетов в RX и TX строках:

### Листинг 1. Использование ifconfig

```
eth0      Link encap:Ethernet HWaddr 00:C0:9F:21:2F:25
          inet addr:192.168.216.90 Bcast:66.98.217.255 Mask:255.255.254.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:6193735 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:6982479 errors:0 dropped:0 overruns:0 carrier:0
```

Попытка активировать интерфейс чем-то вроде **ifconfig eth0 up ...** это хороший первый шаг для определения, что интерфейс *может* быть активирован (во многих случаях используются дополнительные опции командной строки).

### route

[Учебник для экзамена LPI 202 \(тема 205\): Настройка сети](#) подробно описывает **route**. Эта утилита позволяет легко просматривать и изменять таблицы маршрутизации для локальной машины и локальной сети. Используя **route**, вы можете добавлять и удалять маршруты, определять сетевые маски и шлюзы, а также выполнять множество других задач тонкой настройки.

Для большинства случаев, вызов **route** следует производить из инициализационных сценариев, но в процессе работы по диагностике и исправлению проблем может помочь экспериментирование с настройками маршрутизации (затем вы сможете скопировать удачный вариант в соответствующий стартовый сценарий для последующего использования).

### hostname

Эта утилита имеет также псевдонимы для использования в различных аспектах:

- **domainname**
- **nodename**
- **dnsdomainname**
- **nisdomainname**
- **ypdomainname**

Используются эти возможности при помощи ключей для самой **hostname**.

**hostname** используется для установки или отображения текущего хоста, домена или имени узла системы. Эти имена используются многими сетевыми программами для идентификации машины. Имя домена используется также и NIS/YP.

### dmesg

Утилита **dmesg** позволяет обрабатывать журнал сообщений ядра; она работает совместно с **syslogd**. Доступ к любому процессу ядра, включая те, что связаны с сетью, лучше всего осуществляется при помощи утилиты **dmesg**, часто с наложением фильтрации при помощи другого инструмента вроде **grep**, в качестве ключей **dmesg**.

### Ручная установка ARP

Необходимость разбираться с автоматически определенной таблицей ARP записей практически никогда не возникает. Однако, вам может потребоваться вручную изменить кэш

ARP, чтобы проверить какие-то предположения об ошибках в сети. Утилита `arp` позволяет сделать вам это. Ключевыми опциями флагами для утилиты `arp` являются `-d` для удаления, `-S` для установки, и `-f` для установки из файла (стандартным файлом является `/etc/ethers`).

Например, предположим, что связь с некоторым IP адресом в локальной сети неустойчива или ненадежна. Одной из возможных причин такой ситуации являются несколько машин, настроенных некорректно и использующих один и тот же IP адрес. Когда ARP запрос отправляется по сети Ethernet, то нет определенности, какая машина первой отзовется ARP ответом. Конечным результатом может быть то, что пакеты данных могут доставляться то к одной, то к другой машине.

Первым шагом является использование `arp -n` для проверки имеющихся назначений IP. Если вы обнаружите, что рассматриваемый IP адрес не соответствует корректному Ethernet устройству, то это будет четким сигналом о том, что происходит.

Чтобы преодолеть это возникающий в таком случае разнобой в соответствии IP и сетевого адреса, вы можете насильно указать для проблемного IP его ARP адрес, используя опции `arp -S` (или `-f`). Ручная настройка присвоения адресов будет действовать постоянно, если только специально не использован флаг `temp`. Если ручное ARP присвоение адреса устраняет проблему потери данных, то это четко показывает, что проблема в назначении IP адресов.

## Инструменты диагностики сети

### `netstat`

[Учебник для экзамена LPI 202 \(тема 205\): Настройка сети](#) рассматривает `netstat` детально. Эта утилита отображает различную информацию о сетевых соединениях, таблицах маршрутизации, статистику интерфейсов, имитационные соединения и участие в группах. Кроме этого, `netstat` предоставляет весьма детализированную статистику о пакетах, обработанных различными способами.

Man-страница `netstat` предоставляет информацию о большом количестве доступных ключей и опций. Эта утилита является хорошим инструментом общего назначения для углубления в детали состояния сети на конкретной машине.

### `ping`

Хорошей стартовой точкой для проверки возможности подключения к некоторому узлу с данной машины (по IP адресу или символьному имени) является утилита `ping`. Наряду с определением существует ли маршрут как таковой -- включая разрешение имён через DNS или другим способом при использовании символьного имени -- `ping` предоставляет вам информацию о времени отклика, что может служить индикатором перегрузки сети или задержек маршрутизации. Иногда `ping` может отображать процент потерянных пакетов, но при практическом применении вы почти всегда будете видеть или 100, или 0 процентов потерянных пакетов для запросов `ping`.

### `traceroute`

Утилита `traceroute` немного напоминает `ping` на стероидах. Вместо того, чтобы просто сообщать о факте наличия маршрута к указанному хосту, `traceroute` сообщает полную информацию о всех переходах, выполненных при прохождении пути, включая время для каждого маршрутизатора. Маршруты с течением времени могут меняться или вследствие динамических изменений в сети Интернет, или вследствии изменений маршрутизации сделанных вами локально. Тем не менее в данный момент времени `traceroute` показывает вам действительный путь следования.

## Листинг 2. traceroute показывает действительный путь следования

```
$ traceroute google.com
traceroute: Warning: google.com has multiple addresses; using 64.233.187.99
traceroute to google.com (64.233.187.99), 30 hops max, 38 byte packets
 1  ev1s-66-98-216-1.ev1servers.net (66.98.216.1)  0.466 ms  0.424 ms  0.323 ms
 2  ivhou-207-218-245-3.ev1.net (207.218.245.3)  0.650 ms  0.452 ms  0.491 ms
 3  ivhou-207-218-223-9.ev1.net (207.218.223.9)  0.497 ms  0.467 ms  0.490 ms
 4  gateway.mfn.com (216.200.251.25)  36.487 ms  1.277 ms  1.156 ms
 5  so-5-0-0.mpri1.atl6.us.above.net (64.125.29.65)  13.824 ms  14.073 ms  13.826 ms
 6  64.124.229.173.google.com (64.124.229.173)  13.786 ms  13.940 ms  14.019 ms
 7  72.14.236.175 (72.14.236.175)  14.783 ms  14.749 ms  14.476 ms
 8  216.239.49.226 (216.239.49.226)  16.651 ms  16.421 ms  17.648 ms
 9  64.233.187.99 (64.233.187.99)  14.816 ms  14.913 ms  14.775 ms
```

## host, nslookup и dig

Все три утилиты -- [host](#), [nslookup](#) и [dig](#) -- используются для опроса записей DNS; большинство их функций перекрываются. Вообще говоря, [nslookup](#) является улучшенной версией [host](#), а [dig](#) в свою очередь улучшенным [nslookup](#) (хотя ни одна из трех не имеет полной совместимости сверху или снизу с другими). Все эти инструменты зависят от одних и тех же средств ядра, так что выдаваемые результаты должны быть всегда сходны (исключая случай различия детализации). Например, каждая из трех использовалась для опроса google.com:

## Листинг 3. Использование host, nslookup и dig для опроса Google

```
$ host google.com
google.com has address 64.233.187.99
google.com has address 64.233.167.99
google.com has address 72.14.207.99

$ nslookup google.com
Server:          207.218.192.39
Address:         207.218.192.39#53

Non-authoritative answer:
Name:   google.com
Address: 64.233.167.99
Name:   google.com
Address: 72.14.207.99
Name:   google.com
Address: 64.233.187.99

$ dig google.com
; <>>> DiG 9.2.4 <>>> google.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46137
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.           IN      A

;; ANSWER SECTION:
google.com.        295     IN      A      64.233.167.99
google.com.        295     IN      A      72.14.207.99
google.com.        295     IN      A      64.233.187.99
```

```
; ; Query time: 16 msec
; ; SERVER: 207.218.192.39#53(207.218.192.39)
; ; WHEN: Mon Apr 17 01:08:42 2006
; ; MSG SIZE rcvd: 76
```

## Файлы настройки сети

### /etc/network/ и /etc/sysconfig/network-scripts/

Каталог /etc/network/ во многих дистрибутивах Linux содержит различные данные для текущей сети, особенно в файле /etc/network/interfaces. Различные утилиты, особенно [ifup](#) и [ifdown](#) (или [iwup](#) и [iwdown](#) для беспроводных интерфейсов) в некоторых дистрибутивах находятся в /etc/sysconfig/network-scripts/ (но те же скрипты в вашем дистрибутиве могут находиться где-то в другом месте).

### /var/log/syslog и /var/log/messages

Сообщения от ядра или средства [syslogd](#) записываются в файлы журналов /var/log/syslog и /var/log/messages. [Учебник для экзамена LPI 201 \(тема 211\): Обслуживание системы](#) детально рассматривает журналирование системы. Обычно для проверки журналов (log-файлов) используется утилита [dmesg](#).

### /etc/resolv.conf

[Учебник для экзамена LPI 202 \(тема 207\): Служба имен доменов \(Domain Name System\)](#) детально рассматривает /etc/resolv.conf. Вообще говоря, этот файл просто содержит необходимую для нахождения доменных имен серверов. Он может быть настроен вручную или посредством динамического метода типа RIP, DHCP или NIS.

### /etc/hosts

Файл /etc/hosts обычно является тем, что система Linux просматривает в первую очередь при попытке распознать символьное имя хоста. Вы можете добавить в него записи или для пропуска поиска DNS (или иногда средств YP или NIS), или для задания имени хоста, которые не доступны через DNS, часто вследствии того, что они имеют прямые имена в локальной сети. Смотри пример в Листинге 4

#### Листинг 4. /etc/hosts, место для разрешения символьных имен хостов

```
$ cat /etc/hosts
# Set some local addresses
127.0.0.1      localhost
255.255.255.255 broadcasthost
192.168.2.1    artemis.gnosis.lan
192.168.2.2    bacchus.gnosis.lan
# Set undesirable site patterns to loopback
127.0.0.1      *.doubleclick.com
127.0.0.1      *.advertising.com
127.0.0.1      *.valueclick.com
```

### /etc/hostname и /etc/HOSTNAME

Файл /etc/HOSTNAME (на некоторых системах не заглавными буквами) иногда используется для символьного имени локального хоста, известного в сети. Однако, использование этого файла в различных дистрибутивах различается; вообще говоря, /etc/hosts используется только

в новых дистрибутивах.

### /etc/hosts.allow и /etc/hosts.deny

[Учебник для экзамена LPI 201 \(тема 209\): Совместное использование файлов и сервисов](#) и [учебник для экзамена LPI 202 \(тема 212\): Безопасность системы](#) рассматривают файлы /etc/hosts.allow и /etc/hosts.deny детально. Эти конфигурационные файлы используются для позитивных и негативных списков контроля доступа различными сетевыми инструментами. Читайте man-страницы этих конфигурационных файлов, чтобы получить больше информации о спецификациях шаблонов, диапазонов и специальных запретов.

Если после начальной настройки безопасности системы соединения нет, хотя кажется, что оно должно работать, стоит начать анализ с проверки содержимого этих файлов. Вообще, проверка элементов управления доступом при анализе причин проблем следует сразу за проверкой основных интерфейсов и информации о маршрутизации. То есть, если вы вообще не можете "достучаться" до некоторого хоста (или он не может обратиться к вам), то не имеет значения, имеет ли хост запрет на использование предоставляемого вами сервиса. Но причиной отдельных сбоев соединения и сервисов обслуживания часто могут быть элементы управления доступом.

## Заключение

### Используйте преимущества каждого ресурса

Вероятно лучшим ресурсом дополнительной информации по темам рассмотренным в данном учебнике является вся эта серия уроков. Практически все затронутые здесь темы были детально описаны в предыдущих учебниках.

Очень мало тех, кто создает руководства пошагового исправления не работающей сети в Linux. Одним из них, и весьма приличным, является "[Simple Network Troubleshooting \[Простое исправление проблем в сети\]](#)." Подобное быстрое руководство от Debian "[How To Set Up A Linux Network \[Как настроить сеть в Linux\]](#)." Поскольку учебники появляются, исчезают и обновляются по различным схемам, по мере изменения дистрибутивов и команд, вы всегда можете использовать поиск в Интернет, чтобы найти ресурс доступный в настоящее время.

## Ресурсы

### Научиться

- Оригинал статьи [LPI exam 202 prep, Topic 214: Network troubleshooting](#).
- Просмотрите все [серии учебников к экзаменам LPI](#) на developerWorks для изучения основ Linux и подготовки к сертификации на системного администратора.
- В [программе LPIC](#), вы можете найти списки заданий, примеры вопросов и детальные описания тем трех уровней сертификации системных администраторов Linux от Профессионального Института Linux (Linux Professional Institute).
- [TCP/IP Network Administration, Third Edition \[Администрирование сетей TCP/IP. третье издание\]](#) Крейга Ханта (Craig Hunt) (O'Reilly, Апрель 2002) является превосходным ресурсом по сетям в Linux.
- Более специфичная информация содержится в разнообразных полезных документах проекта [Linux Documentation Project](#), особенно в файлах HOWTO.
- В [Linux-разделе developerWorks](#), вы можете найти еще больше ресурсов для

Linux-разработчиков.

- Будьте в курсе [технических событий и Web-трансляций developerWorks](#).

#### **Получить продукты и технологии**

- Создайте свой следующий программный проект для Linux при помощи [trial-программного обеспечения от IBM](#), доступного для скачивания прямо с developerWorks.

# Подготовка к экзамену LPI 301: Тема 302.

## Установка и разработка

*Профессионал Linux высокого уровня (LPIC-3)*

Шон Волберг, старший сетевой инженер, P.Eng

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux Professional Institute® на квалификацию профессионала Linux высокого уровня (LPIC-3). В этом руководстве, втором из серии из шести руководств, Шон расскажет об установке и настройке сервера LDAP (Lightweight Directory Access Protocol), а также о написании сценариев Perl для доступа к данным каталога. Прочитав руководство, вы узнаете о программировании, установке и настройке сервера LDAP.

**Дата:** 11.03.2008

**Уровень сложности:** средний

### Предисловие

Узнайте, чему могут научить вас эти руководства и как получить от них больше пользы.

### Об этой серии руководств

Институт [Linux Professional Institute](#) (LPI) сертифицирует системных администраторов Linux по трём уровням: *младший уровень* (также называемый "уровень сертификации 1"), *углубленный уровень* (также называемый "уровень сертификации 2") и *высший уровень* (также называемый "уровень сертификации 3"). Для того чтобы получить сертификацию на уровне 1, нужно сдать экзамены 101 и 102. Для того чтобы получить сертификацию на уровне 2, нужно сдать экзамены 201 и 202. Для того чтобы получить сертификацию на уровне 3, у вас должна быть действующая сертификация на углубленном уровне и сдан экзамен 301 ("основной"). Кроме того, на высоком уровне вы можете сдать дополнительные специализированные экзамены.

developerWorks предлагает руководства, которые помогут вам подготовиться к пяти экзаменам для младшего, углубленного и высокого уровня. В каждом экзамене охватывается несколько тем, и для каждой темы на developerWorks есть соответствующий учебник для самостоятельного изучения. В таблице 1 перечислены шесть тем и соответствующие им руководства developerWorks для экзамена LPI 301.

**Таблица 1. Экзамен LPI 301: руководства и темы**

Тема экзамена LPI 301	Руководство developerWorks	Краткое описание руководства
Тема 301	Подготовка к экзамену LPI 301: <a href="#">понятия, архитектура и модель</a>	Узнайте о понятиях и архитектуре LDAP, о том, как проектировать и внедрять каталог LDAP, а также о схемах.
Тема 302	Подготовка к экзамену LPI 301: установка и разработка	(Это руководство) Узнайте, как устанавливать, настраивать и использовать программное обеспечение OpenLDAP. См. подробные <a href="#">цели</a> .
Тема 303	Подготовка к экзамену	появится в ближайшее время.

	LPI 301: конфигурирование	
Тема 304	Подготовка к экзамену LPI 301: использование	появится в ближайшее время.
Тема 305	Подготовка к экзамену LPI 301: интеграция и миграция	появится в ближайшее время.
Тема 306	Подготовка к экзамену LPI 301: планирование пропускной способности	появится в ближайшее время.

Для того чтобы сдать экзамен 301 (и получить сертификацию третьего уровня), вы должны:

- обладать несколькими годами опыта установки и поддержки Linux на большом числе компьютеров, используемых в различных целях
- обладать опытом интеграции с различными технологиями и операционными системами
- обладать профессиональным опытом или пройти профессиональную подготовку специалиста Linux корпоративного уровня (включая опыт, полученный при работе в другой роли)
- знать администрирование Linux на углубленном и высоком уровне, включая установку, управление, обеспечение безопасности, решение возникающих проблем и техническое обслуживание.
- уметь использовать инструменты с открытым исходным кодом для проведения измерений, необходимых для планирования пропускной способности и решения проблем с ресурсами
- иметь профессиональный опыт применения LDAP для интеграции с сервисами UNIX® и Microsoft® Windows®, в том числе Samba, Pluggable Authentication Modules (PAM), электронной почты и Active Directory
- уметь планировать, проектировать, разрабатывать, строить и реализовывать полную среду с использованием Samba и LDAP, а также проводить измерения для планирования производительности и оценки безопасности служб
- уметь создавать сценарии на Bash или Perl или знать как минимум один язык системного программирования (например, C)

Институт Linux Professional Institute не дает рекомендаций по каким-либо конкретным материалам и методикам для подготовки к экзаменам, разработанным сторонними лицами.

## Об этом руководстве

Добро пожаловать во второе из шести руководств, призванных помочь вам подготовиться к сдаче экзамена LPI 301, - "Установка и разработка". Из этого руководства вы узнаете об установке и настройке сервера LDAP, а также о том, как использовать Perl для доступа к установленному серверу LDAP.

Это руководство организовано в соответствии с целями LPI по этой теме. Условно говоря, чем выше вес цели, тем больше вопросов по этой теме будет на экзамене.

## Цели

В таблице 2 подробно показаны цели этого руководства.

**Таблица 2. Установка и разработка: цели экзамена, описанные в этом руководстве**

Цель экзамена LPI	Вес цели	Краткое описание цели
302.1 <a href="#"><u>Компиляция и установка OpenLDAP</u></a>	3	Компиляция и установка OpenLDAP из исходного кода и из пакетов
302.2 <a href="#"><u>Разработка для LDAP с применением Perl/C++</u></a>	1	Написание простых сценариев Perl для взаимодействия с каталогом LDAP

#### Необходимые условия

Чтобы получить максимум от этого руководства, вы должны обладать глубокими знаниями Linux и иметь работающую Linux-систему, на которой вы сможете практиковаться в выполнении описываемых команд.

Если ваши базовые знания Linux немного устарели, вы можете сначала познакомиться [с руководствами для экзаменов LPIC-1 и LPIC-2](#).

Различные версии программ могут выводить данные в различных форматах, поэтому результаты, полученные вами, могут отличаться от листингов и рисунков, приведенных в этом руководстве.

#### Требования к системе

Для того чтобы выполнить примеры, приведенные в этом руководстве, вам потребуется рабочая станция под Linux с пакетом OpenLDAP и поддержкой PAM. Большинство современных дистрибутивов удовлетворяют этим требованиям.

## Подготовка к экзамену LPI 301: Тема 302. Установка и разработка

### Компиляция и установка OpenLDAP

В этом разделе описывается материал по теме 302.1 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 3.

Из этого раздела вы узнаете, как:

- Компилировать и настраивать OpenLDAP из исходного кода
- Узнать о базах данных бэк-энда OpenLDAP
- Управлять демонами OpenLDAP
- Устранять ошибки, возникающие во время установки

OpenLDAP - это приложение с открытым исходным кодом, реализующее сервер LDAP и связанные с ним инструменты. Поскольку это приложение с открытым исходным кодом, вы можете загрузить его исходный код бесплатно. Проект OpenLDAP не распространяет двоичный код, но большинство основных дистрибутивов делают пакеты самостоятельно. Из этого руководства вы узнаете, как устанавливать OpenLDAP и из исходного кода, и из пакетов.

## Компиляция из исходного кода

Сначала нужно загрузить последнюю версию OpenLDAP с сайта проекта (ссылку для скачивания можно найти в разделе [Ресурсы](#)). У этого проекта обычно есть две актуальные версии: одна стабильная и одна тестовая. Это руководство было написано с использованием стабильных версий 2.3.30 и 2.3.38. Если вы будете выполнять приведенные здесь примеры, обратите внимание, что названия директорий могут различаться в зависимости от используемой вами версии.

Чтобы извлечь исходный код из загруженного архива, введите: `tar -xzf openldap-stable-20070831.tgz`. Эта команда разархивирует загруженные файлы в папку. Войдите в эту новую папку с помощью команды `cd openldap-2.3.38` (подставив свою версию OpenLDAP).

Теперь вы находитесь в директории с исходным кодом. Теперь вам нужно настроить и собрать среду для вашей системы, после чего собрать программное обеспечение. Для выполнения этих действий в OpenLDAP используется сценарий `configure`. Введите `./configure --help` для просмотра доступных параметров. Некоторые из них определяют, куда будут устанавливаться файлы (например, `--prefix`), другие определяют функции OpenLDAP, которые будут доступны в сборке. В листинге 1 перечислены функции и значения, присваиваемые по умолчанию.

### Листинг 1. Параметры настройки, относящиеся к функциям OpenLDAP

SLAPD (Standalone LDAP Daemon) Options:

<code>--enable-slapd</code>	enable building slapd [yes]
<code>--enable-aci</code>	enable per-object ACIs (experimental) [no]
<code>--enable-cleartext</code>	enable cleartext passwords [yes]
<code>--enable-crypt</code>	enable crypt(3) passwords [no]
<code>--enable-lmpasswd</code>	enable LAN Manager passwords [no]
<code>--enable-spasswd</code>	enable (Cyrus) SASL password verification [no]
<code>--enable-modules</code>	enable dynamic module support [no]
<code>--enable-rewrite</code>	enable DN rewriting in back-ldap and rwm overlay [auto]
<code>--enable-rlookups</code>	enable reverse lookups of client hostnames [no]
<code>--enable-slapi</code>	enable SLAPI support (experimental) [no]
<code>--enable-slp</code>	enable SLPv2 support [no]
<code>--enable-wrappers</code>	enable tcp wrapper support [no]

SLAPD Backend Options:

<code>--enable-backends</code>	enable all available backends no yes mod
<code>--enable-bdb</code>	enable Berkeley DB backend no yes mod [yes]
<code>--enable-dnssrv</code>	enable dnssrv backend no yes mod [no]
<code>--enable-hdb</code>	enable Hierarchical DB backend no yes mod [yes]
<code>--enable-ldap</code>	enable ldap backend no yes mod [no]
<code>--enable-lmdb</code>	enable ldbm backend no yes mod [no]
<code>--enable-lmdb-api</code>	use LDBM API auto berkeley bcompat mdbm gdbm [auto]
<code>--enable-lmdb-type</code>	use LDBM type auto btree hash [auto]
<code>--enable-meta</code>	enable metadirectory backend no yes mod [no]
<code>--enable-monitor</code>	enable monitor backend no yes mod [yes]
<code>--enable-null</code>	enable null backend no yes mod [no]
<code>--enable-passwd</code>	enable passwd backend no yes mod [no]
<code>--enable-perl</code>	enable perl backend no yes mod [no]
<code>--enable-relay</code>	enable relay backend no yes mod [yes]
<code>--enable-shell</code>	enable shell backend no yes mod [no]
<code>--enable-sql</code>	enable sql backend no yes mod [no]

SLAPD Overlay Options:

<code>--enable-overlays</code>	enable all available overlays no yes mod
<code>--enable-accesslog</code>	In-Directory Access Logging overlay no yes mod [no]
<code>--enable-auditlog</code>	Audit Logging overlay no yes mod [no]

```

--enable-denyop          Deny Operation overlay no|yes|mod [no]
--enable-dyngroup        Dynamic Group overlay no|yes|mod [no]
--enable-dynlist          Dynamic List overlay no|yes|mod [no]
--enable-lastmod         Last Modification overlay no|yes|mod [no]
--enable-ppolicy          Password Policy overlay no|yes|mod [no]
--enable-proxycache      Proxy Cache overlay no|yes|mod [no]
--enable-refint           Referential Integrity overlay no|yes|mod [no]
--enable-retcode          Return Code testing overlay no|yes|mod [no]
--enable-rwm               Rewrite/Remap overlay no|yes|mod [no]
--enable-syncprov         Syncrepl Provider overlay no|yes|mod [yes]
--enable-translucent      Translucent Proxy overlay no|yes|mod [no]
--enable-unique            Attribute Uniqueness overlay no|yes|mod [no]
--enable-valsort           Value Sorting overlay no|yes|mod [no]

SLURPD (Replication Daemon) Options:
--enable-slurpd          enable building slurpd [auto]

Optional Packages:
--with-PACKAGE[=ARG]      use PACKAGE [ARG=yes]
--without-PACKAGE         do not use PACKAGE (same as --with-PACKAGE=no)
--with-subdir=DIR          change default subdirectory used for installs
--with-cyrus-sasl          with Cyrus SASL support [auto]
--with-fetch                with fetch(3) URL support [auto]
--with-threads              with threads [auto]
--with-tls                  with TLS/SSL support [auto]
--with-yielding-select     with implicitly yielding select [auto]
--with-odbc                 with specific ODBC support iodbc|unixodbc|auto [auto]

--with-gnu-ld               assume the C compiler uses GNU ld [default=no]
--with-pic                  try to use only PIC/non-PIC objects [default=use both]
--with-tags[=TAGS]          include additional configurations [automatic]

```

В листинге 1 вы можете видеть, что по умолчанию многие функции выключены, например, метакаталоги и модули. Кроме того, многие параметры отмечены как "auto", что означает, что функция включается, если в системе установлены нужные библиотеки. Вместо того чтобы полагаться на такое автоматическое поведение, лучше составить список нужных функций и включить их. Если каких-то библиотек не хватает, то вы получите ошибку во время компиляции, а не когда-либо позже.

Некоторым конфигурационным опциям могут быть переданы значения `no`, `yes` или `mod`. `no` отключает опцию, `yes` статически привязывает опцию к конечному двоичному файлу, а `mod` собирает опцию в отдельной библиотеке общего пользования. Библиотеки общего пользования загружаются на сервер во время работы (см. приведенный ниже раздел "[Параметры сервера \(глобальные\)](#)"). По умолчанию модули привязываются статически; то есть они являются частью двоичного кода и неотделимы от него. Если вы хотите использовать динамические модули, вам также нужно использовать параметр `--enable-modules`. Преимущество динамических модулей состоит в том, что вы можете проверить различные опции, не раздувая чрезмерно двоичный файл, и поставлять модули отдельно.

В листинге 2 показана строка конфигурации, применяемая в Fedora 7, которая включает многие полезные функции. В большинстве своём выбранные опции позволяют использовать функции, которые будут нужны в следующих руководствах, например `--enable-slurpd` и `--enable-multimaster` для репликации и `--enable-meta` для метакаталогов. Другие

опции включают различные бэк-энды, например, ldab, bdb, null и monitor.

### Листинг 2. Пример конфигурации сборки

```
./configure --enable-plugins --enable-modules --enable-slapd --enable-slurpd \
--enable-multimaster --enable-bdb --enable-hdb --enable-ldap --enable-lmdb \
--enable-lmdb-api=berkeley --enable-meta --enable-monitor --enable-null \
--enable-shell --enable-sql=mod --disable-perl \
--with-kerberos=k5only --enable-overlays=mod --prefix=/tmp/openldap
```

В листинге 2 включаются дополнительные модули и несколько бэк-эндов, включая бэк-энды, построенные на языке SQL и файлы базы данных Berkeley Database. Бэк-энды являются способом хранения и получения данных, используемых OpenLDAP; более подробно они рассматриваются ниже в этом руководстве, в разделе "[Бэк-энды и базы данных](#)".

В листинге 2 также собираются демон автономной работы **slapd** и демон репликации **slurpd**. Также в целях тестирования включаются оверлеи, которые облегчают настройку данных бэк-энда. Поскольку установка является тестовой, префикс изменен на **/tmp/openldap**, поэтому результирующий двоичный код будет расположен в папке **/tmp/openldap/libexec**.

При выполнении сценарий **configure** проверяет наличие необходимых библиотек и создаёт среду сборки. Если выполнение **configure** завершается успешно, скомпилируйте OpenLDAP, выполнив **make depend; make**.

После завершения успешной компиляции кода вы можете установить OpenLDAP, выполнив команду **make install**. Она скопирует весь двоичный код, страницы man и библиотеки в соответствующие места в **/tmp/openldap**.

### Установка из пакетов

Если предыдущий раздел о компиляции из исходного кода напугал вас, вы не одиноки. Компиляция из исходного кода - процесс трудоёмкий и может быть осложнён отсутствием нужных библиотек. Если у вас недостаточно большой опыт разработки на C, вам может быть сложно интерпретировать ошибки, появляющиеся в процессе сборки. К счастью, в большинстве дистрибутивов пакет OpenLDAP поставляется в виде готового двоичного кода с предварительно настроенной конфигурацией. Обычно в этих пакетах есть все необходимые функции.

### Дистрибутивы, построенные на RPM

Для загрузки с сервера и установки пакетов RedHat (RPM) в системах Fedora и CentOS используется инструмент **yum**. Команда **yum list** позволит узнать, какие пакеты доступны, необязательное регулярное выражение позволит отфильтровать список пакетов. В листинге 3 показан поиск всех пакетов, содержащих слово **openldap**.

### Листинг 3. Определение доступных пакетов с помощью **yum**

```
# yum list \*openldap\*
Loading "installonlyn" plugin
Setting up repositories
Reading repository metadata in from local files
Installed Packages
openldap.i386                  2.3.30-2.fc6          installed
openldap-clients.i386            2.3.30-2.fc6          installed
openldap-devel.i386              2.3.30-2.fc6          installed
openldap-servers.i386            2.3.30-2.fc6          installed
openldap-servers-sql.i386        2.3.30-2.fc6          installed
```

Available Packages	
compat-openldap.i386	2.3.30_2.229-2.fc6
	updates

В таких больших приложениях, как OpenLDAP, клиентская и серверная часть обычно разбиваются на два отдельных пакета. Также вы можете найти несколько библиотек, обеспечивающих совместимость (чтобы могли работать приложения, связанные со значительно более старыми версиями данного программного обеспечения). Чтобы установить пакет, выполните команду `yum install`, указав название пакета, например, `yum install openldap-clients openldap-servers`; в результате будут загружены и установлены и клиентский, и серверный пакеты со всеми зависимостями.

В Red Hat Enterprise Linux команда поиска пакетов `openldap` будет иметь вид `up2date --showall | grep openldap`. Для установки пакетов укажите в качестве аргумента `up2date` их названия, например, `up2date openldap-clients openldap-servers`.

Для того, чтобы убедиться в том, что сервер OpenLDAP запускается при загрузке системы, выполните команду `chkconfig ldap on`.

### Дистрибутивы, построенные на базе Debian

В дистрибутивах, построенных на базе Debian, например, Ubuntu, для установки пакетов используются инструменты Advanced Packaging (APT). Сначала выполните команду `apt-cache search openldap` для поиска пакетов OpenLDAP, как показано в листинге 4.

#### Листинг 4. Перечень доступных пакетов OpenLDAP в Ubuntu Linux

```
notroot@ubuntu:~$ apt-cache search openldap
libldap2 - OpenLDAP libraries
libldap2-dev - OpenLDAP development libraries
python-ldap - A LDAP interface module for Python. [dummy package]
python-ldap-doc - Documentation for the Python LDAP interface module
python2.4-ldap - A LDAP interface module for Python 2.4
ldap-utils - OpenLDAP utilities
libldap-2.2-7 - OpenLDAP libraries
slapd - OpenLDAP server (slapd)
```

В листинге 4 показано несколько доступных пакетов. Сервер реализован в пакете `slapd`, а все необходимые зависимости будут загружаться в процессе установки. Выполните команду `sudo apt-get install slapd` для установки сервера. Также будет полезно включить пакет `ldap-utils`, содержащий клиент, работающий в командной строке.

### Настройка программного обеспечения

После установки OpenLDAP его необходимо настроить. Для тестирования достаточно указать всего несколько параметров; но в реальных условиях (и на экзамене LPIC 3) вы должны хорошо разбираться во всевозможных параметрах.

Работой OpenLDAP управляет два конфигурационных файла; по умолчанию оба располагаются в папке `/etc/openldap/`. Первый файл, `ldap.conf`, управляет общим поведением клиентов LDAP. Конфигурационный файл для серверов LDAP называется `slapd.conf`. Несмотря на название, в `slapd.conf` также содержится конфигурация для `slurpd`, демона репликации. В этой статье основное внимание уделяется `slapd.conf`, в частности, его разделу, касающемуся демона `slapd`.

Формат slapd.conf очень прост, за одним ключевым словом следует один или несколько аргументов, на которые распространяются следующие условия:

- Ключевое слово должно начинаться в нулевой колонке — перед ним не должно быть пробелов.
- Если в аргументе содержатся пробелы, его необходимо заключить в двойные кавычки ("").
- Если строка начинается с пробела, она считается продолжением предыдущей строки.
- Регистр ключевых слов значения не имеет, но он может быть важен для аргументов, в зависимости от того, какие ключевые слова используются.

Как и в большинстве инструментов UNIX®, символ решетки (#) обозначает комментарий. Всё, что написано после решетки, игнорируется.

Файл slapd.conf разделен на две части: глобальные параметры и параметры базы данных бэк-энда. Хотя порядок указания директив жёстко не задаётся, выбирать место для новых директив следует с осторожностью, поскольку некоторые из них могут изменять порядок обработки последующих директив. Например, если было указано ключевое слово **backend** или **database**, параметр считается глобальным. После того, как будет прочтена директива **database**, все остальные параметры будут относиться к базе данных. Это будет продолжаться до тех пор, пока не будет считана еще одна директива **database**, после которой следующие директивы будут применяться к новой базе данных.

Некоторые глобальные параметры будут рассматриваться в следующих руководствах серии 301, например, посвящённых управлению доступом и репликации. Ниже приведено описание наиболее часто используемых директив.

### Параметры сервера (глобальные)

Некоторые параметры сервера ограничивают работу, выполняемую процессом **slapd**, что позволяет предотвратить недостаток ресурсов. **conn\_max\_pending** принимает целое число, определяющее количество анонимных запросов, которые могут находиться в очереди в любой момент времени. Более подробно с механизмом связывания сервера LDAP вы познакомитесь в следующих руководствах серии 301; если говорить просто, вы можете подавать запросы серверу после ввода имени пользователя и пароля (автентифицированный сеанс) или без имени и пароля (анонимный сеанс). Запросы, превышающие предел, установленный **conn\_max\_pending**, будут отбрасываться сервером. Параметр **conn\_max\_pending\_auth** работает так же, как и **conn\_max\_pending**, но ограничивает аутентифицированные сессии.

Параметр **idletimeout** (в секундах) сообщает **slapd**, как долго будут удерживаться неактивные клиенты прежде, чем будут отключены. Если этому параметру установлено значение 0, отключение не производится.

Параметр **sizelimit** ограничивает количество результатов поиска в одном запросе, а **timelimit** ограничивает время, затрачиваемое сервером на поиск. Эти параметры могут принимать целочисленные значения, ключевое слово **unlimited** или более сложные жёсткие и мягкие ограничения. По умолчанию устанавливается мягкое ограничение по времени и по количеству результатов, однако, если клиент запросил большее число строк или большее время, может быть применено жёсткое ограничение. Например, **sizelimit sizesoft=400 size.hard=1000** указывает, что по умолчанию возвращается 400 строк. Клиент может подать запрос, увеличивающий этот предел до 1 000. Такой формат может быть применен к группам пользователей, позволяя одним пользователям или приложениям выполнять большие запросы, а другим - только маленькие

Когда клиент выполняет поиск по дереву, он обычно указывает узел (который называется *базой поиска* или просто *базой*), с которого будет начинаться поиск — в нем будут находиться

базы поиска всех отличительных имен (Distinguished Names, DN) результатов. Это позволяет ускорить поиск (поскольку теперь необходимо просматривать меньшее количество узлов) и упростить реализацию клиента (поскольку поиск только по части дерева - простой, но эффективный фильтр). Если клиент не указывает конкретную базу, используется значение **defaultsearchbase**. Рекомендуется устанавливать значение этого параметра, чтобы избежать возможных проблем с неверно настроенными клиентами. В зависимости от структуры используемого вами дерева LDAP будет уместно использовать либо контейнер пользователя, либо корень дерева. (Деревья и отличительные имена описаны в [предыдущем руководстве](#).)

Различные функции, которые будет поддерживать сервер, например, поддержка ранних версий и требования клиентов к безопасности, определяются тремя командами. Это команды **allow**, **disallow** и **require**. После каждой команды указывается ряд ключевых слов с пробелами, которые включают, отключают функцию или делают её обязательной. Ключевые слова показаны в таблице 3.

**Таблица 3. Ключевые слова, используемые с **allow**, **disallow** и **require****

Команда(ы)	Ключевое слово	Описание	Значение по умолчанию
allow	bind_v2	Установка функции позволяет подключаться клиентам, использующим устаревшую версию протокола LDAPv2. В документации по OpenLDAP постоянно подчёркивается, что OpenLDAP не предоставляет полной поддержки LDAPv2, поэтому такие запросы могут привести к непредвиденным результатам.	Запрещено
allow	bind_anon_cred	Позволяет клиенту связывание с паролем, но без DN. Если включен этот параметр, клиент может выполнять анонимное связывание.	Запрещено
allow	bind_anon_dn	Позволяет клиенту выполнять связывание с DN, но без пароля, обычно это связано с неправильной конфигурацией клиента. Если включен этот параметр, клиент может выполнять анонимное связывание.	Запрещено
allow, disallow	update_anon	Позволяет выполнять связывание анонимно, что происходит, когда клиент подключается к серверу LDAP без DN и пароля.	Разрешено
disallow	bind_simple	Позволяет выполнять простую аутентификацию (передача имени пользователя и пароля в незашифрованном виде) вместо более защищенных методов, например Simple Authentication and Security Layer (SASL).	Разрешено
require	bind	Обязывает всех клиентов связываться с каталогом с помощью операции <b>bind</b> перед выполнением каждой операции.	Не требуется
require	LDAPv3	Определяет необходимость использования протокола LDAPv3. Обратите внимание, что эта команда может конфликтовать с <b>allow bind_v2</b> .	Не требуется
require	authc	Требует аутентификации, в противоположность анонимному связыванию.	Не требуется

<code>require SASL</code>	Обязывает использовать метод SASL для подключения к серверу	Не требуется
<code>require strong</code>	Обязывает использовать безопасный метод аутентификации. Это может быть либо SASL, либо простая аутентификация по защищённому каналу.	Не требуется
<code>require none</code>	Эта опция снимает все требования, обычно используется в случае, если нужно ослабить требования определенной базы данных, указывая эту команду в разделе базы данных файла slapd.conf. Если нужно не очистить список требований для базы данных, а изменить их, необходимо указать <code>none</code> перед добавлением новых требований, даже если эти требования уже указаны в глобальном разделе.	Не применимо

Несмотря на то, что некоторые команды из таблицы 3 определяют тип входа в систему, соединение всё ещё подчиняется правилам управления доступом. Например, при анонимном входе к части дерева может быть предоставлен доступ только для чтения. Включение или выключение различных методов аутентификации определяется природой вашего приложения и возможностями ваших клиентов.

Если вы хотите поддерживать высокий уровень готовности, включите `gentlehup`. При включении этой команды `slapd` перестаёт прослушивать сеть после получения сигнала `SIGHUP`, но не отключает открытые соединения. После этого может быть запущен новый экземпляр `slapd`, как правило, с обновлённой конфигурацией.

Для того, чтобы получить более подробный журнал, измените значение `loglevel`. В качестве параметра этой команды указывается целое число, несколько целых чисел или ряд ключевых слов, которые включают ведение журнала для определенной функции. Полный список ключевых слов и значений можно найти в страницах справки slapd.conf. Например, отслеживанию соединения соответствуют значение 8 и ключевое слово `conns`, а синхронизации - значение 4096 и ключевое слово `sync`. Для того, чтобы включить ведение журнала по двум этим позициям, нужно выполнить `logging 5004`, `logging 8 4096` или `logging conns sync`, результат будет одинаковым.

Если вы скомпилировали OpenLDAP из исходного кода, вы, возможно, уже включили некоторые модули. Также, возможно, вы загрузили дополнительные модули из менеджера пакетов, например, пакет `openldap-server-sql` содержит модуль бэк-энда SQL.

Параметры `modulepath` и `moduleload` используются для загрузки динамических модулей в `slapd`. `modulepath` указывает директорию (или перечень директорий), содержащих общие библиотеки, и каждый экземпляр `moduleload` указывает загружаемый модуль. Версию модуля и расширение указывать не нужно, поскольку `slapd` ищет общие библиотеки. Например, для библиотеки `back_sql-2.3.so.0.2.18` нужно указать `moduleload back_sql`. Также можно указать в `moduleload` полный путь к библиотеке (без версии и расширения), например `moduleload /usr/share/openldap/back_sql`.

Некоторым сценариям нужно, чтобы идентификатор процесса хранился в определенном файле. `pidfile` говорит `slapd`, куда записывать свой идентификатор процесса.

## Параметры схемы

Несколько команд позволяют вам добавить к дереву элементы схемы, либо посредством включения файла схемы, либо путём определения объектов в slapd.conf. Как вы помните из [предыдущего руководства](#), схема предоставляет классы атрибутов и объектов, которые могут использоваться деревом LDAP.

Для того, чтобы добавить новый файл схемы на сервер, используйте команду `include`, после которой укажите полный путь к файлу схемы (обычно они располагаются в `/etc/openldap/schema`). Если одна схема ссылается на другую (например, `inetOrgPerson` наследуется из `organizationalPerson`), нужно указать все необходимые файлы в правильном порядке, сначала включая базовые объекты. OpenLDAP обрабатывает файлы схемы в порядке их включения, поэтому этот порядок важен.

Новые элементы схемы можно добавить напрямую через `slapd.conf` с помощью команд `attributetype` и `objectclass` для классов атрибутов и объектов соответственно. Этот способ аналогичен указанию информации в файле схемы и включению его с помощью команды `include`. Подобным же образом можно определить и идентификаторы объектов (OID), используя `objectidentifier`.

## Бэк-энды и базы данных

Бэк-энды и базы данных - это два отдельных, но очень тесно связанных между собой понятия. База данных представляет собой часть дерева, например, `dc=ertw,dc=com`. Бэк-энд описывает метод, используемый `slapd` для доступа к данным. (Дерево `dc=ertw,dc=com` было первым примером в этой серии.)

Во многих случаях бэк-энд является файлом на диске (в некотором формате, более подробно он будет рассматриваться позже); также он может быть методом для получения данных из другого источника, из базы данных SQL, из DNS, или даже с помощью сценария. Каждая база данных обрабатывается одним бэк-эндом, а один и тот же бэк-энд может использоваться несколькими базами данных.

Как уже было отмечено, `slapd.conf` начинается с глобальных директив. Режим бэк-энда начинается с первого экземпляра директивы `backend`. Все директивы в этом режиме применяются к текущему бэк-энду. Все установленные глобально параметры применяются к бэк-энду, если они не переопределются затем на уровне бэк-энда. Подобным же образом базы данных настраиваются в рамках, определяемым ключевым словом `database`. База данных привязывается к типу бэк-энда и наследует все глобальные параметры и параметры уровня бэк-энда. Вы также можете переопределить все параметры на уровне базы данных.

OpenLDAP разделяет бэк-энды на три типа:

1. Тех, которых хранят данные:
  - `bdb`—Использует механизм баз данных Berkeley (например, Sleepycat, сейчас принадлежит Oracle)
  - `hdb`—Улучшенная версия back-`ldb`, в которой усовершенствована индексация
2. Тех, которых передают данные:
  - `ldap`—Передаёт данные другого сервера LDAP
  - `meta`—Передаёт данные различных серверов LDAP для различных частей дерева
  - `sql`—Возвращает данные из базы SQL
3. Тех, которых создают данные:
  - `Dnssrv`—Возвращает внешние ссылки (referral) LDAP на основании данных в записи DNS SRV
  - `monitor`—Возвращает статистику сервера LDAP
  - `null`—Тестовый модуль, ничего не возвращает
  - `passwd`—Возвращает данные из файла паролей
  - `perl`—Возвращает данные, сформированные сценарием Perl
  - `shell`—Возвращает данные, сформированные сценарием shell

Параметры конфигурации различны для каждого бэк-энда, их можно найти в соответствующих страницах справки (например, `slapd-bdb` для бэк-энда `bdb`).

Базы данных представляют дерево и содержащиеся в нём данные. Примером базы данных является дерево `dc=ertw,dc=com`. Все данные в этом DN будут храниться так же, как если бы они являлись частью одной базы данных. Кроме того, существует возможность хранения `ou=people,dc=ertw,dc=com` в одной базе данных, а все, что располагается под `dc=ertw,dc=com` - в другой. И, наконец, сервер LDAP может обрабатывать несколько деревьев, например, `dc=ertw,dc=com` и `dc=lpi,dc=org`. У каждой базы данных есть собственный способ обработки запросов посредством собственного бэк-энда.

Для начала режима конфигурирования базы данных укажите `database`, а затем - тип базы данных. Чаще всего используется база данных Berkeley, поэтому `database bdb` создаёт базу данных BDB. Следующая команда - `suffix`, которая указывает корень дерева обслуживающей базы данных.

`rootdn` и `rootpw` позволяет вам указывать пользователя со всеми привилегиями (*корневой пользователь*) базы данных. К этому пользователю не применяются правила контроля доступа. `rootdn` должен находиться в рамках указанного суффикса и может содержать или не содержать пароль. Будет использоваться пароль, указанный в `rootpw`. Если этот параметр не указан, будет выполняться поиск записи `rootdn` в дереве и будет выполняться аутентификация по атрибуту `userPassword`. Если корневой пользователь не указан, то заданные правила контроля доступа будут применяться ко всем пользователям.

Если вы указываете `lastmod on`, OpenLDAP хранит несколько скрытых атрибутов (называемых *операционными атрибутами*), например, имя лица, создавшего запись, и когда она была изменена. Некоторые из этих атрибутов необходимы для репликации, поэтому будет разумно оставить параметр `lastmod` включенным (как установлено по умолчанию). Эти операционные атрибуты не показываются клиентам, если их не запросить специально.

Также с помощью команды `restrict` можно ограничить действия, которые можно выполнять с базой данных. В качестве параметров этой команды указываются параметры, соответствующие операциям LDAP, например `add`, `bind`, `compare`, `delete`, `rename` или `search`. Для того, чтобы запретить пользователям удалять узлы дерева, используется команда `restrict delete`. Если в дереве содержатся пользователи, но по каким-то причинам вы не хотите давать им возможность подключения к дереву, используйте `restrict bind`. Кроме того, доступны параметры `read` и `write`, которые не запрещают все действия, а блокируют чтение и запись в дерево соответственно. Также вы можете использовать команду `readonly` и сделать базу данных доступной только для чтения.

Различные элементы дерева могут содержаться в различных базах данных. При правильной настройке OpenLDAP объединяет вместе все части. База данных, содержащая в себе другие базы, называется *главной базой данных*; базы данных, содержащиеся в ней, называются *зависимыми базами данных*. Сначала определите зависимую базу данных и добавьте в соответствующую ей строку команду `subordinate`. После этого определите главную базу данных. В такой конфигурации OpenLDAP может работать с несколькими базами данных как с одной, при этом часть данных будет храниться локально, а другая часть запрашивается из других источников (в особом случае, когда все данные находятся на удалённых серверах LDAP, используется метакаталог). Обратите внимание, что если вы сначала определите главную базу данных, а потом зависимую, вы получите сообщение об ошибке – попытке переопределить часть дерева. В листинге 5 показано дерево `dc=ertw,dc=com`, разбитое на главную и зависимую базы данных.

#### Листинг 5. Настройка главной и зависимой базы данных

```
# Subordinate
database bdb
suffix "ou=people,dc=ertw, dc=com"
rootdn "cn=Sean Walberg,ou=people,dc=ertw,dc=com"
```

```
rootpw mysecret
directory      /var/db/openldap/ertw-com-people
subordinate

# Superior
database bdb
suffix "dc=ertw, dc=com"
rootdn "cn=Sean Walberg,dc=ertw,dc=com"
rootpw mysecret
directory      /var/db/openldap/ertw-com
```

Также обратите внимание, что здесь настраиваются два `rootdn`. Если вы хотите установить пароль, в базе должен быть определен `rootdn`. Для того, чтобы сформировать дерево, нужно назначить вторую учётную запись `root`, которая позволит определить запись `dc=ertw,dc=com`, а первый `root` будет определять организационную единицу для сотрудника и все нижестоящие объекты. После добавления пользователей вы можете входить под различными пользователями, получая доступ ко всему дереву.

Если вы используете бэк-энд `bdb`, вам также нужно указать, где хранятся файлы базы данных, с помощью команды `directory`. Каждый экземпляр базы данных должен быть размещен в отдельной директории.

Настройка новой базы данных очень проста, поскольку достаточно ввести всего несколько команд. Сложности начинаются при попытке настроить бэк-энд; эта тема будет раскрыта в следующем руководстве серии 301.

## Оверлеи

Оверлеями называются расширения базы данных. Добавить функцию базы данных можно не только работая с кодом, но и с помощью оверлея. Например, если нужно записывать в файл журнала все операции записи, достаточно подключить к соответствующей базе данных оверлей `auditlog`.

Оверлеи работают по принципу стека. После настройки базы данных вы указываете одну или несколько баз данных. После этого с помощью команды `overlay`, за которой указывается название оверлея, определяются все нужные оверлеи. У каждого оверлея есть свои параметры.

В случае, если задаётся несколько оверлеев, их запуск производится в порядке, обратном порядку определения. Доступ к базе данных осуществляется только после выполнения всех оверлеев. После того, как данные будут получены из базы, оверлеи запускаются снова в том же порядке, и только после этого `slapd` возвращает данные клиенту.

На каждом шаге оверлеи могут выполнять различные действия - ведение журнала, изменение запроса или ответа, прекращение обработки.

## Разработка для LDAP с применением Perl/C++

В этом разделе описывается материал по теме 302.2 экзамена на професионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 1.

Из этого раздела вы узнаете, как:

- Использовать модуль `Net::LDAP` Perl
- Писать сценарии Perl для связывания, поиска и изменения каталогов
- Разрабатывать программное обеспечение на C/C++

Несмотря на то, что в OpenLDAP реализованы клиенты командной строки, часто бывает полезно использовать информацию LDAP в своих собственных сценариях. Perl - очень популярный язык сценариев. В Perl есть модуль **Net::LDAP**, используемый для подключения к серверу LDAP и работы с ним.

## Первое знакомство

**Net::LDAP** не поставляется с Perl, но он может поставляться вместе с вашим дистрибутивом в качестве пакета. Более подробную информацию о поиске и установке пакетов можно найти в разделе "[Установка из пакетов](#)".

Если пакета **Net::LDAP** в вашем дистрибутиве нет, вы можете загрузить его из сети полного архива Perl (CPAN). Выполните команду `perl -MCPAN -e "install Net::LDAP"` с правами root, она загрузит и установит пакет **Net::LDAP** и все зависимости.

## Использование **Net::LDAP**

Использовать пакет **Net::LDAP** очень просто:

1. Создайте новый объект **Net::LDAP**.
2. Свяжитесь с нужным сервером.
3. Выполните операцию LDAP.

### Создание нового объекта

Как обычно, экземпляр модуля **Net::LDAP** в Perl создаётся с помощью метода `new`. Все остальные операции будут выполняться на этом экземпляре. Для работы `new` требуется как минимум, название сервера, к которому вы хотите подключиться. Например:

```
my $ldap = Net::LDAP->new('localhost') or die "$@";
```

В этом примере с помощью метода `new` создаётся новый объект **Net::LDAP**, и ему передаётся строка `localhost`. Результат записывается в переменную `$ldap`. Если функция не выполнится, программа будет остановлена, и на экран будет выведено сообщение, описывающее проблему. `$@` - это внутренняя переменная Perl, в которой содержится состояние последней операции.

Вы можете продолжать выполнять операции LDAP на новом объекте **Net::LDAP**. Каждая функция возвращает объект **Net::LDAP::Message**, в котором содержатся состояние операции, возможные сообщения об ошибках и данные, полученные от сервера.

### Подключение к дереву

Первая операция, которую необходимо выполнить - подключиться к дереву, или *bind*. В листинге 6 показана операция подключения и соответствующая обработка ошибок.

#### Листинг 6. Код Perl для подключения к дереву

```
my $message = $ldap->bind(
    "cn=Sean Walberg,ou=people,dc=ertw,dc=com",
    password=>"test" );

if ($message->code() != 0) {
    die $message->error();
}
```

Листинг 6 начинается с вызова метода `bind` созданного ранее объекта. Первый параметр функции - DN, под которым вы подключаетесь. Если DN не указывается, будет создано анонимное подключение. Остальные параметры указываются в формате `key=>value`; чаще всего будет использоваться пароль.

Каждый метод `Net::LDAP` возвращает объект `Net::LDAP::Message`, в котором содержится результат работы функции. Код ошибки можно получить с помощью метода `code`. Код 0 означает успешное выполнение, поэтому в листинге 6 работа программы останавливается, если результат отличен от нуля. Обратите внимание, что ошибка извлекается из `$message->error`, а не из `$@`, как в предыдущем примере. Причина в том, что это не ошибка Perl, а внутренняя ошибка `Net::LDAP`.

После успешного подключения можно выполнять любые действия, которые будут выполняться в соответствии с политикой контроля доступа сервера. Для отключения необходимо вызвать метод `unbind`.

### Поиск по дереву

Поиск выполняется с помощью метода `search`. Так же, как и в методе `bind`, необходимо передать некоторые параметры и проверить результат запроса. Однако теперь в возвращаемом объекте содержатся данные и его необходимо обработать. Результат работы `search` будет помещён в объект `Net::LDAP::Search`, который наследует все методы `Net::LDAP::Message` (например, `code` и `error`) и содержит методы, которые помогут вам анализировать данные. В листинге 7 показан поиск по дереву.

#### Листинг 7. Поиск по дереву с помощью `search`

```
$message = $ldap->search(base => "dc=ertw,dc=com", filter=> "(objectClass=*)");
if ($message->code() != 0) {
    print $message->error();
} else {
    foreach my $entry ($message->entries()) {
        print $entry->dn() . ": ";
        print join ", ", $entry->get_value("objectClass");
        print "\n";
    }
}
```

Листинг 7 начинается с вызова метода `search`, которому передаётся два параметра - база и фильтр. База сообщает серверу, с какой точки дерева начинать поиск. Дополнительный параметр `scope` говорит серверу, до каких пор продолжать поиск:

- **base** —Только базовый объект
- **one** —Только объекты, дочерние по отношению к базовому объекту (но не сам базовый объект)
- **sub** —Базовый объект и все его дочерние объекты (используется по умолчанию)

Фильтр представляет собой строку, описывающую интересующие вас объекты. Вы можете выполнять поиск по атрибутам и использовать сложные запросы AND/OR. `objectClass=*` возвращает любой объект.

Результат поиска проверяется и, в случае возникновения проблем, выводится сообщение об ошибке. Поскольку сценарий может исправить ошибку, он просто выводит сообщение и продолжает работу.

Функция `entries` возвращает массив объектов `Net::LDAP::Entry`, в каждом из которых содержится один результат. Сначала выводится DN записи, а затем - все классы объекта. Если же вам нужна текстовая версия всей записи, её выведет метод `dump`.

### Добавление новой записи

Запись к дереву добавляется с помощью метода `add`. Необходимо передать функции DN записи, которую вы добавляете, и её атрибуты. Атрибуты представляют собой массив пар

значений `key =>`. При наличии нескольких экземпляров одного атрибута значение также будет массивом. В листинге 8 показано добавление записи в дерево.

### Листинг 8. Добавление записи с помощью `Net::LDAP`

```
$message = $ldap->add(  
    "cn=Fred Flintstone,ou=people,dc=ertw,dc=com",  
    attr => [  
        cn => "Fred Flintstone",  
        sn => "Flintstone",  
        objectclass => [ "organizationalPerson",  
                         "inetOrgPerson" ],  
    ]  
);  
  
if ($message->code() != 0) {  
    print $message->error();  
}
```

Первый параметр `add` - либо DN, либо объект `Net::LDAP::Entry`. Если передаётся DN, необходимо передать ссылку на массив с помощью метода `attr`. Даже если в качестве ссылки на ассоциативный массив используется формат `key => значение`, `Net::LDAP` ожидает, что будет передана ссылка на массив, будьте внимательны!

### Дополнительная информация о `Net::LDAP`

`Net::LDAP` предоставляет интерфейс ко всем функциям LDAP, в том числе `Compare`, `delete` и `moddn`. Все они используются подобным же образом и подробно описываются в страницах справки `Net::LDAP`.

Все показанные примеры работают в режиме блокировки, что означает, что выход из функции производится после получения ответа от сервера. Кроме того, допускается работа в асинхронном режиме, при этом указывается функция обратного вызова, которая вызывается при получении пакета.

С помощью `Net::LDAP` можно использовать в сценариях данные из дерева LDAP. Perl используется в огромном множестве приложений, поэтому возможности интеграции очень широки.

### Разработка на C/C++

Использование библиотек С сложнее, чем библиотек Perl. На странице справки `ldap(3)` подробно описано, как работать с этой библиотекой, а также содержатся ссылки на другие страницы, описывающие каждую из функций. Для того, чтобы использовать библиотеки С LDAP, сначала необходимо включить в код файл `ldap.h`, например, с помощью инструкции `#include <ldap.h>`. После этого объектные файлы необходимо связать с `libldap` с помощью опции компоновщика `-lldap`.

### Резюме

Из этого руководства вы узнали о том, как установить и настроить автономный сервер OpenLDAP. Для настройки `slapd` используется файл `slapd.conf`. Необходимо располагать глобальные параметры в начале файла, за ними следует конфигурация бэк-энда и базы данных, поскольку `slapd` зависит от порядка директив. При возникновении сомнений

обратитесь к странице справки slapd.conf.

Модуль **Net::LDAP** позволяет работать с сервером LDAP из кода Perl. Сначала создаётся объект, после чего вызываются методы объекта, соответствующие операциям LDAP. Как правило, сначала выполняется **bind**, после чего подаётся нужный запрос. Важно проверить результаты работы функций с помощью функций **code** и **error**.

## Ресурсы

### Научиться

- Оригинал руководства "[Installation and development](#)" (EN).
- Изучите предыдущее руководство в серии 301 - "[Подготовка к экзамену LPI 301, Тема 301: понятия, архитектура и модель](#)" (EN) (developerWorks, октябрь 2007 г.).
- Чтобы освежить знания о командах управления пакетами, воспользуйтесь руководством developerWorks "[Установка Linux и управление пакетами](#)" (EN) (developerWorks, сентябрь 2005 г.).
- Чтобы познакомиться с основами Linux и подготовиться к сертификации в качестве системного администратора, ознакомьтесь со всей [серий руководств для подготовки к экзаменам LPI](#).
- В [программе LPIC](#) вы можете найти перечни заданий, примеры вопросов и подробные цели для трех уровней сертификации системных администраторов Linux института Linux Professional Institute. (EN)
- Ответы на вопросы [Что такое бэк-энд?](#) и [Что такое база данных?](#) в ответах на часто задаваемые вопросы по OpenLDAP.(EN)
- Узнайте больше о [создании оверлеев](#) из документации разработчика. Оверлей - сложная, но мощная концепция.(EN)
- Если вы не нашли нужную информацию на страницах справки для slapd.conf или используемого вами бэк-энда, обратитесь к [Руководству администратора OpenLDAP](#). Кроме того, могут быть полезными [ответы на вопросы по OpenLDAP](#).(EN)
- Очень советую онлайновую книгу [LDAP для больших учёных](#), несмотря на то, что работа над ней ещё не закончена.
- На странице [Perl-LDAP](#) имеется большое количество документации и советов по использованию **Net::LDAP**. (EN)
- В [разделе Linux сайта developerWorks](#) можно найти дополнительные ресурсы для разработчиков для Linux, а также [самые популярные среди наших читателей статьи и руководства](#). (EN)
- Посмотрите все [советы по Linux](#) и [руководства Linux](#) на сайте developerWorks.
- Следите на последними новостями на портале [Web-трансляций и технических мероприятий developerWorks](#).(EN)

### Получить продукты и технологии

- Загрузите [OpenLDAP](#).
- [IBM Tivoli Directory Server](#) один из серверов LDAP, который отлично интегрируется с другими продуктами IBM.(EN)
- [phpLDAPAdmin](#) - инструмент администрирования LDAP на базе Web. Если вам больше

нравится графический интерфейс, вам стоит посмотреть на [Luma](#).(EN)

- Используйте в своем следующем проекте разработки для Linux [ознакомительные версии программного обеспечения IBM](#), которые можно скачать непосредственно с developerWorks.(EN)

# Подготовка к экзамену LPI 301: Тема 303.

## Конфигурирование

*Профессионал Linux высокого уровня (LPIC-3)*

Шон Волберг, старший сетевой инженер, P.Eng

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux Professional Institute на квалификацию профессионала Linux высокого уровня (LPIC-3). В этом руководстве, третьем из [серии из шести руководств](#), Шон расскажет о конфигурировании сервера LDAP (Lightweight Directory Access Protocol), включающем в себя настройку списков контроля доступов, обеспечение безопасности и оптимизацию. Прочитав это руководство, вы узнаете о конфигурировании сервера LDAP.

[Больше статей из этой серии](#)

**Дата:** 26.06.2008

**Уровень сложности:** средний

### Предисловие

Узнайте, чему могут научить вас эти руководства, и как получить от них больше пользы.

### Об этой серии руководств

Институт [Linux Professional Institute](#) (LPI) сертифицирует системных администраторов Linux® по трём уровням: *младший уровень* (также называемый "уровень сертификации 1"), *углубленный уровень* (также называемый "уровень сертификации 2") и *высший уровень* (также называемый "уровень сертификации 3"). Для того чтобы получить сертификацию на уровне 1, нужно сдать экзамены 101 и 102. Для того чтобы получить сертификацию на уровне 2, нужно сдать экзамены 201 и 202. Для того чтобы получить сертификацию на уровне 3, у вас должна быть действующая сертификация на углубленном уровне и сдан экзамен 301 ("основной"). Кроме того, на высоком уровне вы можете сдать дополнительные специализированные экзамены.

developerWorks предлагает руководства, которые помогут вам подготовиться к пяти экзаменам для младшего, углубленного и высокого уровня. В каждом экзамене охватывается несколько тем, и для каждой темы на developerWorks есть соответствующий учебник для самостоятельного изучения. В таблице 1 перечислены шесть тем и соответствующие им руководства developerWorks для экзамена LPI 301.

**Таблица 1. Экзамен LPI 301: руководства и темы**

Тема экзамена	Руководство developerWorks	Краткое описание руководства
Тема 301	<a href="#">Подготовка к экзамену LPI 301: понятия, архитектура и модель</a>	Узнайте о понятиях и архитектуре LDAP, о том, как проектировать и внедрять каталог LDAP, а также о схемах.
Тема 302	<a href="#">Подготовка к экзамену LPI 301: установка и разработка</a>	Узнайте, как устанавливать, настраивать и использовать программное обеспечение OpenLDAP.

Тема 303	Подготовка к экзамену LPI 301: конфигурирование	(Это руководство) Узнайте более подробно о том, как настраивать программное обеспечение OpenLDAP. См. подробные <a href="#">цели</a> .
Тема 304	Подготовка к экзамену LPI 301: использование	Появится в ближайшее время.
Тема 305	Подготовка к экзамену LPI 301: интеграция и миграция	Появится в ближайшее время.
Тема 306	Подготовка к экзамену LPI 301: планирование пропускной способности	Появится в ближайшее время.

Для того чтобы сдать экзамен 301 (и получить сертификацию третьего уровня), вы должны:

- обладать несколькими годами опыта установки и поддержки Linux на большом числе компьютеров, используемых в различных целях
- обладать опытом интеграции с различными технологиями и операционными системами
- обладать профессиональным опытом или пройти профессиональную подготовку специалиста Linux корпоративного уровня (включая опыт, полученный при работе в другой роли)
- знать администрирование Linux на углубленном и высоком уровне, включая установку, управление, обеспечение безопасности, решение возникающих проблем и техническое обслуживание.
- уметь использовать инструменты с открытым исходным кодом для проведения измерений, необходимых для планирования пропускной способности и решения проблем с ресурсами
- иметь профессиональный опыт применения LDAP для интеграции с сервисами UNIX® и Microsoft® Windows®, в том числе Samba, Pluggable Authentication Modules (PAM), электронной почтой и Active Directory
- уметь планировать, проектировать, разрабатывать, строить и реализовывать полную среду с использованием Samba и LDAP, а также проводить измерения для планирования производительности и оценки безопасности служб
- уметь создавать сценарии на Bash или Perl или знать как минимум один язык системного программирования (например, C)

Институт Linux Professional Institute не дает рекомендаций по каким-либо конкретным материалам и методикам для подготовки к экзаменам, разработанным сторонними лицами.

## Об этом руководстве

Добро пожаловать в третье из шести руководств, призванных помочь вам подготовиться к сдаче экзамена LPI 301, - "Конфигурирование". Из этого руководства вы узнаете о конфигурировании сервера LDAP, рассмотрев вопросы управления доступом, обеспечения безопасности, настройки репликации и оптимизации производительности базы данных.

Это руководство организовано в соответствии с целями LPI по этой теме. Условно говоря, чем выше вес цели, тем больше вопросов по этой теме будет на экзамене.

## Цели

В таблице 2 подробно перечислены цели этого руководства.

**Таблица 2. Конфигурирование: цели экзамена, описанные в этом руководстве**

Цель экзамена LPI	Вес цели	Краткое описание цели
303.2 <a href="#"><u>Списки управления доступом в OpenLDAP</u></a>	2	Проектирование и внедрение списков управления доступом.
303.3 <a href="#"><u>Репликация LDAP</u></a>	5	Настройка OpenLDAP для репликации данных между несколькими серверами.
303.4 <a href="#"><u>Обеспечение безопасности каталога</u></a>	4	Настройка шифрованного доступа к серверу LDAP и ограничение доступа на уровне брандмауэра.
303.5 <a href="#"><u>Оптимизация производительности сервера LDAP</u></a>	2	Оценка текущего уровня производительности вашего сервера LDAP и настройка его на максимальный уровень.
303.6 <a href="#"><u>Конфигурация демона OpenLDAP</u></a>	2	Знакомство с основными директивами конфигурационного файла slapd.conf и опциями командной строки демона slapd.

## Необходимые условия

Чтобы получить максимум от этого руководства, вы должны обладать глубокими знаниями Linux и иметь работающую Linux-систему, на которой вы сможете практиковаться в выполнении рассматриваемых задач.

Если ваши базовые знания Linux немного устарели, вы можете сначала познакомиться с [руководствами для экзаменов LPIC-1 и LPIC-2](#).

Различные версии программ могут выводить данные в различных форматах, поэтому результаты, полученные вами, могут отличаться от листингов и рисунков, приведенных в этом руководстве.

## Требования к системе

Для того чтобы выполнить примеры, приведенные в этом руководстве, вам потребуется рабочая станция под управлением Linux с пакетом OpenLDAP и поддержкой RAM. Большинство современных дистрибутивов удовлетворяют этим требованиям.

# Подготовка к экзамену LPI 301: Тема 303. Конфигурирование

*Профессионал Linux высокого уровня (LPIC-3)*

[Шон Волберг](#), старший сетевой инженер, P.Eng

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux Professional Institute на квалификацию профессионала Linux высокого

уровня (LPIC-3). В этом руководстве, третьем из [серии из шести руководств](#), Шон расскажет о конфигурировании сервера LDAP (Lightweight Directory Access Protocol), включающем в себя настройку списков контроля доступов, обеспечение безопасности и оптимизацию. Прочитав это руководство, вы узнаете о конфигурировании сервера LDAP.

## Списки управления доступом в LDAP

В этом разделе описывается материал по теме 303.2 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 2.

Из этого раздела вы узнаете, как:

- Проектировать списки управления доступом LDAP
- Узнаете о синтаксисе управления доступом
- Предоставлять и отзывать права доступа LDAP

В дереве LDAP могут храниться самые различные данные, такие как телефонные номера, дни рождения или информация о денежных выплатах. Некоторые сведения могут быть публичными, а некоторые могут предназначаться только для определенного круга лиц. Также для различных пользователей могут быть определены различные типы доступов к этой информации. Например, можно указать, что только хозяин записи и администраторы могут изменять телефонный номер, а просматривать его может каждый. Все эти ограничения задаются через списки управления доступом (Access Control Lists, ACLs).

### Проектирование списков управления доступом LDAP

Прежде чем приступить к написанию конфигурации, вы должны определить цели, которых вы хотите достичь. Какие разделы дерева каталога будут содержать конфиденциальную информацию? Какие атрибуты необходимо защитить и от кого? Каким образом будет использоваться дерево каталога?

### Компоненты ACL

Каждая запись ACL содержит следующую информацию:

1. **Какие элементы и атрибуты определены в ACL**
2. **К кому применяется ACL**
3. **Какой уровень доступа предоставляется**

## Регулярные выражения

Регулярные выражения используются для поиска и изменения текста на основе выбранных правил. Вы можете иметь общее представление о том, как может выглядеть текстовый фрагмент, или знать, каким определенным шаблонам он может соответствовать, и на основании этого построить регулярное выражение, которое поможет вам найти необходимый текст. Регулярное выражение, или сокращенно *регекс* (от англ. regex) состоит из символьных констант и метасимволов, на основе которых строятся различные шаблоны.

Простым регулярным выражением является `hello`, что соответствует любой строке символов, содержащей шаблон `hello`.

Вы можете не знать, является ли первая буква строки заглавной, или нет. В этом случае используйте метасимволы `[ ]`, соответствующие любому единичному символу из числа заключённых в скобки. Так, регекс `[Hh]ello` соответствует как строке `hello`, так и строке `Hello`.

Точка в регулярном выражении соответствует любому единичному символу. Регекс `.ello` соответствует строкам `Hello` и `hello`, но также и строкам `fellow` и `cello`. Тем не менее,

этот регекс не соответствует строке `ello`, поскольку точка должна соответствовать какому-либо символу.

Символы `?`, `*` и `+` соответствуют нулю или одной копии предыдущего символа, нулю или нескольким копиям предыдущего символа, и одной или более копиям предыдущего символа соответственно. Таким образом, регулярное выражение `hello+` соответствует строкам `hello` и `heloooooooo`, но не строке `hell`.

Регулярные выражения имеют множество различных опций и позволяют вам эффективно находить нужные строки в текстовых файлах. В контексте OpenLDAP регулярные выражения используются для определения общих фрагментов в различающихся именах (Distinguished Name, DN), таким образом, избавляя от необходимости набирать вручную сотни возможных вариантов.

В условии "what" вы можете указать определенное различающееся имя (DN) объекта, фильтр запроса LDAP, список атрибутов или определенную комбинацию всех вышеперечисленных элементов. Фильтрация по различающимся именам позволит вам выбрать точное значение, такое как `ou=People,dc=ertw,dc=com`, или значения, соответствующие регулярному выражению (см. раздел "[Регулярные выражения](#)"). С помощью фильтра запроса LDAP можно выбрать определенный класс `objectClass` или другие атрибуты объекта. Список атрибутов представляет собой список имен атрибутов, разделенных запятыми. Более сложным условием может являться такое условие, как "Все пароли пользователей подразделения `ou=People,dc=ertw,dc=com`, являющихся администраторами".

При указании учетных записей, к которым применяется запись ACL, вам предоставляется большая гибкость. Обычно для идентификации пользователей используется различающееся имя под названием `bindDN`, под которым они привязываются к дереву объектов. Каждый элемент LDAP может иметь атрибут `userPassword`, использующийся для аутентификации определенного пользователя. В некоторых ситуациях вы можете использовать ключевое слово `self`, соответствующее имени текущего пользователя, выполнившего вход в систему. Это может оказаться полезным в случаях, когда вы хотите разрешить пользователям редактировать их собственные данные.

Если пользователь не привязан к дереву LDAP, он считается *анонимным* (*anonymous*). По умолчанию анонимные пользователи могут просматривать данные каталога, поэтому вы должны решить, стоит ли оставлять им такой доступ. Далее вы можете сгруппировать анонимных пользователей (также как и любых других) по IP-адресам или методу подключения к каталогу LDAP, такому как использование открытых паролей или шифрованное подключение.

После того как вы решили, доступ к каким объектам и кому необходимо предоставить, вы должны определиться с уровнем доступа, который может варьироваться от `none` (запретить все) до `write` (разрешить изменения). Также вы можете разрешить пользователю проходить аутентификацию на основе определенной записи, но запретить ее чтение, или, предположим, разрешить выполнять операции чтения (`read`), поиска (`search`) и сравнения (`compare`).

Независимо от настроек ACL любые учетные записи, определенные как `rootDN` (пользователи, являющиеся администраторами LDAP), всегда будут иметь полный доступ к соответствующей базе данных. Не существует других способов изменить это поведение, кроме как удалить конфигурацию `rootDN` из файла `slapd.conf`.

#### Синтаксис записей управления доступом

Основная форма записи ACL, представленная в нормальной форме Бэкуса-Наура, имеет следующий вид:

```
access to <what> [ by <who> [ <access> ] [ <control> ] ]+
```

## **Нормальная форма Бэкуса-Наура**

Нормальная форма Бэкуса-Наура (БНФ) является способом описания синтаксиса языков программирования, данных и протоколов, в том числе синтаксиса ACL. Являясь краткой, и в то же время очень точной, БНФ часто используется при разработке протоколов Интернета.

В нотации БНФ используется конструкция из двух частей, разделенных знаком ::=, который означает, что левая часть может быть заменена элементами, перечисленными в правой части. Элементы правой части конструкции БНФ, заключенные в угловые скобки (< и >), относятся к элементам левой части, заключенным в такие же скобки.

Элементы, заключенные в квадратные скобки ([ и ]), являются необязательными.

Вертикальная черта (|) означает "одно из нескольких", а символы + и \* означают "одно или несколько из предшествующих" и "ничего, одно или несколько из предшествующих" соответственно. Если вы имели дело с регулярными выражениями, вам будут знакомы многие использующиеся здесь обозначения.

контексте описания ACL с помощью БНФ используется следующий синтаксис. Каждая запись ACL состоит из символьной строки "access to", после которой идет условие "what", определенное где-либо в другом месте. Далее идут одна или более строк вида "by <who> [ <access> ] [ <control> ]", в которых параметры who, access и control определены где-либо в другом месте, причем параметры access и control являются необязательными.

Остальной синтаксис мы рассмотрим в оставшейся части руководства.

### **Описание условия what**

Условие *what* определяет, какие объекты и атрибуты попадают под действие правила ACL. Описание синтаксиса этого условия в нотации БНФ представлено в листинге 1.

### **Листинг 1. Описание синтаксиса условия what в нотации БНФ**

```
<what>      ::= * |  
             [dn[.<basic-style>]=<regex> | dn.<scope-style>=<DN>]  
             [filter=<ldapfilter>] [attrs=<attrlist>]  
<basic-style> ::= regex | exact  
<scope-style> ::= base | one | subtree | children  
<attrlist>   ::= <attr> [val[.<basic-style>]=<regex>]  
                 | <attr> , <attrlist>  
<attr>       ::= <attrname> | entry | children
```

Некоторые из элементов листинга 1, такие как DN и regex, не определены непосредственно в этом фрагменте кода. Формат различающегося имени вам уже знаком, а регулярные выражения лучше всего изучать отдельно от БНФ.

Из листинга 1 видно, что условием *what* правила ACL может являться либо знак звездочки (\*), который соответствует любому значению, либо комбинация различающегося имени, фильтра поиска LDAP и списка атрибутов. В последнем случае могут использоваться один или несколько из этих трех элементов, поскольку каждый из них заключен в квадратные скобки.

В листинге 2 приведены три условия *what*, определяющих различающееся имя (DN).

### **Листинг 2. Три примера условий what**

```
dn.exact="ou=people,dc=ertw,dc=com"
```

```
dn.regex="ou=people,dc=ertw,dc=com$"
dn.regex="^cn=Sean.*,dc=com$"
```

Первому условию удовлетворяет исключительно объект `ou=people,dc=ertw,dc=com`; эта запись ACL не будет соответствовать ни каким-либо дочерним объектам, таким как `cn=Sean Walberg,ou=people,dc=ertw,dc=com`, ни родительскому объекту.

Второе условие похоже на первое за исключением того, что в нем используется регулярное выражение и якорь строки – знак доллара (\$). Якорь определяет не часть строки, а ее расположение. Знак доллара означает конец строки, и поэтому второму условию удовлетворяет любая строка, оканчивающаяся на `ou=people,dc=ertw,dc=com`, в том числе строка `cn=Sean Walberg,ou=people,dc=ertw,dc=com`. Обратите внимание на то, что без использования якоря строка поиска может располагаться в любом месте целевой строки, такой как `ou=people,dc=ertw,dc=com,o=MegaCorp`.

В последнем примере листинга 2 используется еще один якорь – символ ^, означающий начало строки. Кроме того, в этом примере используется регулярное выражение `.*`. Точка означает любой символ, а знак звездочки означает ноль или более предшествующих символов. Таким образом регекс `.*` соответствует любой строке, состоящей из нуля или более символов. Объединив все правила, мы увидим, что третьему условию удовлетворяет любая строка, начинающаяся на `cn=Sean` и оканчивающаяся на `dc=com`.

Также при определении условия `what` вы можете использовать фильтры запросов LDAP – наиболее полезное средство для поиска объектов на основе их классов (`objectClass`). Например, условию `filter=(objectClass posixAccount)` удовлетворяют все объекты класса `posixAccount`. Для получения информации об атрибуте `objectClass` обратитесь к первому руководству этой серии – [Подготовка к экзамену LPI 301: понятия, архитектура и модель \(EN\)](#).

Частью составления условий `what` является указание атрибутов. Как правило, данный способ используется для указания того, какие конфиденциальные атрибуты, в особенности пароли, могут быть доступны пользователям. Для того чтобы определить правило, относящееся к паролям, укажите атрибут `attrs=userPassword`.

После того как вы укажете, какие объекты и атрибуты попадают под действие правила ACL, вы должны будете указать пользователей, на которых будет распространяться это правило.

### Описание условия who

Доступы предоставляются пользователю на основании различающегося имени DN, которое определяется в момент привязки клиента к каталогу. Обычно поиск различающегося имени производится в дереве каталога, но также этим именем может являться имя `rootDN`, указанное в файле `slapd.conf`.

Описание синтаксиса условия `who` в нотации БНФ представлено в листинге 3.

### Листинг 3. Описание синтаксиса условия who в нотации БНФ

```
<who> ::= * | [anonymous | users | self[.<selfstyle>]
                  | dn[.<basic-style>]=<regex> | dn.<scope-style>=<DN>]
                  [dnattr=<attrname>]
                  [group[/<objectclass>[/<attrname>][.<basic-style>]]=<regex>]
                  [peername[.<peernamestyle>]=<peername>]
                  [sockname[.<style>]=<sockname>]
                  [domain[.<domainstyle>[,<modifier>]]=<domain>]
                  [ssf=<n>]
```

```

[transport_ssf=<n>]
[tls_ssf=<n>]
[sasl_ssf=<n>]

<style> ::= {exact|regex|expand}
<selfstyle> ::= {level{<n>}}
<dnstyle> ::= {{exact|base(object)}|regex
                |one(level)|sub(tree)|children|level{<n>}}
<groupstyle> ::= {exact|expand}
<peernamestyle> ::= {<style>|ip|path}
<domainstyle> ::= {exact|regex|sub(tree)}
<modifier> ::= ={expand}

```

Также как и в условии `what`, здесь знак звездочки означает любое значение. Существует множество способов конкретизировать данное условие. В OpenLDAP определены три ключевых слова – `anonymous`, `users` и `self`, которые означают незарегистрированных пользователей, пользователей, прошедших проверку, и текущего пользователя, выполнившего вход в систему, соответственно. Ключевое слово `self` часто используется для того, чтобы разрешить вошедшему в систему пользователю изменять данные своего профиля. Данная функция основана на точном совпадении с DN пользователя; если информация о пользователе размещена в нескольких различных объектах, ссылка `self` действует только для той записи, с помощью которой была выполнена привязка к каталогу.

Ключевое слово `self` имеет интересную особенность: вы можете применять правило ACL к родительским или дочерним записям текущей записи пользователя, используя другое ключевое слово - `level`. Ссылка `self.level{1}` соответствует текущей и родительской записям пользователя, а ссылка `self.level{-1}` – текущей и всем непосредственно присоединенным дочерним записям.

Что касается различающегося имени, то вы можете отфильтровать атрибут DN на основе регулярного выражения или точного совпадения, используя конструкции `dn.exact="DN"` и `dn.regex="regex"` соответственно. Далее в этом руководстве будет рассмотрен пример динамического связывания условий `what` и `who`.

Произвольные элементы каталога могут быть защищены с помощью ключевого слова `dnattr`, которое используется совместно с именем атрибута. Если DN инициатора запроса содержится в указанном атрибуте целевого объекта, то считается, что условие ACL выполняется. Например, если вы добавите в ACL условие `dnattr=manager`, а в запись пользователя Fred Smith добавите атрибут `manager: cn=Joe Blow,ou=people,dc=ertw,dc=com`, то при обращении пользователя Joe Blow к записи пользователя Fred Smith условие ACL будет выполняться.

Ключевое слово `group` похоже на `dnattr` за исключением того, что его параметры относятся к группе, определенной где-либо в другом месте дерева, а не к атрибуту элемента каталога. По умолчанию классом объекта (`objectClass`) группы является `groupOfNames`, а участники группы перечислены в атрибуте `member`.

Ключевые слова `peername`, `sockname` и `domain` используются для идентификации клиентского подключения. Ключевое слово `peername` используется для указания IP-адреса клиента, например, `peernameip=127.0.0.1`. Ключевое слово `sockname` относится к редко используемым подключениям через именованные каналы (named pipes), а `domain` используется для указания связанного с IP-адресом имени узла, которое может быть легко подделано.

Заключительный ряд опций относится к уровню безопасности подключения, который в

терминах OpenLDAP называется Security Strength Factor (SSF). Эти опции окажутся более понятными, если вы знакомы с механизмами безопасности, использующимися для подключений к OpenLDAP, такими как защита транспортного уровня (Transport Layer Security, TLS) и механизм аутентификации Simple Authentication and Security Layer (SASL).

Вы можете использовать различные комбинации всех вышеперечисленных опций. Например, вы можете разрешить изменять пароли только тем пользователям, которые одновременно являются администраторами, подключающиеся с заданного диапазона IP-адресов и используют определенный уровень шифрования. Вы можете определять и более простые условия; например, вы можете разрешить подключаться к каталогу только существующим в системе пользователям или же предоставлять доступ вне зависимости от результатов проверки подлинности.

### **Описание параметра access**

После того как вы определили, кому и к каким объектам каталога разрешен доступ, вы должны указать уровень предоставляемого доступа. Описание параметра access в нотации БНФ представлено в листинге 4.

#### **Листинг 4. Описание параметра access в нотации БНФ**

```
<access> ::= [[real]self]{<level>|<priv>}
<level> ::= none|disclose|auth|compare|search|read|write
<priv> ::= {=|+|-}{w|r|s|c|x|d|0}+
```

Когда уровень доступа указывается в формате **level**, каждый последующий уровень включает в себя все предыдущие уровни. Таким образом, предоставляя разрешение **read**, вы также предоставляете разрешения **search**, **compare**, **auth** и **disclose**. Уровни доступа **none** и **disclose** запрещают любой доступ к данным и отличаются лишь тем, что некоторые сообщения об ошибках, которые могут раскрыть информацию о содержимом дерева каталога, не выводятся на уровне **none** и выводятся на уровне **disclose**.

Альтернативным способом является указание уровня доступа в терминах разрешенных операций LDAP с использованием формата **priv**. В этом формате опции перечислены в обратном порядке относительно формата **level**; опция **w** означает операцию записи, а опция **0** – полный запрет любых операций. При указании уровня доступа в формате **priv** не происходит неявного наследования разрешений, как в случае использования формата **level**; если вы хотите предоставить полный доступ, вы должны сделать это с помощью строки **wrscx**.

Символы **=/+/-** перед буквенными опциями определяют способ совместного использования указанного разрешения и уже действующих разрешений в тех случаях, когда применяются несколько правил. Если перед опцией доступа стоит символ **=**, все определенные ранее разрешения игнорируются, и используется указанное разрешение. Если используются символы **+** или **-**, то указанное разрешение соответственно добавляется или удаляется из действующих разрешений.

### **Параметр control**

По умолчанию применение списков доступа в OpenLDAP происходит по методу первого совпадения. OpenLDAP находит первую запись ACL, удовлетворяющую условию **what**, и в пределах этой записи ищет первый элемент, удовлетворяющий условию **who**. Этот метод соответствует ключевому слову **stop**, указанному после описания уровня доступа. Два других ключевых слова – это **continue** и **break**. Если вы используете ключевое слово **continue**, то в текущей записи ACL производится поиск следующего элемента,

удовлетворяющего условию who. Если вы используете ключевое слово **break**, обработка текущей записи ACL прекращается, но выполняется поиск следующей записи ACL, удовлетворяющей условию who.

### Собираем вместе все компоненты правила ACL

Теперь, когда мы рассмотрели все три (четыре, если считать параметр control) компонента правила ACL, можно собрать их воедино и создать политику доступа. В листинге 5 приведен простой список ACL, который позволяет зарегистрированным в системе пользователям просматривать дерево каталога и изменять (но не просматривать) свои личные пароли.

### Листинг 5. Простой список ACL

```
access to attrs=userPassword
    by self =xw
    by anonymous auth

access to *
    by self write
    by users read
```

Первое правило ACL применяется в случаях обращения пользователей к полю **userPassword**. Каждый пользователь может изменять свой пароль и использовать его для входа в систему. Эти разрешения предоставляются путем использования знака "**=**". Анонимным пользователям разрешено проходить аутентификацию. Поскольку в момент привязки к дереву каталога любой пользователь является анонимным, ему необходимо предоставить разрешение **auth** для того, чтобы он мог войти в систему и стать обычным, привилегированным пользователем.

Если пользователи обращаются к элементам каталога, которые не являются паролями, то в действие вступает второе правило ACL. Опять же, каждый пользователь имеет полный доступ ко всем полям своей записи (за исключением поля **userPassword** в силу действия первого правила ACL), в то время как все прошедшие проверку пользователи могут просматривать остальные данные каталога.

В листинге 6 приведена запись ACL, связывающая условия what и who и содержащая регулярные выражения.

### Листинг 6. Немного фантазии с использованием регулярных выражений

```
access to dn.regex="cn=([^\,]+),ou=addressbook,dc=ertw,dc=com"
    by dn.regex="cn=$1,ou=People,dc=ertw,dc=com" write
    by users read
```

Правило ACL, приведенное в листинге 6, разрешает пользователям изменять соответствующие им записи в ветке **ou=addressbook,dc=ertw,dc=com** дерева каталога. Регулярное выражение **[^\,]+** соответствует строке любых символов за исключением запятой, а круглые скобки сохраняют эту строку в качестве значения переменной **\$1**; следующая пара скобок сохраняется в переменную **\$2**, и так далее до **\$9** включительно. В условии who полученное имя пользователя повторно используется для определения того, кто может получить доступ к записи каталога. Если имя пользователя совпадает с именем записи, к которой он обращается, то ему предоставляется полный доступ. В противном случае прошедший проверку пользователь получает доступ только на просмотр

этой записи.

### Практические рекомендации

Поскольку в OpenLDAP используется метод первого совпадения, размещайте более конкретизированные записи ACL в начале списка; в противном случае существует большая вероятность того, что при соответствии записи ACL более общему условию последующие записи списка будут проигнорированы. Этот прием также можно использовать для предоставления или отзыва доступов у отдельного пользователя; для этого просто поместите запись ACL, содержащую условие для конкретного пользователя, в начало списка.

По возможности старайтесь создавать простые записи ACL. Это уменьшает вероятность ошибок и повышает производительность системы, поскольку записи ACL анализируются каждый раз при обращении к дереву каталога.

## Репликация LDAP

В этом разделе описывается материал по теме 303.3 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 5.

Из этого раздела вы узнаете, как:

- Познакомитесь с принципами репликации
- Настроить репликацию OpenLDAP
- Работать со службой slurpd
- Анализировать журналы репликации
- Узнаете, что такое узлы реплик
- Настроить возвращаемые ссылки LDAP
- Настроить репликацию LDAP посредством sync

В какой-то момент использование единственного сервера LDAP может перестать удовлетворять вашим потребностям. Возможность выхода из строя сервера LDAP в вашей организации может оказаться неприемлемой, или же нагрузка на сервер может оказаться достаточно высокой, и вы решите распределить ее между несколькими серверами. Неважно, по какой причине, но у вас может возникнуть потребность в использовании нескольких серверов.

Вы можете разместить отдельные части каталога LDAP на нескольких различных серверах, однако это ведет к снижению уровня надежности, не говоря о сложности правильного распределения запросов. В идеальном случае на каждом сервере содержится одинаковая копия каталога. Все изменения, произошедшие на любом из серверов, передаются на остальные серверы определенным способом, обеспечивающим актуальность данных в любой момент времени. Этот процесс называется *replication*.

Этот сценарий, называющийся *multi-master*репликацией, является довольно сложным, поскольку в этом случае не существует единственного, четко определенного сервера, управляющего данными. Чаще всего используется репликация master-slave, при которой один главный (master) сервер управляет всеми изменениями каталога и рассыпает их на подчиненные (slave) серверы. Запросы LDAP при этом могут быть обработаны любым из серверов. Данная схема может быть расширена до репликации с использованием узла *реплики*, когда данные с главного сервера реплицируются на один подчиненный сервер, а он в свою очередь реплицирует эти данные на все остальные подчиненные серверы.

В OpenLDAP существует два метода репликации. В первом методе используется *slurpd* – отдельный демон, который отслеживает изменения на главном сервере и передает их на подчиненные серверы. Во втором методе используется встроенный механизм репликации

LDAP Sync сервера OpenLDAP, также известный как *syncrep*. Репликация посредством slurpd считается устаревшей, и пользователям все больше предлагается использовать syncrep. В этом разделе будут рассмотрены оба этих метода.

### Репликация посредством slurpd

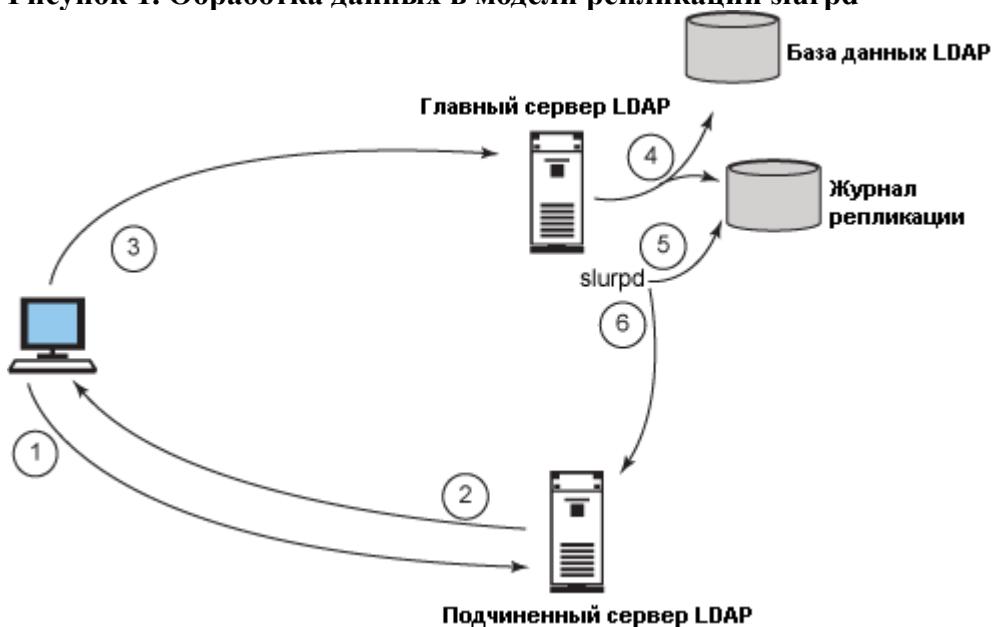
Репликация посредством slurpd является извещающей (push) репликацией, при которой главный сервер извещает об изменениях все подчиненные серверы. Если клиент пытается обновить данные на подчиненном сервере, этот сервер посыпает клиенту так называемую *возвращающую ссылку* (referral), перенаправляющую его на главный сервер. За подачу повторного запроса на главный сервер отвечает клиентский компьютер. Slurpd является отдельным демоном и настраивается в файле slapd.conf.

### Обработка данных в модели репликации slurpd

Главный сервер – это сервер, обрабатывающий все поступающие от клиентов запросы на изменение каталога и содержащий достоверный источник данных. Любые изменения, произошедшие в дереве каталога главного сервера, записываются в *журнал репликации*, за которым наблюдает демон slurpd. Обнаружив изменения в журнале репликации, slurpd извещает о них все подчиненные серверы.

На рисунке 1 изображена схема работы демона slurpd.

**Рисунок 1. Обработка данных в модели репликации slurpd**



Описание процесса:

1. Клиентский компьютер посылает запрос на обновление данных, который случайным образом попадает на подчиненный сервер.
2. Подчиненный сервер знает о том, что он может выполнять операции записи только в случае получения их от своего партнера по репликации, и поэтому посыпает клиенту возвращающую ссылку, перенаправляющую его на главный сервер.
3. Клиент повторно посыпает запрос на обновление данных, обращаясь к главному серверу.
4. Главный сервер выполняет обновление данных и записывает изменения в журнал репликации.
5. Slurpd, который также запущен на главном сервере, обнаруживает изменения в журнале репликации.
6. Slurpd направляет изменения на подчиненный сервер.

Таким образом, подчиненные серверы могут быть синхронизированы с главным сервером с небольшой задержкой. Slurpd всегда знает, какие подчиненные сервера нуждаются в обновлении, если произошла какая-либо задержка или сбой.

## Настройка slurpd

Настройка репликации посредством slurpd состоит из следующих шагов:

1. Создание учетной записи, которую slurpd будет использовать для аутентификации подчиненной реплики.
2. Настройка имени подчиненного сервера на главном сервере.
3. Настройка подчиненного сервера в качестве реплики, в том числе настройка необходимых списков управления доступом.
4. Копирование базы данных с главного сервера на подчиненный сервер.

Создание учетной записи реплики не представляет особой сложности. Единственным требованием для этого является наличие в учетной записи атрибута `userPassword`. Вы можете использовать класс объекта `inetOrgPerson objectClass`, как в случае создания учетных записей, принадлежащих служащим, или более общий класс объекта, такой как `account`, добавив к нему вспомогательный класс `simpleSecurityObject`. Возвращаясь к первому руководству, вспомним, что структурные классы объектов описывают запись (и поэтому вы можете использовать только один класс объекта для каждой записи), тогда как вспомогательные классы объектов добавляют атрибуты к записи независимо от ее структурного класса. В листинге 7 приведен код в LDIF-формате (LDAP Data Interchange Format – формат обмена данными LDAP) для добавления учетной записи реплики.

### Листинг 7. LDIF код для добавления учетной записи реплики

```
dn: uid=replica1,dc=ertw,dc=com
uid: replica1
userPassword: replica1
description: Account for replication to slave1
objectClass: simpleSecurityObject
objectClass: account
```

В листинге 7 представлена простая запись, содержащая только имя пользователя, пароль и описание, чего вполне достаточно для целей репликации. Описание не является обязательным, однако рекомендуется использовать его в целях документирования. Запомните пароль – он понадобится на следующем шаге!

Теперь для работы slurpd необходимо настроить главный сервер на сохранение всех изменений в журнале репликации, а также настроить реплику. Важно помнить о том, что slurpd извещает об изменениях все подчиненные серверы, а его параметры конфигурации настраиваются в файле `slapd.conf`. Это в свою очередь поможет вам помнить о том, где нужно настраивать репликацию, и о том, что учетные данные для аутентификации располагаются на главном сервере. Поскольку учетные данные являются частью дерева каталога, подчиненный сервер всегда сможет проверить их. В листинге 8 приведены настройки главного сервера, обеспечивающие возможность репликации.

## **Листинг 8. Настройка репликации посредством slurpd на главном сервере**

```
replica uri=ldap://slaveserver.ertw.com
    suffix="dc=ertw,dc=com"
    binddn="uid=replica1,dc=ertw,dc=com"
    credentials="replica1"
    bindmethod=simple

replogfile /var/tmp/replicationlog
```

Настройка репликации происходит в режиме базы данных, поэтому убедитесь, что команда replicareplica стоит где-нибудь после настройки первого параметра database. Команда replicacодержит ряд параметров в формате параметр=значение. Параметр uriопределяет имя или IP-адрес подчиненного сервера в формате унифицированного идентификатора ресурса (Uniform Resource Identifier, URI). Перед именем подчиненного сервера ставится префикс ldap://.

После того, как вы указали имя подчиненного сервера, вы можете указать с помощью параметра suffix необязательное имя базы данных для репликации. По умолчанию реплицируются все базы данных.

Заключительным требованием является предоставление информации об учетных данных, чтобы slurpd мог подключаться к указанной ссылке uri. Для выполнения простой аутентификации вам будет достаточно настроить параметры binddn, bindmethod и credentials (параметр userPassword был настроен вами ранее).

На заключительном этапе настройки главного сервера необходимо указать службе slapd, где должен храниться журнал репликации. Необходимо указать полный путь, поскольку относительные пути не будут работать. Вам не нужно беспокоиться о создании файла журнала, поскольку slapd сделает это за вас; тем не менее, указанный вами путь должен быть доступен для записи пользователю, от имени которого выполняются демоны slapd и slurpd.

На подчиненном сервере вы должны настроить учетную запись репликации, а также указать, что получая запросы на изменение данных, он должен перенаправлять клиентов на главный сервер с помощью возвращаемой ссылки.

## **Листинг 9. Конфигурация подчиненного сервера**

```
updatedn uid=replica1,dc=ertw,dc=com
updateref ldap://masterserver.ertw.com
```

Значением параметра updatedпоявляется учетная запись, созданная ранее на главном сервере. Эту учетную запись будет использовать демон slurpd для извещения подчиненных серверов об изменениях, произошедших в дереве каталога. Параметр updateref – это еще одна ссылка URI, указывающая на главный сервер LDAP. В листинге 10

приведен пример обновления клиентом подчиненного сервера и получения возвращаемой ссылки после настройки вышеуказанной конфигурации.

#### **Листинг 10. Получение клиентом возвращаемой ссылки при попытке обновления данных на подчиненном сервере**

```
[root@slave openldap]# ldapadd -x -D cn=root,dc=ertw,dc=com -w mypass -f newaccount.ldif
adding new entry "cn=David Walberg,ou=people,dc=ertw,dc=com"
ldap_add: Referral (10)
referrals:
    ldap://masterserver.ertw.com/cn=David%20Walberg,ou=People,dc=ertw,dc=com
```

Клиент командной строки OpenLDAP не переходит по возвращаемым ссылкам, но другие библиотеки LDAP делают это. Если вы используете LDAP в окружении с работающей репликацией, вы должны убедиться, что ваши приложения корректно переходят по возвращаемым ссылкам.

На заключительном этапе настройки процесса репликации необходимо обеспечить идентичность базы данных на главном и подчиненных серверах. Для этого выполните следующие шаги:

1. Остановите главный сервер LDAP.
2. Остановите подчиненный сервер LDAP.
3. Скопируйте все файлы базы данных с главного сервера на подчиненный сервер.
4. Запустите главный и подчиненный серверы.
5. Запустите демон slurpd.

Перед копированием базы данных оба сервера LDAP должны быть остановлены для того, чтобы во время этой процедуры в каталоге не могли произойти какие-либо изменения. Крайне важно, чтобы оба сервера начали работу с идентичным набором данных, иначе впоследствии может произойти их рассинхронизация. Репликация посредством slurpd, по существу, выполняет на подчиненном сервере все те же транзакции, что происходят на главном сервере, поэтому любые расхождения могут привести к проблемам.

В зависимости от дистрибутива и сценариев загрузки операционной системы slurpd может автоматически запускаться вместе с slapd. Если демон slurpd не запускается автоматически, запустите его вручную из командной строки, выполнив команду slurpd.

Прежде чем двигаться дальше, у вас должна быть настроена работающая репликация. Создайте учетную запись на главном сервере и проверьте работу службы репликации. Убедитесь также, что при получении запросов на обновление данных подчиненный сервер посылает клиенту возвращаемые ссылки.

#### **Мониторинг репликации**

Очень важно уметь выполнять мониторинг репликации, поскольку в процессе работы могут возникать различные ошибки, приводящие к рассинхронизации данных. Эти же знания пригодятся и при отладке.

Файлы slurpd хранятся в каталоге var/lib/ldap/replica (отдельно от журнала репликации службы slapd). В этом каталоге содержатся собственные журналы репликации службы slurpd и прочие так называемые *reject*-файлы (файлы отклонения). Если попытка slurpd обновить подчиненный сервер оканчивается неудачей, то данные сохраняются в файл с расширением .rej. В этом файле содержится LDIF код записи, а также описание возвращенной сервером ошибки, например, ERROR: Already exists. В листинге 11 приведен пример reject-файла, содержащего другую ошибку.

### Листинг 11. Reject-файл репликации

```
ERROR: Invalid DN syntax: invalid DN
replica: slaveserver.ertw.com:389
time: 1203798375.0
dn: sendmailMTAKey=testing,ou=aliases,dc=ertw,dc=com
changetype: add
objectClass: sendmailMTAAliasObject
sendmailMTAAliasGrouping: aliases
sendmailMTACluster: external
sendmailMTAKey: testing
sendmailMTAAliasValue: testinglist@ertw.com
structuralObjectClass: sendmailMTAAliasObject
entryUUID: 5375b66c-7699-102c-822b-fbf5b7bc4860
creatorsName: cn=root,dc=ertw,dc=com
createTimestamp: 20080223202615Z
entryCSN: 20080223202615Z#000000#00#000000
modifiersName: cn=root,dc=ertw,dc=com
modifyTimestamp: 20080223202615Z
```

Файл отклонения, приведенный в листинге 11, начинается с текстового описания ошибки ("ERROR: Invalid DN syntax: invalid DN"), после которого следует код в формате LDIF. Обратите внимание на то, что первым атрибутом является **replica** – имя реплики, которая не смогла выполнить обновление, а вторым атрибутом является **time** – время возникновения ошибки (в формате времени UNIX). Следующие несколько атрибутов относятся к записи, которая была отклонена.

Последние семь атрибутов называются **операционными атрибутами**. Эти атрибуты не относятся к исходному обновлению, а были добавлены сервером LDAP в целях внутреннего мониторинга. Записи LDAP был присвоен универсальный уникальный идентификатор (Universally Unique Identifier, UUID), а также некоторая информация о том, когда и кем эта запись изменялась.

Ошибка, показанная в листинге 11, скорее всего, произошла по причине отсутствия необходимой схемы на подчиненном сервере, который не смог опознать атрибут **sendmailMTAKey**, и, как следствие, посчитал DN записи некорректным. Прежде чем можно будет возобновить репликацию, необходимо обновить схему подчиненного сервера.

Чтобы применить отклоненную запись, вы должны найти ошибку и

устранить причины ее возникновения. Когда вы уверены, что отклоненная запись будет применена без ошибок, воспользуйтесь режимом `slurpd one-shot mode`, выполнив команду `slurpd -r /path/to/rejection.rej -o`. Параметр `-r` указывает `slurpd` прочесть указанный журнал репликации, а параметр `-o` переводит `slurpd` в режим `one-shot mode`, который означает, что по завершении обработки журнала `slurpd` закончит свою работу, а не перейдет в режим ожидания с целью добавления других записей (этот режим используется по умолчанию).

Если репликация не работает вовсе, лучше всего начать диагностику с главного сервера. Прежде всего, остановите процесс `slurpd` с помощью команды `kill` и измените какой-нибудь объект дерева каталога. Затем проверьте, увеличивается ли размер файла журнала репликации. Если размер файла не увеличивается, значит, главный сервер настроен неправильно. Далее, запустите `slurpd` с параметром командной строки `-d 255`. Данный режим отладки позволит отслеживать все действия `slurpd`, выполняемые при обработке журнала репликации. Ищите ошибки, в особенности, относящиеся к открытию файлов и контролю доступа.

Наконец, выполните на подчиненном сервере команду `loglevel auth sync`, чтобы проверить, не возникают ли какие-либо ошибки в процессе репликации (`slapd` ведет запись событий в журнал `syslog` от имени источника `local4`, поэтому, возможно вам придется добавить строку `local4.* /var/log/slapd.log` в файл `/etc/syslog.conf`).

## Репликация LDAP Sync

Репликация посредством `slurpd` является простым, открытым решением, но имеет ряд недостатков. Остановка главного сервера с целью синхронизации подчиненного сервера в лучшем случае может доставить неудобства, а в худшем – повлиять на качество предоставляемых услуг. Архитектура `slurpd` на основе извещений также имеет ряд ограничений. Для своего времени `slurpd` работал достаточно хорошо, но необходимо было создать нечто лучшее. В документации RFC 4533 описывается процесс синхронизации контента LDAP, реализованный в OpenLDAP посредством механизма LDAP Sync, также известного как `syncrepl`.

`Syncrepl` является оверлейной программой, встраиваемой между ядром `slapd` и базой данных. Все операции записи в дерево каталога отслеживаются механизмом `syncrepl`; при этом не требуется использовать какую-либо отдельную службу. За исключением механизма репликации и поддержки ролей (этот вопрос будет рассмотрен далее) принципы работы `syncrepl` аналогичны работе `slurpd`. Попытки обновления реплики отклоняются, а клиент получает возвращаемую ссылку на главный сервер.

Служба `syncrepl` запускается на подчиненном сервере, который теперь называется *получателем*. Главный сервер выступает в роли *provider*. При репликации посредством `syncrepl` получатель подключается к источнику для получения обновлений каталога. В наиболее часто использующемся режиме работы, который называется `refreshOnly`, получатель получает все измененные с момента последнего обновления

записи, запрашивает маркер, содержащий сведения о последней синхронизации, и затем отключается. При следующем подключении этот маркер передается источнику, которые возвращает те записи, которые были изменены с момента последней синхронизации.

Другим режимом работы syncrep1 является режим *refreshAndPersist*, работающий почти так же, как и *refreshOnly*, за исключением того, что получатель не отключается от источника после выполнения синхронизации, а продолжает получать все изменения, поддерживая подключение. Все изменения, произошедшие после первоначальной синхронизации, немедленно передаются от источника получателю.

### Настройка syncrep1

В листинге 12 представлена конфигурация сервера-источника для обоих режимов работы syncrep1 (*refreshOnly* и *refreshAndPersist*).

#### Листинг 12. Настройка репликации syncrep1 на сервере-источнике

```
overlay syncprov
syncprov-checkpoint 100 10
syncprov-sessionlog 100
```

Первая строка в листинге 12 включает возможность использования оверлейной программы syncprov. Оверлей должен настраиваться в отношении определенной базы данных, поэтому код в указанном листинге должен располагаться после настройки параметра *database*. Следующие две строки являются необязательными, однако повышают надежность. Стока *syncprov-checkpoint 100 10* указывает серверу сохранять значения параметра *contextCSN* на жесткий диск каждые 100 операций записи или каждые 10 минут. Параметр *contextCSN* является частью упомянутого ранее маркера, помогающего серверу-получателю получать изменения, произошедшие с момента завершения предыдущей репликации. Стока *syncprov-sessionlog 100* регистрирует все операции записи в журнале, хранящемся на жестком диске, что также помогает поддерживать цикл синхронизации.

Для получения дополнительной информации о настройке источника обратитесь к *man*-руководству *slapo-syncprov(5)*.

В листинге 13 приведена настройка сервера-получателя, участвующего в двустороннем процессе репликации.

#### Листинг 13. Настройка репликации syncrep1 в режиме refreshOnly на сервере-получателе

```
updateref ldap://masterserver.ertw.com
syncrep1 rid=1
provider=ldap://masterserver.ertw.com
type=refreshOnly
interval=00:01:00:00
searchbase="dc=ertw,dc=com"
bindmethod=simple
binddn="uid=replica1,dc=ertw,dc=com"
```

```
credentials=replica1
```

Так же, как и для команды `replica` в случае синхронизации посредством `slurpd`, для команды `syncrep1` требуется указать ссылку `updateref` предоставить информацию о дереве каталога, которое вы пытаетесь реплицировать. Также необходимо предоставить учетные данные, которые будут использоваться для репликации. На этот раз учетные данные указываются на стороне получателя, а уровень доступа к источнику, предоставляемый для этих учетных данных, должен обеспечивать возможность чтения реплицируемой части дерева каталога. Все изменения базы данных на сервере-получателе выполняются от имени `rootdn`.

Параметры `rid`, `provider`, `type` и `interval` относятся к `syncrep1`.

Параметр `rid` идентифицирует получателя на главном сервере.

Получатель должен иметь уникальный идентификатор (ID), который может принимать значение от 1 до 999. Параметр `provider` является ссылкой LDAP в формате URI, указывающей на сервер-источник.

Параметр `type` указывает, что синхронизация будет выполняться только в режиме `refreshOnly`, а параметр `interval` определяет периодичность синхронизации, равную одному часу. Значение параметра `interval` задается в формате `DD:hh:mm:ss`.

Запустите сервер-получатель с пустой базой данных, и его данные будут реплицироваться с сервером-источником каждый час.

Перевести синхронизацию в режим `refreshAndPersist` очень просто. Для этого удалите в листинге 13 строку `interval` и замените значение параметра `type` на `refreshAndPersist`.

### Фильтрация `syncrep1`

Следует заметить, что вам не обязательно может потребоваться реплицировать дерево LDAP целиком. Для того чтобы выбрать только необходимые для репликации данные, вы можете использовать следующие команды.

**Таблица 3. Команды для фильтрации трафика репликации**

Команда	Описание
<code>searchbase</code>	Различающееся имя, указывающее на узел дерева, с которого начнется репликация. При необходимости OpenLDAP заполнит нужные родительские узлы, чтобы обеспечить целостность дерева.
<code>scope</code>	Может принимать одно из значений: <code>sub</code> , <code>one</code> или <code>base</code> . Этот параметр определяет глубину репликации данных относительно начальной точки, указанной в параметре <code>searchbase</code> . Значением по умолчанию является <code>sub</code> ; это значение охватывает узел <code>searchbase</code> и все его дочерние узлы.
<code>filter</code>	Фильтр поиска LDAP, такой как <code>(objectClass/inetOrgPerson)</code> , определяющий набор записей для репликации.
<code>attrs</code>	Список атрибутов, которые будут извлечены из выбранных записей.

Также как и остальные опции `syncrep`, все вышеперечисленные параметры записываются в формате параметр=значение.

## Обеспечение безопасности LDAP

В этом разделе описывается материал по теме 303.4 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 4.

Из этого раздела вы узнаете, как:

- Защитить каталог с помощью SSL и TLS
- Настроить и сгенерировать сертификаты клиента и сервера
- Рассмотрите вопросы, касающиеся работы брандмауэра
- Настроить методы доступа без аутентификации
- Настроить методы аутентификации с использованием учетных данных "пользователь/пароль"

До этого момента доступ к `slapd` осуществлялся по незащищенным каналам с использованием открытых, нешифрованных паролей. Этот метод называется *простой* аутентификацией. В данном разделе мы рассмотрим методы шифрования соединений между клиентами и сервером.

### Использование SSL и TLS для защиты подключений

Вы можете быть знакомы с протоколом защищенных сокетов (Secure Sockets Layer, SSL) и протоколом защиты транспортного уровня (Transport Layer Security, TLS) как с протоколами, использующимися для защиты Web-транзакций. Всякий раз, используя ссылку URI с префиксом `https`, вы имеете дело с протоколом SSL или TLS. TLS является усовершенствованием протокола SSLv3 и в некоторых случаях имеет обратную совместимость с SSL. Из-за своей общей наследственности и совместимости эти два протокола часто рассматриваются вместе как единый протокол – SSL.

Протокол SSL использует сертификаты X.509 – файлы стандартизированного формата, содержащие цифровую подпись, поставляемую доверенной третьей стороной – центром сертификации (Certificate Authority, CA). Действительная цифровая подпись означает, что подписанные данные не были подделаны с момента их подписания. Если хотя бы один бит данных, содержащих цифровую подпись, будет изменен, эта подпись не сможет пройти проверку, и будет являться недействительной. Независимые участники процессов, такие как клиент и сервер, могут выполнять проверку цифровых подписей, поскольку на обоих из них настроены доверительные отношения с центром сертификации.

Сертификат сервера содержит информацию о владельце сервера, включающую в себя публичное Интернет-имя сервера. Таким образом, вы можете быть уверены, что подключаетесь именно тому серверу, к которому намереваетесь подключиться, поскольку его имя в точности соответствует имени, указанному в сертификате (при условии, что вы доверяете центру сертификации, проверившему и подписавшему сертификат). Кроме того, сертификат содержит открытый ключ сервера, который может использоваться для шифрования данных. Если данные зашифрованы таким ключом, то расшифровать и прочесть их сможет только владелец секретного (личного) ключа.

Открытый и секретный ключи формируют основу метода шифрования с использованием *открытого ключа*, или *ассиметричного шифрования*. Шифрование является ассиметричным по той причине, что информация, зашифрованная с помощью открытого ключа, может быть расшифрована только с помощью секретного ключа, и наоборот – информация, зашифрованная с помощью секретного ключа, может быть расшифрована только с помощью открытого. Для шифрования данных в традиционном понимании (например, сохранение в

тайне определенного сообщения) используется первый метод – открытый ключ является публичным, а личный ключ хранится в секрете. Благодаря механизму ассиметричного шифрования, зашифровать сообщение можно с помощью секретного ключа, а любой владелец открытого ключа сможет расшифровать его – именно так работают цифровые подписи.

После того как клиент подключается к серверу и получает его сертификат, он может проверить, соответствует ли имя сервера заявленному. Это помогает защититься от атак типа *man in the middle* (человек посередине). Открытый ключ можно использовать в определенном протоколе, работа которого завершается тем, что клиент и сервер договариваются об использовании общего секретного ключа, который не может быть определен ни одной наблюдающей за соединением стороной. В дальнейшем этот секретный ключ используется для шифрования оставшихся данных соединения между клиентом и сервером – этот процесс называется *симметричным* шифрованием, поскольку для шифрования и расшифровки данных используется один и тот же ключ. Разделение на асимметричный и симметричный методы шифрования существует по той причине, что последний из них работает на порядок быстрее. Шифрование на основе открытого ключа используется для аутентификации и установления договоренности об использовании общего секретного ключа, после чего в действие вступает симметричное шифрование.

Чтобы применить все вышеизложенное к OpenLDAP, необходимо создать сертификат сервера и настроить сервер на его использование. В нашем примере будет использоваться самоподписанный сертификат, а не сертификат, выпущенный центром сертификации. Это означает, что конечный сертификат будет подписан им же самим. Такой подход не обеспечивает того уровня доверия, как в случае использования подписи ЦС, но этого оказывается вполне достаточно для целей тестирования. В листинге 14 показан процесс генерации ключей.

#### Листинг 14. Генерация пары ключей TLS

```
[root@bob ssl]# openssl genrsa -out ldap.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
```

В листинге 14 показан процесс генерации ключа, запущенный путем выполнения команды `openssl genrsa`. Длина ключа составляет 1024 бита и на сегодняшний день считается достаточной для открытых ключей (обратите внимание, что использование более длинных ключей приводит к замедлению криптографических операций и может ввести в замешательство некоторых клиентов). Далее команда `openssl req` берет открытую часть только что созданной пары ключей, добавляет некоторую информацию о местоположении и запаковывает получившийся результат – запрос на подписание сертификата (Certificate Signing Request, CSR), который должен быть подписан центром сертификации. Этот процесс показан в листинге 15.

#### Листинг 15. Создание запроса на подписание сертификата

```
[root@bob ssl]# openssl req -new -key ldap.key -out ldap.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,  
If you enter '.', the field will be left blank.
```

```
----  
Country Name (2 letter code) [GB]:CA  
State or Province Name (full name) [Berkshire]:Manitoba  
Locality Name (eg, city) [Newbury]:Winnipeg  
Organization Name (eg, company) [My Company Ltd]:ERTW  
Organizational Unit Name (eg, section) []:Directory Services  
Common Name (eg, your name or your server's hostname) []:masterserver.ertw.com  
Email Address []:sean@ertw.com
```

```
Please enter the following 'extra' attributes  
to be sent with your certificate request
```

```
A challenge password []:  
An optional company name []:
```

Сформированный файл ldap.csr (а также платеж на довольно приличную сумму) можно послать в центр сертификации на подпись. Эта же процедура используется для генерации сертификата Web-сервера. Если вы посыаете ваш запрос на подпись в ЦС, убедитесь, что вся указанная вами информация написана без ошибок, аббревиатуры используются только для поля Country Name, а поле Common Name *в точности* совпадает с DNS-именем вашего сервера, которое будет использоваться клиентами для подключения к нему.

Вместо того чтобы подписать запрос CSR в центре сертификации, в нашем примере мы сделаем это самостоятельно, как показано в листинге 16.

### Листинг 16. Подписание запроса CSR

```
[root@bob ssl]# openssl x509 -req -days 1095 -in ldap.csr -signkey ldap.key -out ldap.cert  
Signature ok  
subject=/C=CA/ST=Manitoba/L=Winnipeg/O=ERTW/OU=Directory Services/  
CN=masterserver.ertw.com/emailAddress=sean@ertw.com  
Getting Private key
```

В листинге 16 показан процесс подписания ключа, запущенный путем выполнения команды **openssl x509**. Опция **-req** говорит команде openssl о том, что входным файлом является запрос на подписание сертификата. Срок действия сертификата составляет 1095 дней, или 3 года. Теперь у вас есть файлы ldap.key (личный ключ) и ldap.cert (сертификат и открытый ключ).

Прежде чем продолжить, добавьте в файл /etc/openldap/ldap.conf строку **TLS\_REQCERT allow**, которая указывает клиентским утилитам LDAP игнорировать факт использования самоподписанного сертификата. Если вы не добавите эту строку, а будете использовать параметры по умолчанию, сертификат будет считаться недействительным.

Настроить OpenLDAP на использование нового ключа и сертификата несложно. Предполагая, что сгенерированные ключи хранятся в папке /etc/openldap/ssl/, строки в листинге 17 настраивают ваш сервер на использование TLS-подключений после перезапуска slapd.

### Листинг 17. Настройка slapd на использование SSL

```
TLSCertificateFile /etc/openldap/ssl/ldap.cert
```

```
TLS Certificate Key File /etc/openldap/ssl/ldap.key
```

Команды в листинге 17 указывают службе slapd местоположение сертификата и личного ключа. Чтобы проверить работу сервера после выполнения вышеуказанных настроек, выполните команду `ldapwhoami -v -x -Z`, которая анонимно привязывается к безопасному порту. Если вы получите сообщение "success", значит, все работает правильно. В противном случае опция `-V` поможет вам установить причину возникшей ошибки.

Таким же способом вы можете сгенерировать сертификат клиента, что не является обязательной процедурой. В этом случае вместо использования команд, приведенных в листинге 17, добавьте строки `TLS_KEY` и `TLS_CERT` с соответствующими значениями в файл `ldap.conf`. Эти строки указывают на местоположения ключа и сертификата, соответственно. Клиентские сертификаты необходимы только в тех случаях, когда вам требуется выполнять идентификацию клиентов на основе их сертификатов.

### Вопросы, касающиеся работы брандмауэра

LDAP использует TCP-порт 389, а LDAPS (LDAP поверх SSL) – TCP-порт 636. Если между вашим сервером и клиентами работает брандмауэр, то для успешного подключения эти порты должны быть открыты. Клиенты всегда подключаются к серверам, а серверы могут подключаться к другим серверам в зависимости от вашей стратегии репликации.

### Правила iptables в OC Linux

Если для настройки брандмауэра на вашем сервере LDAP используются правила `iptables`, вам необходимо изменить их таким образом, чтобы позволить серверу принимать входящие подключения. Как правило, команд, приведенных в листинге 18, оказывается достаточно.

### Листинг 18. Добавление правил iptables для разрешения подключений к LDAP

```
iptables -A INPUT -p tcp --dport 389 -j ACCEPT  
iptables -A INPUT -p tcp --dport 636 -j ACCEPT
```

Команды листинга 18 работают, если у вас используется простая политика. Команда `-A INPUT` добавляет правило в таблицу INPUT, которая предназначена для проверки всех входящих пакетов. Вы можете добавить эти правила в начало списка (используя команду `-I INPUT` вместо `-A INPUT`) или использовать инструменты для работы с брандмауэром из состава вашего дистрибутива, чтобы открыть TCP-порт 389, а также порт 636 (если вам нужна функциональность LDAPS).

ваш брандмауэр Linux используется в качестве маршрутизатора (например, клиенты подключены к одному сетевому интерфейсу, а сервер LDAP – к другому), то вместо `INPUT` вы должны использовать цепочку `FORWARD`. Возможно, вы захотите определить интерфейс для входящих пакетов с помощью опции `-i`, например `-i eth0`, чтобы указать, что приниматься будут только пакеты, приходящие на интерфейс eth0. Если пакет был принят, то также будут приняты и ответные пакеты.

### Защита посредством использования оболочек TCP

ной из опций конфигурирования, доступной при компиляции пакета OpenLDAP, является опция `--enable-wrappers`, которая связывает конечные исполняемые файлы с библиотеками оболочек TCP (TCP Wrappers). Для принятия или отклонения клиентских подключений оболочки TCP используют два файла: `/etc/hosts.allow` и `/etc/hosts.deny` соответственно.

Прежде всего, проверьте с помощью команды `ldd /usr/sbin/slapd | grep libwrap`, использует ли slapd оболочки TCP. Если вы получите какой-либо ответ на вышеуказанный запрос, значит, оболочки TCP используются. В противном случае необходимо выполнить повторную компиляцию slapd с опцией `--enable-wrappers` или использовать правила iptables, как было описано выше.

Включив поддержку TCP Wrappers, вы можете запретить доступ к серверу LDAP для всех клиентов, добавив в файл `/etc/hosts.deny` строку `slapd: ALL`. После этого вы можете разрешить доступ с определенных IP-адресов, например, добавив строку `slapd: 192.168.1. , 127.0.0.1`, которая разрешает подключаться любым клиентам сети 192.168.1.0/24 или любым локальным клиентам. Обратите внимание на то, что отключение клиента оболочками TCP происходит следующим образом: сначала клиент подключается, а затем автоматически отключается. Этот процесс отличается от работы брандмауэра, когда пакет отбрасывается прежде, чем он достигнет службы slapd.

Формат файлов `hosts.allow` и `hosts.deny` позволяет разрешать и отклонять подключения многими различными способами. Для получения дополнительной информации обратитесь к `man`-руководству `hosts_access(5)`.

### Дополнительные сведения об аутентификации

До сих пор обсуждение вопросов аутентификации не выходило за рамки рассмотрения открытых паролей, определенных в файле `slapd.conf`, и методов простой аутентификации между клиентом и сервером. Проблема использования нешифрованных паролей решается с помощью команды `slappasswd`. Введите в командной оболочке команду `slappasswd`, после чего вам будет предложено ввести и подтвердить пароль. На выходе вы получите безопасный хэш пароля, такой как `{SSHA}oxmMsm9Ff/xVQ6zv+bgmMQjCUFL5x22+`. Математические методы гарантируют, что этот хэш необратим, хотя если кто-то получит его в свои руки, он может начать пробовать перебирать различные пароли и проверять, будет ли их хэш совпадать с исходным.

Вы уже имели дело с анонимной привязкой к каталогу, когда имя пользователя и пароль не указываются, а также с привязкой на основе аутентификации, когда клиент должен указать свои действующие имя пользователя и соответствующий ему пароль. Кроме этого OpenLDAP поддерживает привязку без аутентификации, когда указывается имя пользователя, но не указывается пароль. Привязка без аутентификации обычно отключена; для ее включения необходимо добавить в файл конфигурации строку `allow bind_anon_cred`. Если включена привязка без аутентификации, то все подключения, выполненные таким способом, считаются анонимными.

Альтернативу простой аутентификации составляет простая аутентификация и слой безопасности (Simple Authentication and Security Layer, SASL) – платформа для поддержки подключаемых модулей аутентификации и шифрования. Более подробно архитектура SASL будет рассмотрена в руководстве, которое должно скоро появиться, а пока ограничимся тем, что SASL предусматривает различные методы аутентификации – от открытых паролей до Kerberos.

При рассмотрении списков управления доступом в предыдущей части руководства было упомянуто, что доступ может быть предоставлен на основе метода подключения, определяющего фактор уровня защиты (Security Strength Factor, SSF). Нешифрованное соединение имеет SSF, равный 0, а шифрованному соединению обычно соответствует значение SSF, равное длине ключа шифрования. Таким образом, определенное правило ACL может содержать в условии `who` требование к уровню защиты подключения (строка `ssf=1`).

## **Оптимизация производительности сервера LDAP**

В этом разделе описывается материал по теме 303.5 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 2.

Из этого раздела вы узнаете, как:

- Оценивать производительность LDAP
- Выполнять тонкую настройку программного обеспечения для увеличения производительности
- Использовать индексы

OpenLDAP является базой данных. Вы обращаетесь к OpenLDAP с целью выполнения запроса или задания, после чего OpenLDAP ищет данные и возвращает их вам. Чтобы этот процесс занимал как можно меньше времени, вы должны распределить ресурсы так, чтобы они использовались наиболее эффективно, например, настроить кэширование часто используемой информации и индексацию баз данных.

### **Оценка производительности**

Прежде чем приступать к оптимизации slapd, вы должны оценить его текущую производительность. Для этого можно измерять время выполнения определенной операции в вашем приложении, пытаясь затем найти способ улучшить результат, или же можно вручную выполнить несколько запросов и оценить среднее время их выполнения. Измерять можно не только временные показатели; можно, например, попытаться уменьшить загрузку жесткого диска, если текущая конфигурация сервера LDAP приводит к большому количеству операций чтения и записи.

И в том, и в другом случае полезно произвести несколько измерений различных показателей до и после изменений конфигурации. В этом вам могут помочь следующие команды:

- **vmstat** – показывает статистику ввода/вывода и загрузку центрального процессора, а именно, время, затрачиваемое на выполнение процессов пользователя, и время ожидания
- **iostat** – показывает более подробную информацию об операциях чтения и записи на жесткий диск, а также о загрузке дискового контроллера
- **ps** – показывает статистику использования памяти процессом slapd (использование большого объема памяти само по себе не является чем-то плохим, но важно убедиться, что вы не используете больше ОЗУ, чем установлено в системе)
- **time** – команда для временной оценки различных операций командной строки

### **Оптимизация работы демона**

Оптимизация всегда требует принятия компромиссных решений. Часто вы увеличиваете объем системных ресурсов (обычно это оперативная память или жесткие диски), выделяемый под определенный процесс, чтобы он выполнялся быстрее. Это приводит к тому, что под остальные процессы выделяется меньше ресурсов. Точно так же, если определенный процесс выполняется быстрее, он зачастую потребляет больше ресурсов, таких как циклы центрального процессора или операции чтения/записи на диск, которые при этом становятся недоступными для остальных процессов.

Необходимость находить компромиссы может существовать и на уровне приложений. Пожертвовав некоторой производительностью операций записи, вы в большинстве случаев можете довольно существенно повысить производительность операций чтения. Также вы можете повысить быстродействие вашего приложения, отключив некоторые функции безопасности, такие как ведение журналов транзакций. При этом в случае сбоя вы можете столкнуться с тем, что вам придется восстанавливать вашу базу данных из резервной копии; однако только вам предстоит решить, насколько приемлемым является этот компромисс.

Большинство людей используют базу данных Berkeley Database (BDB), которая основана на быстрой встроенной БД Sleepycat Berkeley Database, в настоящее время принадлежащей корпорации Oracle. Эта база данных не поддерживает язык запросов, а основана на поиске записей в хэш-таблицах. Оптимизация BDB может производиться в двух файлах: файл конфигурации slapd.conf и специальный файл, используемый исполняемыми модулями BDB.

## Конфигурационные директивы файла slapd.conf

База данных BDB не является отдельной серверной службой, как большинство SQL-серверов, а линкуется вместе с исполняемыми файлами, которые ее используют. По существу, за некоторые аспекты работы базы данных BDB отвечает приложение, использующее ее. Все директивы, которые могут использоваться в файле slapd.conf, описаны в *ман-руководстве slapd-bdb(5)*; мы же рассмотрим только наиболее важные из них.

Так же, как и многие другие базы данных SQL, базы данных BDB записывают все изменения в журналы транзакций для обеспечения надежности, а также хранят данные в памяти для уменьшения количества операций записи на диск. Операция, которая выгружает содержимое памяти на жесткий диск и выполняет запись в журнал транзакций, называется *контрольной точкой*. Команда **checkpoint** указывает slapd, как часто нужно выполнять выгрузку данных на диск, оперируя такими параметрами, как количество килобайт данных и количество минут, прошедших с момента последней контрольной точки. Добавление строки **checkpoint 128 15** в файл slapd.conf означает, что данные будут выгружаться при достижении объема в 128 килобайт или, как минимум, каждые 15 минут. По умолчанию никакие операции с контрольными точками не выполняются, что равносильно использованию команды **checkpoint 0 0**.

Записи, к которым происходят частые обращения, могут кэшироваться в ОЗУ для ускорения доступа к ним. По умолчанию кэшируются 1000 записей. Для изменения этого значения используйте команду **cachesize** с указанием количества записей. Чем больше значение параметра **cachesize**, тем больше вероятность того, что запись будет помещена в оперативную память, однако, при этом slapd расходует большее количество оперативной памяти. Выбор значения этого параметра зависит от того, сколько различных записей содержится в дереве каталога, а также от шаблона доступа. Убедитесь, что объема оперативной памяти достаточно для того, чтобы уместить элементы (такие как список пользователей), обращения к которым происходят наиболее часто.

Команда **cachesize** аналогична команде **idlcachesize** и определяет, сколько памяти отводится под кэширование индексов. Выбор значения этого параметра зависит от того, сколько индексов у вас определено (это будет обсуждаться позже), но разумно указывать то же самое значение, что и для параметра **cachesize**.

## Оптимизация баз данных BDB

Как было замечено ранее, некоторые конфигурационные параметры BDB содержатся в отдельном файле, который считывается исполняемыми модулями программы и игнорируется службой slapd. Этот файл называется **DB\_CONFIG** и расположен в той же папке, что и ваша база данных. Самым важным параметром в этом файле является параметр **set\_cachesize**, который устанавливает размер внутреннего кэша BDB (отдельно от кэша slapd). Эта команда имеет следующий формат: **set\_cachesize <GigaBytes> <Bytes> <Segments;>**. Параметры **GigaBytes** и **Bytes** относятся к размеру кэша (эти два параметра суммируются), а параметр **Segments** позволяет вам разделять кэш между отдельными блоками памяти для преодоления ограничений 32-разрядной адресации (значения этого параметра, равные как 0, так и 1, работают одинаково, позволяя использовать только один сегмент памяти).

Чтобы задать кэш размером в 1 Гб, используйте команду `set_cachesize 1 0 0`.

Самый простой способ определить оптимальный размер кэша BDB – это посмотреть статистику использования кэша в работающей системе и при необходимости увеличить его размер. Статистику использования кэша BDB можно посмотреть с помощью команды `db_stat -h /path/to/database -m`. Наиболее значимая информация содержится в первых 20 строках вывода этой команды. Если из кэша удаляется большое количество страниц или количество страниц, найденных в кэше, составляет менее 95% от общего количества, подумайте об увеличении его размера. В некоторых дистрибутивах команда `db_stat` может называться `slapd_db_stat`, что позволяет обособить ее от системных библиотек и инструментов BDB.

Кроме оптимизации кэша вам необходимо обеспечить наблюдение за журналами транзакций. Укажите путь к журналам транзакций в качестве значения параметра `set_lg_dir`. Если вы сможете разместить базу данных и журнал транзакций на отдельных физических дисках или дисковых массивах, быстродействие системы существенно повысится.

Несмотря на то, что BDB является простой базой данных, существует необходимость блокировок файлов для выполнения операций записи. Обычно количество блокировок по умолчанию достаточно велико, но вам следует отслеживать максимальное количество используемых блокировок с помощью команды `db_stat -h /path/to/database -c`.

Блокировки BDB делятся на три типа, для каждого из которых ведется отдельная статистика: `lockers`, `locks` и `lock objects`. Разница между этими тремя типами несущественна, тем не менее, максимальное количество блокировок каждого типа задается с помощью трех отдельных параметров – `set_lk_max_lockers`, `set_lk_max_locks` и `set_lk_max_objects` соответственно.

Каждый раз, когда вы вносите изменения в файл `DB_CONFIG`, вы должны перезапустить `slapd`. В листинге 19 приведен пример конфигурационного файла `DB_CONFIG` с использованием всех вышеупомянутых директив.

### Листинг 19. Пример файла `DB_CONFIG`

```
# 256K cache
set_cachesize 0 268435456 0
set_lg_dir /var/log/openldap
set_lk_max_lockers 1000
set_lk_max_locks 1000
set_lk_max_objects 1000
```

### Индексация базы данных

Большинство операций LDAP выполняют поиск определенных атрибутов, таких как имя пользователя, телефонный номер или адрес электронной почты. Без использования дополнительных средств `slapd` должен выполнять поиск, перебирая каждую запись. При добавлении индекса к определенному атрибуту создается специальный файл, который

позволяет slapd находить данные гораздо быстрее. Недостатками использования индексов являются более низкая скорость записи в базу данных, а также увеличение загрузки жесткого диска и оперативной памяти. По этой причине лучше всего индексировать только те атрибуты, обращения к которым происходят чаще всего.

В зависимости от типа выполняемого поиска в OpenLDAP поддерживаются различные типы индексов. Все они перечислены в таблице 4.

**Таблица 4. Типы индексов OpenLDAP**

Тип	Ключевое слово	Описание	Пример запроса
Наличие (Presence)	pres	Используется для выяснения существования атрибута.	uid=*
Эквивалентность (Equality)	eq	Используется для нахождения определенного значения.	uid=42
Вхождение (Substring)	sub	Используется для нахождения строки, содержащейся в значении атрибута. Помимо основного типа <b>sub</b> вы можете использовать три его подтипа, оптимизированные для различных условий. Индекс вхождения, использующийся для нахождения строки, содержащейся в начале значения атрибута. Индекс вхождения, использующийся для нахождения строки, содержащейся в середине значения атрибута. Индекс вхождения, использующийся для нахождения строки, содержащейся в конце значения атрибута.	cn=Sea n*
Подобие (Approximate)	approx	Используется для нахождения значений, звучащих подобно указанной строке поиска.	cn~=Ja son

Чтобы создать индекс для атрибута, используйте следующий синтаксис: `index [attrlist] [indices]`, где `[attrlist]` – это разделенный запятыми список атрибутов, а `[indices]` – разделенный запятыми список типов индексов, перечисленных в таблице 4. Можно использовать несколько строк с директивами `index`. Ключевое слово `default` определяет список типов индексов по умолчанию, который будет использоваться в том случае, когда опущен параметр `[indices]`. Ознакомьтесь с индексами, определенные в листинге 20.

## Листинг 20. Примеры индексов

```
index default eq,sub
index entryUUID,objectClass eq
index cn,sn,mail eq,sub,subinitial,subany,subfinal
```

```
index userid,telephonenumber  
index ou eq
```

В первой строке листинга 20 определен список типов индексов по умолчанию, включающий в себя индексы эквивалентности и вхождения. Во второй строке создаются индексы эквивалентности для атрибутов entryUUID (полезно для повышения производительности syncerl) и objectClass (для общего поиска). В третьей строке создаются индекс эквивалентности и все типы индексов вхождения для атрибутов cn, sn и mail, поскольку по этим полям часто выполняется поиск с использованием различных специальных символов. Для атрибутов userid и telephonenumber создаются индексы по умолчанию, поскольку не указаны никакие дополнительные параметры. Наконец, для атрибута ou создается индекс эквивалентности.

После определения индексов вы должны перестроить их, остановив slapd и выполнив команду slapindex от имени пользователя ldap (если вы работаете под учетной записью root, не забудьте назначить пользователя ldap владельцем всех файлов, расположенных в каталоге базы данных, после выполнения команды slapindex). Запустите slapd, после чего ваши индексы начнут использоваться.

## Конфигурирование

В этом разделе описывается материал по теме 303.6 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 2.

Из этого раздела вы узнаете, как:

- Узнаете о конфигурационных директивах slapd.conf
- Узнаете об описаниях базы данных в файле slapd.conf
- Управлять slapd и использовать его параметры командной строки
- Анализировать файлы журналов slapd

Содержимое файла slapd.conf было широко рассмотрено ранее в этом руководстве, а также в предыдущем руководстве. Определенный интерес представляют параметры командной строки и команды для работы с файлами журналов slapd.

## Параметры командной строки

Запуск slapd без каких-либо аргументов является наиболее простым. При таком запуске slapd считывает конфигурационный файл по умолчанию, переходит в фоновый режим работы и освобождает терминал.

В таблице 5 перечислены некоторые полезные параметры командной строки.

**Таблица 5. Параметры командной строки slapd**

Параметр	Значение	Описание
-d	Целое число	Запускает slapd в режиме расширенной отладки; при этом slapd работает на переднем плане.
-f	Имя файла	Указывает альтернативный конфигурационный файл.
-h	Список URL	Указывает IP-адреса и порты, на которых

		работает slapd.
-s	Уровень отладочной информации syslog	Указывает уровень syslog, использующийся для вывода отладочной информации.
-l	Целое число	Указывает локальное имя источника syslog (такое как local4), используемое для вывода отладочной информации.
-u	Имя пользователя	Запускает slapd от имени указанного пользователя.
-g	Имя группы	Запускает slapd от имени указанной группы.

Список URL позволяет вам привязывать slapd к нескольким различным интерфейсам.

Например, команда `-h "ldap://127.0.0.1/ ldaps://"` указывает, что slapd будет прослушивать TCP-порт 389 (нешифрованные подключения LDAP) только на loopback-интерфейсе, а TCP-порт 636 (защищенные подключения) – на всех интерфейсах. Вы также можете изменять номера портов: например, команда `ldap://:5565/` указывает, что slapd будет прослушивать TCP-порт 5565 (нешифрованные подключения) на всех интерфейсах.

## Регистрация событий

Для регистрации событий slapd использует демон ОС Unix под названием syslog. По умолчанию все сообщения посылаются от имени источника LOCAL4. По этой причине вам необходимо добавить в файл `syslog.conf`, как минимум, строку `local4.* /var/log/openldap.log` для записи сообщений в файл `/var/log/openldap.log`. Далее, команда `loglevel` в файле `slapd.conf` указывает slapd, какие типы сообщений необходимо регистрировать. Эти типы перечислены в таблице 6.

**Таблица 6. Типы регистрируемых сообщений slapd**

Ключевое слово	Соответствующее целочисленное значение	Описание
trace	1	Трассировать вызовы функций
packet	2	Отладка обработки пакетов
args	4	Тщательная отладочная трассировка
conns	8	Управление соединением
BER	16	Печать принятых и отправленных пакетов
filter	32	Обработка фильтра поиска
config	64	Обработка конфигурационного файла
ACL	128	Обработка списка контроля доступа
stats	256	Регистрировать статистику соединения/обработки/результатов
stats2	512	Регистрировать статистику отправленных элементов
shell	1024	Печать коммуникаций с shell механизмом баз данных
parse	2048	Печать отладки анализа элемента
sync	16384	Репликация LDAPSsync

Для параметра `loglevel` вы можете использовать список ключевых слов или целочисленных значений, разделенных пробелами, а также сумму целочисленных значений.

Например, каждая из команд `loglevel args ACL`, `loglevel 4 128` и `loglevel 132` включает регистрацию тщательной отладочной трассировки и обработки списков ACL.

## Резюме

Из этого руководства вы узнали о списках контроля доступа, репликации, безопасности, оптимизации, а также о других основных аспектах конфигурирования LDAP.

Списки контроля доступа (ACLs) определяют, кому и к каким элементам предоставляется доступ определенного уровня. Для настройки списков контроля доступа вы должны использовать следующий синтаксис: `access to <what> [ by <who> [ <access> ] [ <control> ] ]+`. Для определения условия `what` вы можете использовать различные конструкции, включая прямые соответствия и регулярные выражения. Для определения условия `who` вы можете использовать такие ключевые слова, как `self`, `users` и `anonymous` в дополнение к вышеупомянутым конструкциям. Кроме того, в условии `who` можно определять такие параметры, как фактор уровня защиты подключения или номер сети, из которой подключается пользователь.

Репликация позволяет обеспечить идентичность данных удаленного и главного серверов LDAP. Существует два метода репликации: `slurpd` и `syncerl`. При выполнении репликации посредством `slurpd` на главном сервере работает отдельный демон, передающий все изменения на подчиненные серверы. Подчиненные серверы должны запускаться только тогда, когда на них имеется копия данных главного сервера, что приводит к простоям последнего. При выполнении репликации посредством `syncerl` на источнике данных (главный сервер) выполняется оверлейная программа, обрабатывающая задачи репликации. Получатели данных (подчиненные серверы) подключаются к источнику и скачивают все изменения. Если получатель скачивает изменения периодически, он работает в режиме `refreshOnly`. Если же после загрузки обновлений получатель не разрывает соединение, то он работает в режиме `refreshAndPersist` и продолжает получать обновления, выполняющиеся на сервере-источнике.

Протоколы TLS и SSL позволяют устанавливать шифрованные соединения между клиентом и сервером, а также шифровать трафик репликации. Для использования TLS вы должны сгенерировать ключи сервера, а затем подписать их в центре сертификации. Для передачи обычного трафика LDAP используется TCP-порт 389, а для передачи шифрованного трафика LDAP – TCP-порт 636; с учетом этого должны быть соответствующим образом настроены ваши брандмауэры.

Оптимизация производительности включает в себя распределение системных ресурсов под различные области для промежуточного хранения данных, а также использование индексов для наиболее часто используемых атрибутов. Управление системными ресурсами производится посредством редактирования двух файлов: `slapd.conf` и `DB_CONFIG`. В зависимости от задач оптимизации поиска, существуют индексы следующих типов: индексы наличия, эквивалентности, вхождения и подобия.

Большинство параметров работы `slapd` задается в файле `slapd.conf`, поэтому существует лишь несколько опций командной строки; эти опции задают номера портов, на которых работает `slapd`, имя пользователя, в контексте которого он работает, а также некоторые параметры регистрации событий. Информация, которую `slapd` регистрирует в журналах событий, определяется директивой `loglevel` в файле `slapd.conf`.

На данном этапе вы обладаете достаточными знаниями для того, чтобы установить, настроить и управлять работающим сервером OpenLDAP, а также разбираться в вопросах защиты, репликации и оптимизации производительности. В следующих двух руководствах будут рассмотрены такие приложения LDAP, как интеграция LDAP с почтовой системой и

системой аутентификации, и выполнение поиска в дереве каталога из командной строки.

## Ресурсы

### Научиться

- Оригинал руководства "[LPI exam 301 prep, Topic 303: Configuration](#)" (EN).
- Изучите предыдущее руководство в серии 301 - "[Подготовка к экзамену LPI 301, Тема 302: установка и разработка](#)" (developerWorks, декабрь 2007 г.) или [все руководства в серии 301](#) (EN).
- Чтобы познакомиться с основами Linux и подготовиться к сертификации в качестве системного администратора, ознакомьтесь со всей [серий руководств для подготовки к экзаменам LPI](#).
- Ответ на вопрос [Как определить правильный размер кэша базы данных BDB?](#) (EN) в ответах на часто задаваемые вопросы по OpenLDAP.
- Для более глубокого понимания прочтите статью Википедии [Нормальная форма Бэкуса-Наура \(БНФ\)](#) (EN), содержащую несколько примеров. Вы можете быть знакомы с нотацией БНФ, если вы изучали формат LDIF (LDAP Data Interchange Format) или читали справочные страницы OpenLDAP.
- [Руководство администратора OpenLDAP](#) содержит главу [Управление доступом](#) (EN), в которой детально описан синтаксис. Справочная страница `slapd.access(5)` будет хорошим дополнением к вышеупомянутому руководству.
- Также прочтайте главы [Репликация LDAP Sync](#) и [Репликация посредством slurpd](#) в [Руководстве администратора OpenLDAP](#) (EN), в которых детально описывается работа обоих механизмов репликации.
- Спецификация [RFC 4533 \(Операция синхронизации контента протокола Lightweight Directory Access Protocol\)](#) (EN), разработанная сообществом OpenLDAP Foundation и корпорацией IBM, описывает метод синхронизации серверов LDAP, являющийся более эффективным по сравнению с синхронизацией `Slurpd`.
- Очень советую онлайновую книгу [LDAP для больших ученых](#) (EN), несмотря на то, что работа над ней ещё не закончена.
- В [разделе Linux сайта developerWorks](#) можно найти дополнительные ресурсы для разработчиков Linux, а также [самые популярные среди наших читателей статьи и руководства](#) (EN).
- Посмотрите все [советы по Linux](#) (EN) и [руководства Linux](#) на сайте developerWorks.
- Следите на последними новостями на портале [Web-трансляций и технических мероприятий developerWorks](#) (EN).

### Получить продукты и технологии

- Утилита [Firewall Builder](#) упрощает задачу создания правил iptables, предоставляя в ваше распоряжение графический интерфейс и набор инструментов для развертывания обновлений на ваших брандмауэрах.
- Загрузите [OpenLDAP](#).
- [phpLDAPadmin](#) - инструмент администрирования LDAP на базе Web. Если вам больше нравится графический интерфейс, вам стоит посмотреть на [Luma](#) (EN).

- Используйте в своем следующем проекте разработки для Linux [ознакомительные версии программного обеспечения IBM](#), которые можно скачать непосредственно с developerWorks (EN).

# Подготовка к экзамену LPI 301: Тема 304.

## Использование

*Senior Level Linux Professional (LPIC-3)*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux Professional Institute на квалификацию профессионала Linux высокого уровня (LPIC-3). В этом руководстве, четвертом из [серии из шести руководств](#), Шон расскажет о том, как следует выполнять поиск по дереву каталога LDAP и использовать утилиты командной строки. Также вы узнаете о том, как необходимо настраивать Microsoft Outlook для выполнения запросов к вашему серверу LDAP.

[Больше статей из этой серии](#)

**Дата:** 20.01.2009

**Уровень сложности:** средний

### Предисловие

Узнайте, чему могут научить вас эти руководства, и как получить от них больше пользы.

### Об этой серии руководств

Институт [Linux Professional Institute](#) (LPI) сертифицирует системных администраторов Linux® по трём уровням: *младший уровень* (также называемый "уровень сертификации 1"), *углубленный уровень* (также называемый "уровень сертификации 2") и *высший уровень* (также называемый "уровень сертификации 3"). Чтобы получить сертификацию на уровне 1, нужно сдать экзамены 101 и 102. Чтобы получить сертификацию на уровне 2, нужно сдать экзамены 201 и 202. Чтобы получить сертификацию на уровне 3, у вас должна быть действующая сертификация на углубленном уровне и сдан экзамен 301 ("основной"). Кроме того, на высоком уровне от вас может потребоваться сдать дополнительные специализированные экзамены.

Сайт developerWorks предлагает руководства, которые помогут вам подготовиться к пяти экзаменам для младшего, углубленного и высокого уровня. В каждом экзамене охватывается несколько тем, и по каждой теме на developerWorks есть соответствующий учебник для самостоятельного изучения. В таблице 1 перечислены шесть тем и соответствующие им руководства developerWorks для экзамена LPI 301.

**Таблица 1. Экзамен LPI 301: руководства и темы**

Тема экзамена	Руководство developerWorks	Краткое описание руководства
Тема 301	<a href="#">Подготовка к экзамену LPI 301: понятия, архитектура и модель</a>	Узнайте о понятиях и архитектуре LDAP, о том, как проектировать и внедрять каталог LDAP, а также о схемах.
Тема 302	<a href="#">Подготовка к экзамену LPI 301: установка и разработка</a>	Узнайте, как устанавливать, настраивать и использовать программное обеспечение OpenLDAP.

Тема 303	<a href="#"><u>Подготовка к экзамену LPI 301: конфигурирование</u></a>	Узнайте более подробно о том, как настраивать программное обеспечение OpenLDAP.
Тема 304	Подготовка к экзамену LPI 301: использование	(Это руководство) Узнайте, как выполнять поиск по дереву каталога LDAP и использовать утилиты OpenLDAP. См. подробные <a href="#"><u>цели</u></a> .
Тема 305	Подготовка к экзамену LPI 301: интеграция и миграция	Появится в ближайшее время.
Тема 306	Подготовка к экзамену LPI 301: планирование пропускной способности	Появится в ближайшее время.

Чтобы сдать экзамен 301 (и получить сертификацию третьего уровня), вам следует:

- обладать несколькими годами опыта установки и поддержки Linux на большом числе компьютеров, используемых в различных целях
- обладать опытом интеграции с различными технологиями и операционными системами
- обладать профессиональным опытом или пройти профессиональную подготовку специалиста Linux корпоративного уровня (включая опыт, полученный при работе в другой роли)
- знать администрирование Linux на углубленном и высоком уровне, включая установку, управление, обеспечение безопасности, решение возникающих проблем и техническое обслуживание
- уметь использовать инструменты с открытым исходным кодом для проведения измерений, необходимых для планирования пропускной способности и решения проблем с ресурсами
- иметь профессиональный опыт применения LDAP для интеграции с сервисами UNIX® и Microsoft® Windows®, в том числе Samba, Pluggable Authentication Modules (PAM), электронной почтой и Active Directory
- уметь планировать, проектировать, разрабатывать, строить и реализовывать полную среду с использованием Samba и LDAP, а также проводить измерения для планирования производительности и оценки безопасности служб
- уметь создавать сценарии на Bash или Perl или знать как минимум один язык системного программирования (например, C).

Для дальнейшей подготовки к сертификации уровня 3 ознакомьтесь с [серий руководств для подготовки к экзамену 301 Института LPI](#), а также со всей [серий руководств developerWorks для подготовки к экзаменам LPI \(EN\)](#).

Институт Linux Professional Institute не дает рекомендаций по каким-либо конкретным материалам и методикам для подготовки к экзаменам, разработанным сторонними лицами.

## Об этом руководстве

Добро пожаловать в четвертое [из шести руководств](#), призванных помочь вам подготовиться к сдаче экзамена LPI 301, - "Использование". Из этого руководства вы узнаете о том, как выполнять поиск по дереву каталога LDAP, использовать утилиты командной строки для поиска и администрирования, а также о том, как настраивать сторонние приложения на

использование вашего дерева LDAP в качестве службы "белых страниц" (Whitepages). Это руководство организовано в соответствии с целями LPI по этой теме. Условно говоря, чем выше вес цели, тем больше вопросов по этой теме будет на экзамене.

## Цели

В таблице 2 подробно перечислены цели этого руководства.

**Таблица 2. Использование: цели экзамена, описанные в этом руководстве**

Цель экзамена LPI	Вес	Цели	Краткое описание цели
304.1 <a href="#">Поиск по каталогу</a>	2		Используйте расширенные возможности поиска по каталогу LDAP
304.2 <a href="#">Утилиты командной строки LDAP</a>	4		Используйте различные утилиты командной строки для поиска, модификации и администрирования сервера LDAP
304.3 <a href="#">Служба "белых страниц"</a>	1		Используйте ваш сервер LDAP в качестве службы "белых страниц" для таких приложений, как Microsoft Outlook®

## Необходимые условия

Чтобы извлечь максимум пользы из этого руководства, вы должны обладать глубокими знаниями Linux и иметь работающую Linux-систему, на которой вы сможете практиковаться в выполнении рассматриваемых задач.

Если ваши базовые знания Linux немного устарели, вы можете сначала ознакомиться с [руководствами для экзаменов LPIC-1 и LPIC-2](#).

Различные версии программ могут выводить данные в различных форматах, поэтому результаты, полученные вами, могут отличаться от листингов и рисунков, приведенных в этом руководстве.

## Требования к системе

Чтобы выполнить примеры, приведенные в этом руководстве, вам потребуется рабочая станция под управлением Linux с пакетом OpenLDAP и поддержкой PAM. Большинство современных дистрибутивов удовлетворяют этим требованиям.

# Подготовка к экзамену LPI 301: Тема 304. Использование

*Senior Level Linux Professional (LPIC-3)*

[Шон Уолберг](#), старший сетевой инженер, P.Eng

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux Professional Institute на квалификацию профessionала Linux высокого уровня (LPIC-3). В этом руководстве, четвертом из [серии из шести руководств](#), Шон расскажет о том, как следует выполнять поиск по дереву каталога LDAP и использовать утилиты командной строки. Также вы узнаете о том, как необходимо настраивать Microsoft Outlook для выполнения запросов к вашему серверу LDAP.

## Поиск по каталогу

В этом разделе описывается материал по теме 304.1 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 2.

Из этого раздела вы узнаете, как:

- Использовать инструменты поиска OpenLDAP на базовом уровне
- Использовать расширенные инструменты поиска OpenLDAP
- Оптимизировать поисковые запросы LDAP
- Использовать фильтры поиска и их синтаксис

Данные дерева каталога полезны только тогда, когда вы можете выполнять поиск нужных вам записей. LDAP содержит ряд мощных инструментов, позволяющих вам извлекать информацию из каталога.

## Основы поиска

Для выполнения поиска по дереву каталога вам необходима следующая информация:

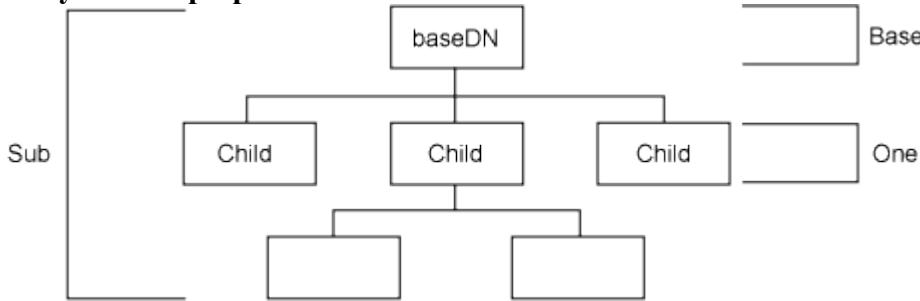
1. Учетные данные на сервере, содержащем каталог
2. Отличительное имя (Distinguished Name, DN) в дереве каталога, на основе которого вы будете выполнять поиск
3. Область поиска
4. Фильтр поиска

Из [предыдущих руководств](#) этой серии вы уже знаете об учетных данных для сервера. Вы можете выполнять анонимную привязку к каталогу, не используя учетные данные, либо использовать в качестве учетных данных отличительное имя и пароль для определенной записи. Если сервер сможет опознать учетные данные и признает их действительными, он позволит вам выполнять поиск.

Отличительное имя (DN), на основе которого вы выполняете поиск, называется *отличительным именем базового объекта* (Base DN). Результатом поиска будет являться либо само отличительное имя базового объекта, либо его дочерние элементы. Если DN базового объекта – это `ou=people,dc=ertw,dc=com`, то результатом поиска может быть `cn=Sean Walberg,ou=people,dc=ertw,dc=com`, но не `cn=Users,ou=Groups,dc=ertw,dc=com`, поскольку последний элемент выходит за рамки DN базового объекта, по которому вы выполняете поиск.

Область поиска указывает, по каким элементам отличительного имени базового объекта будет выполняться поиск. Например, вы можете ограничить область поиска из соображений производительности или потому, что нужную вам информацию содержат только определенные дочерние элементы базового объекта. Область поиска по умолчанию, или *subordinate* (обычно указывается сокращенно как *sub*), включает в себя имя базового объекта и все его дочерние объекты. Вы можете выполнять поиск только имени базового объекта в области *base* и всех его дочерних объектов. Вы можете выполнять поиск только имени базового объекта в области *base*, например, в тех случаях, когда вы хотите проверить существование элемента. Область поиска *one* указывает, что поиск выполняется только в пределах списка дочерних объектов первого уровня, и исключает из поиска все дочерние объекты последующих уровней, а также сам базовый объект. На рисунке 1 изображено дерево и его элементы, которые будут включены в три различные области поиска.

**Рисунок 1. Три различные области поиска**



Самым мощным (и сложным) инструментом поиска является фильтр поиска. Учетные данные, отличительное имя базового объекта и область поиска ограничивают набор записей, по которому будет выполняться поиск, а фильтр поиска является *запросом*, проверяющим каждую запись и возвращающим только те из них, которые удовлетворяют заданным критериям.

### Простые фильтры поиска

Фильтры поиска заключаются в круглые скобки, внутри которых содержится одна пара **атрибут=значение**. Простым фильтром поиска является **(objectClass/inetOrgPerson)**, выполняющий поиск любых записей класса **inetOrgPerson**. Сам атрибут не является чувствительным к регистру, а значение может быть чувствительным или нечувствительным к регистру в зависимости от того, как атрибут определен в схеме. Вспомните материал руководства [Подготовка к экзамену LPI 301: понятия, архитектура и модель](#), в котором говорилось о том, что схема определяет атрибуты и их свойства. Одним из свойств атрибута является чувствительность к регистру при сравнении.

Поиск подстроки выполняется с использованием символа звездочки (\*). Чтобы найти все записи, начинающиеся с **Sean**, выполните поиск подстроки **(Sean\*)**. Символ звездочки может располагаться в любом месте строки; например, результатом поиска подстроки **(\* Walberg)** будут являться все записи, оканчивающиеся на **Walberg**, а результатом поиска **S\*Wa\*berg** – все записи, начинающиеся на **S**, оканчивающиеся на **berg** и содержащие символы **Wa** где-либо в середине. Вы можете использовать этот фильтр, например, для поиска имени автора, которое вы не можете назвать точно (**Sean** или **Shawn**, **Walberg** или **Wahlberg**).

Самая общая форма использования оператора звездочки, **атрибут=\***, проверяет существование указанного атрибута. Например, чтобы найти все записи, содержащие адреса электронной почты, вы можете использовать фильтр **(mail=\*)**.

### Операции AND, OR и NOT

Вы можете выполнять логические операции AND ("И") и OR ("ИЛИ") с помощью операторов "&" и "|" соответственно. В строках поиска LDAP операторы помещаются перед условиями, таким образом, вы можете встретить фильтры, подобные следующим.

### Листинг 1. Пример фильтров поиска с использованием AND и OR

```
((objectClass/inetOrgPerson)(objectClass posixAccount))  
&((objectClass=*)(cn=Sean*)(ou=Engineering))  
&(|(objectClass/inetOrgPerson)  
(objectClass posixAccount))(cn=Sean*)
```

Первая строка поиска в листинге 1 выполняет поиск любых записей класса **inetOrgPerson** или **posixAccount**. Обратите внимание на то, что каждое условие

поиска заключено в круглые скобки, и что оператор OR () вместе с этими двумя условиями также заключен в еще одну пару скобок.

Вторая строка поиска похожа на первую, но начинается с оператора AND, а не OR. Здесь должны выполняться три различных условия, которые следуют за символом амперсанда, и каждое из которых заключено в отдельную пару скобок. Первому условию, **objectClass=\***, удовлетворяют все записи, класс объекта (objectClass) которых определен (при этом сам класс может быть любым). Поиск записей любого класса часто используется в тех случаях, когда вы хотите выбрать все записи, но при этом должны использовать фильтр поиска. Второму условию удовлетворяют все записи, начинающиеся с **Sean**.

В третьей строке поиска совместно используются операторы AND и OR. Эта строка ищет любые записи класса **inetOrgPerson** или **posixAccount**, обычное имя (CN) которых начинается с **Sean**.

Для логической операции NOT ("НЕ") используется знак восклицания (!), подобно операциям AND и OR. Логическая операция NOT имеет только один аргумент, поэтому за знаком восклицания может следовать только один набор скобок. В листинге 2 показаны примеры правильного и неправильного использования логической операции NOT.

## Листинг 2. Как нужно и как не нужно использовать логическую операцию NOT

```
(!cn=Sean)      # неправильно, ! применяется к условию внутри ()
(! (cn=Sean))  # правильно
(! (cn=Sean) (ou=Engineering)) # неправильно,
операция может быть выполнена только для одного условия
(!((&cn=Sean*)(ou=Engineering))) # правильно, операция NOT выполняется для условия AND
```

В четвертом примере листинга 2 операция NOT применяется к условию AND. Таким образом, это правило возвращает только те записи, которые не удовлетворяют обоим условиям в структуре AND. Будьте аккуратны, когда вы выполняете инверсию для составных условий, поскольку результаты не всегда очевидны. В последнем примере листинга 2 будут возвращены записи с атрибутом **OU**, равным **Engineering**, если их обычное имя не начинается с **Sean**. Чтобы запись была исключена из результатов поиска, должны выполняться оба условия.

## Диапазоны поиска

Часто возникает необходимость выполнить поиск диапазона значений. Для поиска атрибутов в LDAP существуют операторы **<=** и **>=**. Обратите внимание на то, что знак равенства (=) входит в состав операторов, поскольку операторов < и > не существует – вы должны также проверять условие равенства.

Не все целочисленные атрибуты можно проверить с помощью операторов диапазонов. Если вы сомневаетесь, проверьте схему, чтобы убедиться в том, что для атрибута реализован тип сортировки через ключевое слово **ORDERING**. Вспомните материал руководства [Подготовка к экзамену LPI 301: понятия, архитектура и модель](#), в котором говорилось о том, что атрибут определяется в схеме, и что частью этого определения является способ сортировки атрибута сервером.

## Поиск созвучий

Каталог LDAP часто используется для хранения имен, которые могут произноситься одинаково, несмотря на различное написание. Можно, например, спутать имена "Sean", "Shawn" и "Shaun". В LDAP существует оператор "**~**", возвращающий результаты, которые произносятся так же, как и строка поиска. Например, (**cn~=Shaun**) возвращает результаты,

обычное имя (CN) которых содержит слово, произносящееся так же, как "Shaun". Так, результатом поиска по подстроке (`cn=~Shaun`) будет являться элемент `cn=Shawn Walberg`. В то же время реализация OpenLDAP не совершенна; этот же запрос не возвратит результатов для подстроки с именем "Sean".

### Поиск отличительного имени

До сих пор все рассмотренные примеры были нацелены на поиск атрибутов, но не на поиск отличительного имени (DN), которое идентифицирует запись. Хотя крайний левый компонент отличительного имени, относительное отличительное имя (Relative DN, RDN), появляется в качестве атрибута и, следовательно, может быть объектом поиска, фильтры поиска, рассмотренные до сих пор, не выполняют поиск в оставшейся части отличительного имени.

Поиск отличительного имени выполняется при помощи специального фильтра запроса, требующего точного совпадения и имеющего следующий формат:

**атрибут :dn:=значение**, где атрибут – это компонент отличительного имени, которое вы хотите найти, а значение – строка поиска (использование подстановочных символов не разрешено). Например, запрос (`ou:dn:=People`) возвратит все записи, содержащие подстроку `ou=People` в отличительном имени, включая сам контейнерный объект.

### Изменение правила matchingRule

По умолчанию большинство строк, таких как обычное имя, нечувствительно к регистру. Если вы хотите изменить это правило, вы можете использовать форму, подобную форме поиска отличительного имени. Например, запросу (`ou:caseexactmatch:=people`) будут соответствовать все организационные подразделения (OU), содержащие строку "people", но не "People". Ниже перечислены некоторые наиболее часто используемые правила сравнения:

- **caseIgnoreMatch** выполняет сравнение строк, игнорируя регистр символов. Также игнорируются начальные и завершающие символы пробела.
- **caseExactMatch** выполняет сравнение строк с учетом регистра символов, который должен в точности совпадать в исходной и искомой строках.
- **octetStringMatch** похож на поиск строки, но знаки пробела не удаляются, то есть выполняется точное побайтовое сравнение.
- **telephoneNumberMatch** выполняет поиск телефонного номера, который имеет свой собственный тип данных в LDAP.

Также вы можете изменить правило сравнения для поиска DN, комбинируя поиск DN с поиском на основе правил сравнения. Например, (`ou:dn:caseexactmatch:=people`) выполняет поиск имени DN, содержащего точную строку "people".

Поиск отличительного имени и поиск на основе правил сравнения также называются *расширенным поиском*. В обоих случаях необходимо указывать точные значения строк без использования подстановочных символов.

### Использование `ldapsearch`

Утилита командной строки `ldapsearch` предназначена для поиска по дереву каталога. Она позволяет выполнять привязку к каталогу различными способами, выполнять один или несколько поисков и получать данные в LDIF-формате.

По умолчанию `ldapsearch` работает следующим образом:

- Пытается выполнить аутентификацию на сервере с использованием SASL (Simple Authentication and Security Layer)
- Подключается к серверу по адресу `ldap://localhost:389`
- Использует в качестве фильтра поиска конструкцию (`objectClass=*`)

- Считывает базу поиска из файла /etc/openldap/ldap.conf
- Выполняет поиск по базе поиска и всем ее дочерним областям
- Возвращает все пользовательские атрибуты, игнорируя рабочие (для внутреннего использования) атрибуты
- Использует для вывода результатов расширенный LDIF-формат (LDAP Data Interchange Format)
- Выводит результаты поиска без сортировки

## Аутентификация на сервере

Если вы не используете SASL, вам необходимо выполнить простую аутентификацию, используя параметр -x. Параметр -X выполняет *анонимную привязку*, при которой не используется отличительное имя или пароль. При остальных параметрах по умолчанию **ldapsearch -x** выведет полное дерево каталога, начиная с базы поиска, указанной в файле /etc/openldap/ldap.conf. В листинге 3 показан пример использования простого анонимного поиска.

### Листинг 3. Простой анонимный поиск

```
$ ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# people, ertw.com
dn: ou=people,dc=ertw,dc=com
ou: people
description: All people in organization
objectClass: organizationalUnit
... дальнейшее содержимое не показано ...
```

В листинге показаны заголовок и первая запись, возвращенная в результате выполнения простого анонимного поиска. Первые семь строк формируют заголовок и в соответствии с принятыми в LDIF правилами закомментированы путем добавления в начало строки символа решетки (#). Первые три строки определяют весь последующий текст в качестве данных в расширенном LDIF-формате, возвращенных с использованием LDAP версии 3. Следующая строка показывает, что отличительное имя базового объекта не было указано, и что используется поиск в подразделах дерева (subtree search). Последние две текстовые строки задают фильтр поиска, запрашивающий любые данные, а также указывают на то, что были запрошены все атрибуты.

Если вы хотите выводить результаты, не содержащие комментариев, вы можете использовать параметр -LLL.

После заголовка перечисляются все записи. Каждая запись начинается с заголовка, описывающего ее, а затем идет список всех атрибутов, начиная с DN. Атрибуты не сортируются

Если для подключения вам необходимо указать имя пользователя и пароль, используйте для этого параметры -D и -W соответственно. Например, команда **ldapsearch -x D cn=root,dc=ertw,dc=com -w мой\_пароль** выполнит простую аутентификацию с

использованием отличительного имени пароля пользователя root. Если вы хотите, чтобы при вводе пароль не отображался на экране, используйте для этого параметр `-W` вместо `-W пароль`.

Вы также можете подключиться к другому серверу, передав унифицированный идентификатор ресурса (Uniform Resource Identifier, URI) удаленного сервера LDAP с помощью параметра `-H`, например, `ldapsearch -x -H ldap://192.168.1.1/` – чтобы подключиться к LDAP-серверу по адресу 192.168.1.1.

## Выполнение запросов

Для выполнения запроса укажите в командной строке фильтр поиска. Вероятнее всего, вам придется заключить фильтр в кавычки, чтобы специальные символы в строке поиска не были восприняты системой как управляющие символы командной оболочки. В листинге 4 приведен пример простого поиска обычного имени.

### Листинг 4. Простой поиск из командной строки

```
$ ldapsearch -LLL -x '(cn=Fred Smith)'  
dn: cn=Fred Smith,ou=people,dc=ertw,dc=com  
objectClass: inetOrgPerson  
sn: Smith  
cn: Fred Smith  
mail: fred@example.com
```

Поиск в листинге 4 был выполнен с параметром `-LLL` для пропуска комментариев и параметром `-X` для использования простой аутентификации. Последний параметр – это строка поиска, которая ищет запись пользователя Fred Smith. Обратите внимание, что запрос заключен в круглые скобки, а также в одиночные кавычки. Эти кавычки необходимы, чтобы скобки не были восприняты системой как обращение к подоболочке, а подстрока "Smith" не была интерпретирована как отдельный аргумент (поскольку строка запроса содержит знак пробела).

В листинге 4 были возвращены все атрибуты записи Fred Smith. Если же вам нужны всего один или два атрибута, то поиск значений всех атрибутов записи будет пустой тротай ресурсов как клиента, так и сервера. Чтобы выполнить поиск только нужных вам атрибутов, добавьте их имена в конец командной строки `ldapsearch`. В листинге 5 показано, как будет выглядеть предыдущий запрос в случае, если вы просто хотите узнать адрес электронной почты пользователя Fred Smith.

### Листинг 5. Запрос адреса электронной почты пользователя Fred Smith

```
$ ldapsearch -LLL -x '(cn=Fred Smith)' mail  
dn: cn=Fred Smith,ou=people,dc=ertw,dc=com  
mail: fred@example.com
```

В листинге 4 в конец командной строки добавлен атрибут `mail`, в результате чего было найдено отличительное имя и его запрошенные атрибуты.

Чтобы определить базу поиска, `ldapsearch` ищет в файле `/etc/openldap/ldap.conf` строку, начинающуюся с `BASE`, и в случае, если такая строка не обнаружена, использует параметр сервера `defaultsearchbase`. База поиска – это точка дерева каталога, с которой начинается поиск. Поиск будет выполняться только в дочерних элементах базы поиска

(включая ее саму). Чтобы указать другую базу поиска, используйте параметр **-b**, например, `ldapsearch -x -b ou=groups,dc=ertw,dc=com` – для поиска в контейнере `groups` дерева `ertw.com`.

### Управление выводом результатов

LDAP может хранить двоичные данные, например, изображения. Стандартным способом хранения изображений в дереве каталога является использование атрибута `jpegPhoto`. Если вы получаете значение атрибута из командной строки, оно возвращается вам в кодировке `base64`. Для сохранения любых двоичных атрибутов во временном файле используется параметр **-t**. В листинге 6 показано, как использовать этот параметр.

### Листинг 6. Сохранение двоичных атрибутов в файловой системе

```
$ ldapsearch -LLL -x 'cn=joe*' jpegphoto | head
dn: cn=Joe Blow,ou=people,dc=ertw,dc=com
jpegPhoto:: /9j/4AAQSkZJRgABAQEASABIAAD/
/gAXQ3JlYXR1ZCB3aXRoIFRoZSBHSU1Q/9sAQw
... далее следуют еще более 1300 строк ...

$ ldapsearch -LLL -t -x '(cn=joe*)' jpegphoto
dn: cn=Joe Blow,ou=people,dc=ertw,dc=com
jpegPhoto:< file:///tmp/ldapsearch-jpegPhoto-VaIjkE

$ file /tmp/ldapsearch-jpegPhoto-VaIjkE
/tmp/ldapsearch-jpegPhoto-VaIjkE:
JPEG image data, JFIF standard 1.01, comment: \
"Created with The GIMP\377"
```

В листинге 6 показаны два запроса, выполняющих поиск всех людей с именем, начинающимся на "Joe", и возвращающих только атрибут `jpegPhoto`. В первом запросе параметр **-t** не используется, поэтому значение атрибута `jpegPhoto` выводится в консоль в формате `base64`. При использовании командной строки такой запрос бесполезен, поэтому во втором запросе указан параметр **-t**. На этот раз значение атрибута `jpegPhoto` является URI-ссылкой на файл (вы можете изменить директорию с помощью параметра **-T**). Наконец, полученный файл проверяется, и он действительно является бинарной версией изображения, которое теперь можно просмотреть.

По умолчанию `ldapsearch` выводит результаты в том порядке, в котором они были получены от сервера. Вы можете выполнить сортировку с помощью параметра **-S**, указав имя атрибута, значения которого вы хотите отсортировать. Чтобы выполнить сортировку по нескольким атрибутам, разделяйте их имена запятыми (,).

## Утилиты командной строки LDAP

В этом разделе описывается материал по теме 304.2 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 4.

Из этого раздела вы узнаете, как:

- Использовать утилиты `ldap*` для доступа и изменения каталога
- Использовать утилиты `slap*` для доступа и изменения каталога

В OpenLDAP имеется несколько утилит для управления каталогом и администрирования сервера. Вы уже знакомы с утилитой `ldapsearch`, которая была рассмотрена в [предыдущем](#)

[разделе](#). Команды, начинающиеся с `ldap`, предназначены для пользователей дерева каталога, а команды, начинающиеся с `slap` – для администраторов.

## Утилиты управления каталогом

Команды, которые будут рассмотрены в этом разделе, предназначены для управления деревом каталога; выполняется либо изменение, либо чтение данных. Утилита `ldapsearch` также попадает в эту категорию. Чтобы использовать эти команды, необходимо пройти аутентификацию на сервере.

### `ldapadd` и `ldapmodify`

Эти две команды используются для добавления и изменения элементов каталога. Вспомните материал руководства [Подготовка к экзамену LPI 301: понятия, архитектура и модель](#), в котором говорилось о том, что для добавления, изменения и удаления данных из дерева каталога может использоваться LDIF-формат. В листинге 7 показан пример использования LDIF для добавления записи.

#### Листинг 7. Использование LDIF-формата для добавления записи в дерево каталога

```
dn: cn=Sean Walberg,ou=people,dc=ertw,dc=com
objectclass: inetOrgPerson
cn: Sean Walberg
cn: Sean A. Walberg
sn: Walberg
homephone: 555-111-2222
```

Листинг 7 начинается с описания отличительного имени записи. Эта запись будет располагаться в контейнере `ou=people,dc=ertw,dc=com` и иметь относительное отличительное имя `cn=Sean Walberg`; оно получается путем разделения отличительного имени (DN) на части после первой пары атрибут/значение. Классом объекта (`objectclass`) записи будет являться `inetOrgPerson` – этот класс является наиболее общим типом для описания любого сотрудника организации. Далее следуют два варианта обычного имени, фамилия и, наконец, номер домашнего телефона.

В листинге 7 показан пример добавления записи, в отличие от изменения или удаления. Вспомните, что в LDIF-файлах может быть указано ключевое слово `changetype`, говорящее о том, что нужно делать с данными.

Команда `ldapadd` используется для обработки LDIF-файла. Если содержимое листинга 7 сохранить в файле под именем "sean.ldif", команда `ldapadd -x -D cn=root,dc=ertw,dc=com -w mypass -f sean.ldif` будет являться одним из способов добавления новой записи в дерево каталога. Часть команды `-x -D cn=root,dc=ertw,dc=com -w mypass` должна быть вам знакома из предыдущего материала об утилите `ldapsearch` – она является способом аутентификации на сервере каталога с использованием простой аутентификации и всемогущей учетной записи root. Параметры аутентификации на сервере каталога одинаковы для всех команд `ldap`, рассматриваемых в этом разделе, поэтому вы будете часто встречать эти параметры.

Команда `ldapadd` реализована в виде символьной ссылки к `ldapmodify`, и когда происходит вызов `ldapadd`, эта команда интерпретируется как `ldapmodify -a`. Параметр `-a` указывает команде `ldapmodify` использовать в качестве значения `changetype` параметр `add`, предназначенный для добавления новых записей в дерево каталога. При вызове команды `ldapmodify` предполагается, что значением `changetype` по умолчанию является операция изменения (`modify`).

Команда `ldapadd` (а также `ldapmodify`) является эффективным способом загрузки очень большого количества данных на сервер без необходимости его остановки. LDIF-файлы могут содержать множество операций, и часто оказывается проще сгенерировать LDIF-файл из каких-либо источников данных, которые вы пытаетесь импортировать, чем написать собственный код для их разбора и добавления непосредственно через LDAP.

### **ldapdelete**

Команда `ldapdelete`, как и подразумевает ее имя, удаляет запись из дерева каталога. Все записи идентифицируются в каталоге по их уникальным отличительным именам; следовательно, `ldapdelete` удаляет записи на основе их DN, не используя никаких других запросов.

Кроме рассмотренных выше параметров аутентификации, `ldapdelete` может принимать список отличительных имен для удаления, как из командной строки, так и из файла. Для удаления записей из командной строки просто укажите в строке имена DN, например, `ldapdelete -x -D cn=root,dc=ertw,dc=com -w mypass "cn=Sean Walberg,ou=people,dc=ertw,dc=com"`. Если вам необходимо удалить много записей, вы можете поместить их отличительные имена в файл (по одному имени на каждую строку) и указать `ldapdelete` на этот файл с помощью параметра `-f имя_файла`.

Заметьте, что вы также можете удалять записи с помощью LDIF и команд `ldapadd/ldapmodify`. Команда `ldapdelete` является наиболее подходящей для большинства случаев, но это не единственный способ удаления записей.

### **ldapmodrdn**

Команда `ldapmodrdn` изменяет относительное отличительное имя объекта, то есть первую пару атрибут/значение в имени DN. Эта команда переименовывает запись в текущей ветви дерева каталога. В отличие от значения `moddn` LDIF-параметра `changetype`, эта команда может только переименовывать запись, но не может перемещать ее в другое место дерева каталога.

Использовать эту команду достаточно просто: укажите учетные данные для аутентификации, отличительное имя (DN) записи и новое относительное отличительное имя (RDN). В листинге 8 показана процедура переименования учетной записи "Joe Blow" в "Joseph Blow".

#### **Листинг 8. Переименование записи**

```
$ ldapmodrdn -x -D cn=root,dc=ertw,dc=com -w dirtysecret \
  'cn=Joe Blow,ou=people,dc=ertw,dc=com' 'cn=Joseph Blow'
$ ldapsearch -LLL -x '(cn=Joseph Blow)'
dn: cn=Joseph Blow,ou=people,dc=ertw,dc=com
objectClass: inetOrgPerson
sn: Blow
cn: Joe Blow
cn: Joseph Blow
```

Обратите внимание на то, что старое имя RDN все еще является атрибутом записи: `cn: Joe Blow`. Если вы хотите удалить старое имя RDN, добавьте в командной строке параметр `-r`. Добавление этого параметра имеет тот же эффект, что и добавление `deleteoldrdn: 1` в ваш LDIF-код (как ни странно, это является поведением по умолчанию для LDIF, но не для `ldapmodrdn`).

### **ldapcompare**

Команда `ldapcompare` позволяет вам сравнивать предопределенные значения со

значениями, хранящимися где-нибудь в дереве LDAP. В следующем примере показано, как это работает.

### Листинг 9. Использование `ldapcompare`

```
$ ldapcompare -x "cn=Sean Walberg,ou=people,dc=ertw,dc=com" userPassword:мой_пароль
TRUE
$ ldapcompare -x "cn=Sean Walberg,ou=people,dc=ertw,dc=com" userPassword:неверный_пароль
FALSE
```

В листинге 9 выполняется команда `ldapcompare`. Вслед за параметрами аутентификации указываются два параметра, которые представляют собой имя DN для сравнения, а также атрибут и значение, с которыми оно будет сравниваться. В обоих примерах, приведенных в листинге, именем DN для сравнения является "cn=Sean Walberg", а атрибутом, с которым выполняется сравнение – `userPassword`. Когда указан правильный пароль, `ldapcompare` выводит строку `TRUE` и возвращает код 6. Если указанное значение не совпадает с тем, что содержится в записи, то выводится строка `FALSE` и возвращается код 5. Параметр `-Z` предотвращает любой вывод на экран, при этом предполагается, что результаты сравнения будут оценены по возвращаемому коду.

Хотя в примерах листинга 9 проверялся пароль, можно использовать любой атрибут, включая `objectClass`. Если атрибут имеет несколько значений, например, указано несколько обычных имен (CN) или классов `objectClass`, то результат сравнения будет успешным, если произойдет совпадение хотя бы с одним из них.

### `ldapwhoami`

Команда `ldapwhoami` позволяет вам выполнить проверку аутентификации на LDAP-сервере и определять, под каким именем DN вы выполнили аутентификацию. Просто вызовите команду `ldapwhoami` с обычными параметрами аутентификации, как показано в листинге 10.

### Листинг 10. Демонстрация работы `ldapwhoami`

```
$ ldapwhoami -x
anonymous
Result: Success (0)
$ ldapwhoami -x -D "cn=Sean Walberg,
ou=people,dc=ertw,dc=com" -w мой_пароль
dn:cn=Sean Walberg,ou=people,dc=ertw,dc=com
Result: Success (0)
$ ldapwhoami -x -D "cn=Sean
Walberg,ou=people,dc=ertw,dc=com" -w неверный_пароль
ldap_bind: Invalid credentials (49)
```

В первом примере листинга 10 показана привязка без использования имени пользователя или пароля. Команда `Ldapwhoami` возвращает строку `anonymous`, показывающую, что была выполнена анонимная привязка, а также строку состояния, показывающую, что аутентификация прошла успешно. Во втором примере показана привязка с использованием отличительного имени пользователя. На этот раз возвращенное имя DN совпадает с именем, указанным при аутентификации. Наконец, в третьем примере при попытке аутентификации были указаны неверные учетные данные. Возвращенный результат указывает на возникшую проблему.

**ldapwhoami** может оказаться полезной для диагностики конфигурации сервера, а также для ручной проверки паролей. Списки доступа могут помешать выполнению команды **ldapsearch**, так что использование вместо этого команды **ldapwhoami** может помочь вам определить, заключается ли проблема в учетных данных или в списках доступа.

## Утилиты администрирования

Команды, начинающиеся с **slap**, предназначены для администраторов и оперируют непосредственно файлами базы данных вместо того, чтобы работать через протокол LDAP. Таким образом, для работы с этими командами вам, как правило, необходимо будет использовать учетную запись пользователя **root**, а в некоторых случаях будет необходимо также останавливать сервер.

### slapacl

**slapacl** – это утилита, которая позволяет администратору проверять списки доступа и устанавливать взаимосвязи между именами DN для привязки, записями и атрибутами. Например, с помощью **slapacl** можно выяснить уровень доступа какого-либо отдельного пользователя к атрибутам другого пользователя. Эта команда должна выполняться из-под учетной записи **root**, поскольку она напрямую обращается к базе данных и файлам конфигурации, не используя LDAP.

Использование утилиты **slapacl** лучше всего показать на примерах. В листинге 11 администратор выполняет проверку с целью выяснить, какой доступ имеет пользователь к своему собственному паролю до и после внедрения списков доступа (ACL), призванных ограничить доступ к наиболее конфиденциальным данным.

### Листинг 11. Использование **slapacl** для проверки эффекта от внедрения ACL

```
# slapacl -D "cn=Sean Walberg,ou=people,dc=ertw,dc=com" \
    -b "cn=Sean Walberg,ou=People,dc=ertw,dc=com" userPassword
authcDN: "cn=sean walberg,ou=people,dc=ertw,dc=com"
userPassword: read(=rscxd)

... изменение slapd.conf ...

# slapacl -D "cn=Sean Walberg,ou=people,dc=ertw,dc=com" \
    -b "cn=Sean Walberg,ou=People,dc=ertw,dc=com" userPassword
authcDN: "cn=sean walberg,ou=people,dc=ertw,dc=com"
userPassword: =wx

# slapacl -D "cn=Joseph Blow,ou=people,dc=ertw,dc=com" \
    -b "cn=Sean Walberg,ou=People,dc=ertw,dc=com" userPassword
authcDN: "cn=joseph blow,ou=people,dc=ertw,dc=com"
userPassword: =0
```

Каждый раз команде **slapacl** должны передаваться два обязательных набора данных. Первая часть – это имя DN для привязки, которое является именем DN пользователя, чьи права доступа проверяются. Вторая часть – это имя DN записи, с которой будет выполняться сравнение. Имя для привязки указывается с помощью параметра **-D**, а целевое имя записи – с помощью параметра **-b**. При желании вы можете ограничить проверку одним атрибутом, включив его в конец строки (как, например, атрибут **userPassword** в листинге 11). Если вы не указываете атрибут, вы получите результаты для каждого атрибута записи.

В первом примере листинга 11 администратор проверяет элемент **cn=Sean Walberg** с целью выяснить, какой доступ имеет этот пользователь к своему собственному паролю.

Результат – доступ на чтение. Вспомните материал руководства [Подготовка к экзамену LPI 301: конфигурирование](#), в котором говорилось о том, что пользователям должно быть разрешено изменять и использовать свой атрибут `userPassword` для аутентификации, но не просматривать его. После изменения списков контроля доступа (ACL) проверка выполняется снова, и теперь пользователь имеет разрешение только на запись и на выполнение аутентификации. Наконец, в третьем примере выполняется проверка с целью выяснить, какой доступ имеет пользователь Joseph Blow к паролю пользователя Sean Walberg. Результат – нет доступа.

Утилита `slapacl` является эффективным способом проверки изменений списков доступа и устранения сопутствующих проблем. Эта утилита чрезвычайно эффективна, поскольку она считывает данные непосредственно из базы данных и из файла `slapd.conf`; таким образом, все изменения, сделанные в файле `slapd.conf`, отражаются в выводе `slapacl`, и перезагрузка демона `slapd` не требуется.

### **slapcat**

Утилита `slapcat` выгружает содержимое каталога LDAP в виде LDIF в устройство стандартного вывода или файл (если вы используете параметр `-l имя_файла`). При желании вы можете использовать параметр `-S`, чтобы указать начальное имя DN, или параметр `-a`, чтобы задать фильтр запроса.

Утилита `slapcat` работает непосредственно с базой данных и не может быть запущена, пока сервер работает. Поддерживаются только базы данных типа `bdb`.

### **slapadd**

Утилита `slapadd` является инструментом импорта большого объема данных и работает непосредственно с серверными СУБД. Это означает, что для использования этой утилиты демон `slapd` должен быть остановлен. Этот инструмент разработан для использования с результатами вывода утилиты `slapcat`. `slapadd` не выполняет подробной проверки входных данных, поэтому возможно, что результатом ее работы будут разрозненные ветви дерева каталога. Это может случиться в том случае, если какие-либо контейнерные объекты не будут импортированы.

Входными данными утилиты `slapadd` является LDIF-файл, например, сгенерированный командой `slapcat`. В справочном man-руководстве `slapadd(8C)` вместо этой утилиты предлагается использовать `ldapadd`, поскольку в этом варианте предусмотрена проверка данных. Также в man-руководстве упоминается, что нет гарантий того, что вывод `slapcat` будет упорядочен в соответствии с требованиями совместимости `ldapadd` (контейнерные объекты могут следовать за дочерними, следовательно, проверка может закончиться с ошибками). Использование в `slapcat` фильтров также может привести к потере важных данных. Таким образом, следует использовать утилиту `slapadd` только с LDIF-файлами, сгенерированными командой `slapcat`, и использовать `ldapadd` для любых других LDIF-форматов.

После того как вы остановите ваш LDAP-сервер, вы можете просто запустить команду `slapadd` и направить ее вывод на устройство стандартного ввода с помощью `pipe-канала`. Если вы хотите считывать данные из файла, используйте параметр `-l`. Как и в случае с утилитой `slapcat`, поддерживаются только базы данных типа `bdb`.

### **slappasswd**

Утилита `slappasswd` используется для создания хэшированных паролей, которые будут храниться в каталоге или в файле `slapd.conf`. Наиболее распространенный случай применения утилиты – создание хэшированных паролей административных учетных записей `rootDN`, хранящихся в файле `slapd.conf`; в этом случае, даже имея доступ к файлу конфигурации, невозможно узнать пароль администратора. Если вы не укажете никаких параметров в

командной строке, **slappasswd** предложит вам ввести пароль для хэширования, как показано в листинге 12.

### Листинг 12. Использование **slappasswd** для хэширования пароля

```
$ slappasswd
New password:
Re-enter new password:
{SSHA}G8Ly2+t/HMHJ30WWE7LN+GRmZJAweXoE
```

После этого вы можете скопировать получившийся результат в строку **rootpw** файла **slapd.conf**. **slapd** распознает формат пароля и поймет, что **{SSHA}** указывает на то, что последующие данные представляют собой хэш SHA1. Человек, прочитавший файл **slapd.conf**, не сможет узнать пароль учетной записи **root**.

Хэши, сгенерированные утилитой **slappasswd**, могут быть также использованы и в LDIF-файлах; утилиты **ldapadd** и **ldapmodify** позволяют хранить защищенные хэши ваших паролей вместо того, чтобы хранить пароли в открытом виде или в кодировке base64.

### **slapindex**

Вы, возможно, помните об утилите **slapindex** из материала руководства [Подготовка к экзамену LPI 301: конфигурирование](#). После создания или изменения индекса при помощи ключевого слова **index** в файле **slapd.conf** необходимо перестроить индексы, иначе **slapd** возвратит неправильные результаты. Чтобы перестроить индексы, остановите **slapd** и запустите **slapindex**. В зависимости от размера вашей базы данных это может занять некоторое время или, как говорится в **man**-руководстве, "Эта команда предоставляет пользователю прекрасную возможность найти и выпить свой любимый напиток".

### **slaptest**

Утилита **slapdtest** просто выполняет проверку вашего файла **slapd.conf** на ошибки. Это полезная утилита, поскольку если вы будете перезапускать **slapd**, а файл конфигурации окажется некорректным, вы не сможете запустить сервер, пока не исправите все ошибки. **slapdtest** позволяет вам выполнить проверку на отсутствие ошибок в вашем конфигурационном файле прежде, чем вы начнете перезапускать сервер.

Для использования **slaptest** достаточно просто набрать **slaptest** в командной строке. Если файл **slapd.conf** не содержит ошибок, вы увидите сообщение **config file testing succeeded**. В противном случае вы получите ошибку с описанием возникшей проблемы.

**slaptest** также проверяет существование различных файлов и директорий, необходимых для работы. Тем не менее, во время проверки автор смог найти несколько ошибок в конфигурационных файлах, которые не обнаружила утилита **slaptest** и которые при этом могли бы привести к ошибкам в работе **slapd**.

### Служба "белых страниц"

В этом разделе описывается материал по теме 304.3 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 1.

Из этого раздела вы узнаете, как:

- планировать службу "белых страниц"
- настраивать службу "белых страниц"

- настраивать клиентские компьютеры на получение данных от службы "белых страниц".

Служба "белых страниц" позволяет клиентам электронной почты получать контактную информацию из базы данных LDAP. Используя такие основные атрибуты, как, например, атрибуты класса `inetOrgPerson objectClass`, вы можете обеспечить наилучшую совместимость с почтовыми клиентами. Например, как Microsoft Outlook, так и Evolution используют атрибут `mail` для хранения электронного почтового адреса пользователя, а атрибуты `givenName`, `displayName`, `cn` и `sn` – для хранения имени пользователя в различных формах.

### **Настройка клиентов электронной почты на использование каталога LDAP**

Теоретически, любой почтовый клиент, поддерживающий LDAP, может использовать ваше дерево каталога. Для настройки почтового клиента вам потребуется следующая информация:

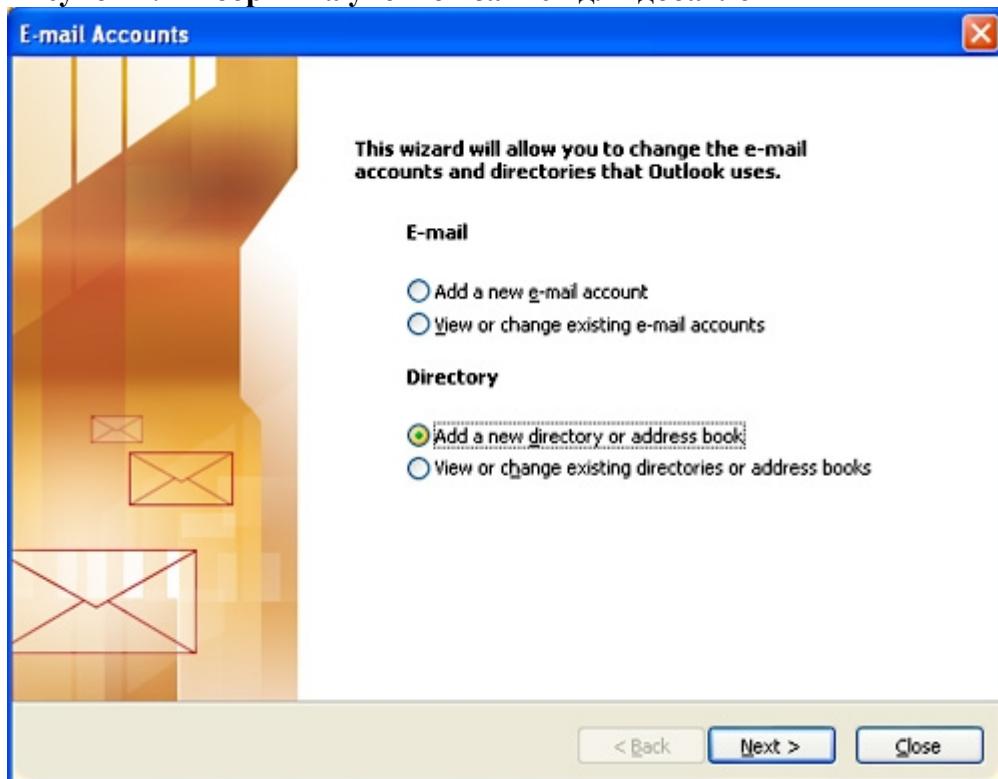
- Адрес или имя хоста сервера LDAP
- Учетные данные для привязки, если вы не выполняете анонимную привязку
- Отличительное имя базового объекта (Base DN), с которого необходимо начинать поиск
- Фильтр поиска, например, (`mail=*`), чтобы выбирать только учетные записи, содержащие адрес электронной почты (необязательно)

После того как вы укажете эту информацию в настройках почтового клиента, у вас должна появиться возможность выполнять поиск контактов.

### **Настройка Microsoft Outlook на использование каталога LDAP**

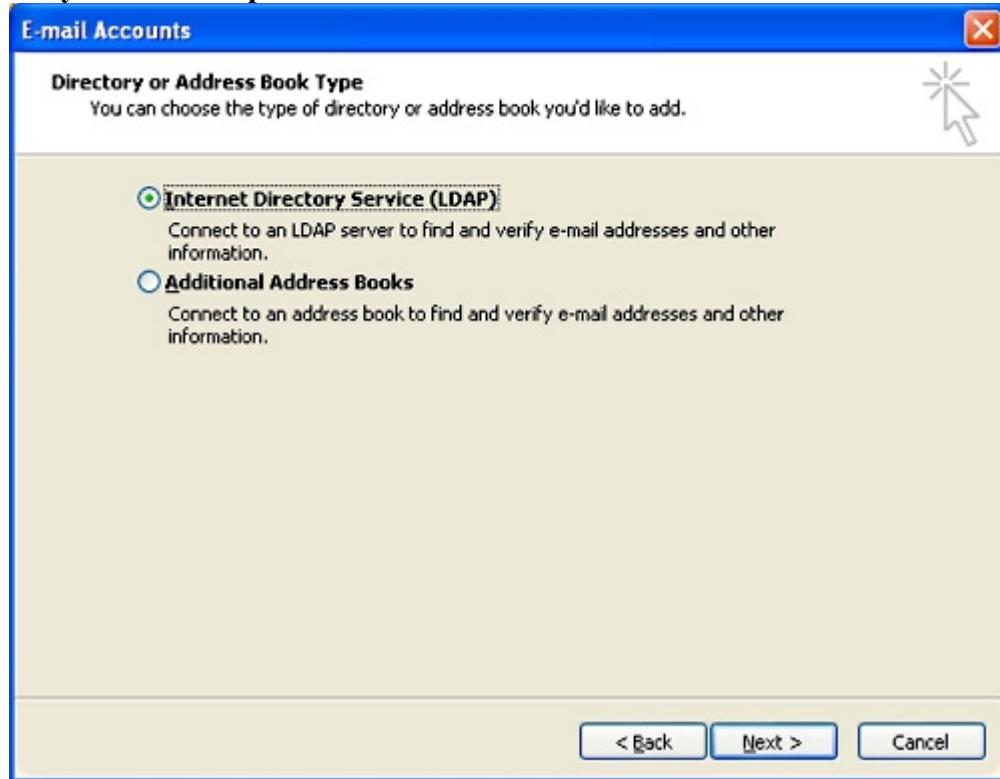
Для настройки Microsoft Outlook (проверено в Outlook 2003) выберите пункт меню **Tools > Email Accounts**. Вы увидите диалоговое окно, изображенное на рисунке 2.

**Рисунок 2. Выбор типа учетной записи для добавления**



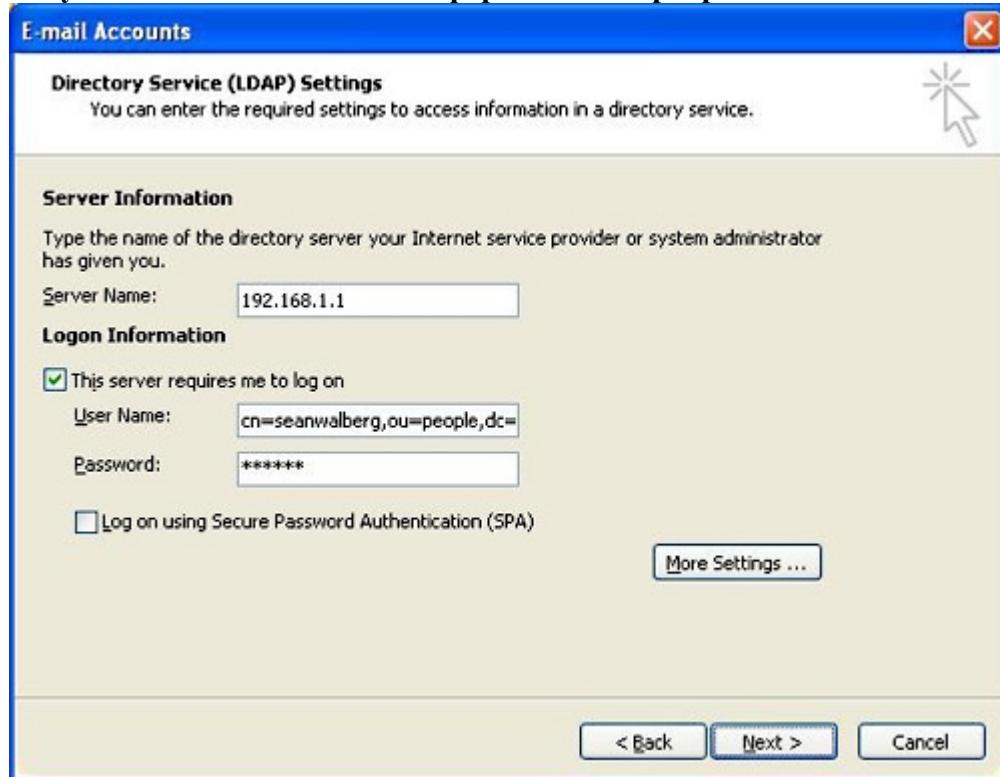
Выберите добавление нового каталога и нажмите кнопку **Next**. Вы увидите диалоговое окно, изображенное на рисунке 3.

**Рисунок 3. Выбор типа каталога для добавления**



Выберите добавление нового каталога LDAP и нажмите кнопку **Next**. Вы увидите диалоговое окно, изображенное на рисунке 4.

**Рисунок 4. Ввод детальной информации о сервере LDAP**



Введите соответствующие данные о вашем сервере LDAP в диалоговом окне, показанном на рисунке 4. В показанном примере для привязки к дереву каталога используются учетные

данные пользователя. Вы можете использовать анонимный доступ, если конфигурация вашего сервера позволяет сделать это.

После ввода основной информации нажмите кнопку **More Settings**, и вам будет предложено ввести дополнительную информацию, как показано на рисунке 5.

**Рисунок 5. Ввод дополнительной информации о конфигурации сервера LDAP**



На рисунке 5 показаны дополнительные параметры, наиболее важным из которых является база поиска (Search base). После указания базы поиска нажмите кнопку **OK**, чтобы возвратиться в главное окно Outlook.

Теперь вы можете использовать базу данных LDAP всякий раз, когда вам предлагается выполнить поиск пользователей, указав имя сервера в поле "Show Names From".

## Заключение

Из этого руководства вы узнали, как выполнять поиск по вашему каталогу и использовать утилиты командной строки. Также вы узнали, как настраивать почтовые клиенты на использование дерева каталога для хранения контактной информации.

Для выполнения поиска по дереву каталога LDAP необходимо построить фильтр запроса. Различные операторы, использующиеся для построения поисковых запросов, перечислены в таблице 3.

**Таблица 3. Операторы поиска в LDAP**

Оператор	Описание	Пример запроса
=	Проверка равенства	(cn=Walberg)
*	Проверка существования атрибута	(cn=*)
	Поиск подстроки	(sn=Walb*)
&	Логическое "И"	(&(condition1)(condition2))
	Логическое "ИЛИ"	( (condition1)(condition2))
!	Логическое "НЕ"	(!(mail=*))
~=	Возвращение созвучных результатов	(cn~=Shawn)

<= и >= Поиск в диапазоне

(pagesPerMinute >= 20)

Некоторые утилиты предназначены для повседневного использования каталога, например, `ldapsearch` – для поиска, а `ldapadd` и `ldapmodify` – для добавления и изменения данных. Утилиты, имена которых начинаются с `ldap`, работают через LDAP-протокол и требуют использования учетных записей для подключения к серверу. Утилиты, имена которых начинаются с `slap`, предназначены для администраторов и работают непосредственно с базой данных.

Это руководство, а также три предыдущих руководства из [серии 301](#), содержат информацию о том, как управлять и работать с сервером LDAP. В следующем руководстве этой серии будут рассмотрены различные приложения, включая серверы электронной почты, а также то, как использовать LDAP в качестве источника данных.

## Ресурсы

### Научиться

- Оригинал руководства "[LPI exam 301 prep, Topic 304: Usage](#)" (EN).
- Изучите предыдущее руководство в серии 301 - "[Подготовка к экзамену LPI 301, Тема 303: конфигурирование](#)" (developerWorks, март 2008) или [все руководства в серии 301](#) (EN).
- Чтобы познакомиться с основами Linux и подготовиться к сертификации в качестве системного администратора, ознакомьтесь со всей [серий руководств для подготовки к экзаменам LPI](#).
- В [программе LPIC](#) (EN) вы можете найти перечни заданий, примеры вопросов и подробные цели для трех уровней сертификации системных администраторов Linux института Linux Professional Institute.
- В спецификации [RFC 4515 – Строковое представление фильтров поиска LDAP](#) (EN) вы можете найти дополнительную информацию о фильтрах поиска.
- Прочтайте статьи "[MS Outlook: What LDAP Attributes Are Recognised?](#)" (EN) и "[MS Outlook: How Do LDAP Attributes Map to Address Book Fields?](#)" (EN), чтобы увидеть результаты разбора структуры параметров атрибутов, которые использует клиент Microsoft Outlook.
- Очень советую онлайновую книгу [LDAP для больших ученых](#) (EN), несмотря на то, что работа над ней ещё не закончена.
- Узнайте больше о [правилах сравнения](#) (EN).
- В [разделе Linux сайта developerWorks](#) можно найти дополнительные ресурсы для разработчиков Linux, а также [самые популярные среди наших читателей статьи и руководства](#).
- Посмотрите все [советы по Linux](#) и [руководства Linux](#) (EN) на сайте developerWorks.
- Следите на последними новостями на портале [Web-трансляций и технических мероприятий developerWorks](#) (EN).

### Получить продукты и технологии

- Утилита [Firewall Builder](#) упрощает задачу создания правил iptables, предоставляя в ваше распоряжение графический интерфейс и набор инструментов для развертывания обновлений на ваших брандмауэрах.

- Загрузите [OpenLDAP](#).
- [phpLDAPadmin](#) - инструмент администрирования LDAP на базе Web. Если вам больше нравится графический интерфейс, вам стоит посмотреть на [Luma](#) (EN).
- Используйте в своем следующем проекте разработки для Linux [ознакомительные версии программного обеспечения IBM](#), которые можно скачать непосредственно с developerWorks (EN).

# Подготовка к экзамену LPI 301: Тема 305. Интеграция и миграция

*Профессионал Linux высокого уровня (LPIC-3)*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux® Professional Institute на квалификацию профессионала Linux высокого уровня (LPIC-3). В этом руководстве, пятом из [серии из шести руководств](#), Шон расскажет об интеграции LDAP с вашими системными учетными данными и приложениями. Также он подробно расскажет о процедуре интеграции вашего сервера в среду Microsoft® Active Directory.

[Больше статей из этой серии](#)

**Дата:** 22.01.2009

**Уровень сложности:** средний

## Предисловие

Узнайте, чему могут научить вас эти руководства, и как получить от них больше пользы.

## Об этой серии руководств

Институт [Linux Professional Institute](#) (LPI) сертифицирует системных администраторов Linux® по трём уровням: *младший уровень* (также называемый "уровень сертификации 1"), *углубленный уровень* (также называемый "уровень сертификации 2") и *высший уровень* (также называемый "уровень сертификации 3"). Чтобы получить сертификацию на уровне 1, нужно сдать экзамены 101 и 102. Чтобы получить сертификацию на уровне 2, нужно сдать экзамены 201 и 202. Чтобы получить сертификацию на уровне 3, у вас должна быть действующая сертификация на углубленном уровне и сдан экзамен 301 ("основной"). Кроме того, на высоком уровне от вас может потребоваться сдача дополнительных экзаменов.

Сайт developerWorks предлагает руководства, которые помогут вам подготовиться к пяти экзаменам для младшего, углубленного и высокого уровня. В каждом экзамене охватывается несколько тем, и для каждой темы на developerWorks есть соответствующий учебник для самостоятельного изучения. В таблице 1 перечислены шесть тем и соответствующие им руководства developerWorks для экзамена LPI 301.

**Таблица 1. Экзамен LPI 301: руководства и темы**

Тема экзамена	Руководство developerWorks	Краткое описание руководства
Тема 301	<a href="#">Подготовка к экзамену LPI 301: понятия, архитектура и модель</a>	Узнайте о понятиях и архитектуре LDAP, о том, как проектировать и внедрять каталог LDAP, а также о схемах.
Тема 302	<a href="#">Подготовка к экзамену LPI 301: установка и разработка</a>	Узнайте, как устанавливать, настраивать и использовать программное обеспечение OpenLDAP.

Тема 303	<a href="#">Подготовка к экзамену LPI 301: конфигурирование</a>	Узнайте более подробно о том, как настраивать программное обеспечение OpenLDAP.
Тема 304	<a href="#">Подготовка к экзамену LPI 301: использование</a>	Узнайте, как следует выполнять поиск по дереву каталога LDAP и использовать утилиты OpenLDAP.
Тема 305	Подготовка к экзамену LPI 301: интеграция и миграция	(Это руководство) Узнайте, как использовать LDAP в качестве источника данных для ваших системных приложений. См. подробные <a href="#">цели</a> .
Тема 306	Подготовка к экзамену LPI 301: планирование пропускной способности	Появится в ближайшее время.

Чтобы сдать экзамен 301 (и получить сертификацию третьего уровня), вы должны:

- обладать несколькими годами опыта установки и поддержки Linux на большом числе компьютеров, используемых в различных целях
- обладать опытом интеграции с различными технологиями и операционными системами
- обладать профессиональным опытом или пройти профессиональную подготовку специалиста Linux корпоративного уровня (включая опыт, полученный при работе в другой роли)
- знать администрирование Linux на углубленном и высоком уровне, включая установку, управление, обеспечение безопасности, решение возникающих проблем и техническое обслуживание
- уметь использовать инструменты с открытым исходным кодом для проведения измерений, необходимых для планирования пропускной способности и решения проблем с ресурсами
- иметь профессиональный опыт применения LDAP для интеграции с сервисами UNIX® и Microsoft® Windows®, в том числе Samba, Pluggable Authentication Modules (PAM), электронной почтой и Active Directory
- уметь планировать, проектировать, разрабатывать, строить и реализовывать полную среду с использованием Samba и LDAP, а также проводить измерения для планирования производительности и оценки безопасности служб
- уметь создавать сценарии на Bash или Perl или знать как минимум один язык системного программирования (например, C)

Для дальнейшей подготовки к сертификации уровня 3, ознакомьтесь с [серий руководств для подготовки к экзамену 301 Института LPI](#) (EN), а также со всей [серий руководств developerWorks для подготовки к экзаменам LPI](#) .

Институт Linux Professional Institute не дает рекомендаций по каким-либо конкретным материалам и методикам для подготовки к экзаменам, разработанным сторонними лицами.

## Об этом руководстве

Добро пожаловать в пятое [из шести руководств](#), призванных помочь вам подготовиться к сдаче экзамена LPI 301, - "Интеграция и миграция". Из этого руководства вы узнаете об интеграции LDAP с системами аутентификации и другими службами UNIX.

Это руководство организовано в соответствии с целями LPI по этой теме. Условно говоря, чем выше вес цели, тем больше вопросов по этой теме будет на экзамене.

## Цели

В таблице 2 подробно перечислены цели этого руководства.

**Таблица 2. Интеграция и миграция: цели экзамена, описанные в этом руководстве**

Цель экзамена LPI	Вес цели	Краткое описание цели
305.1 <a href="#"><u>Интеграция LDAP и PAM/NSS</u></a>	2	Осуществите интеграцию LDAP и основных систем аутентификации
305.2 <a href="#"><u>Миграция с NIS на LDAP</u></a>	1	Разработайте и реализуйте стратегию миграции NIS, включая развертывание шлюза с NIS на LDAP
305.3 <a href="#"><u>Интеграция LDAP и служб UNIX</u></a>	1	Используйте ваш LDAP-сервер в качестве источника данных для SSH, FTP, HTTP и других служб.
305.4 <a href="#"><u>Интеграция LDAP и Samba</u></a>	1	Используйте ваш LDAP-сервер в качестве источника данных для Samba.
305.5 <a href="#"><u>Интеграция LDAP и Active Directory</u></a>	2	Используйте ваш LDAP-сервер вместе со службой каталога Active Directory.
305.6 <a href="#"><u>Интеграция LDAP и служб электронной почты</u></a>	1	Осуществите интеграцию каталога LDAP и ваших почтовых служб.

[В начало](#)

## Необходимые условия

Чтобы извлечь максимум пользы из этого руководства, вы должны обладать глубокими знаниями Linux и иметь работающую Linux-систему, на которой вы сможете практиковаться в выполнении рассматриваемых задач.

Если ваши базовые знания Linux немного устарели, вы можете сначала ознакомиться с [руководствами для экзаменов LPIC-1 и LPIC-2](#).

Различные версии программ могут выводить данные в различных форматах, поэтому результаты, полученные вами, могут отличаться от листингов и рисунков, приведенных в этом руководстве.

## Требования к системе

Чтобы выполнить примеры, приведенные в этом руководстве, вам потребуется рабочая станция под управлением Linux с пакетом OpenLDAP и поддержкой PAM. Большинство современных дистрибутивов удовлетворяют этим требованиям.

# Подготовка к экзамену LPI 301: Тема 305. Интеграция и миграция

*Профессионал Linux высокого уровня (LPIC-3)*

[Шон Уолберг](#), старший сетевой инженер, P.Eng

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux® Professional Institute на квалификацию професионала Linux высокого уровня (LPIC-3). В этом руководстве, пятом из [серии из шести руководств](#), Шон расскажет об интеграции LDAP с вашими системными учетными данными и приложениями. Также он подробно расскажет о процедуре интеграции вашего сервера в среду Microsoft® Active Directory.

## Интеграция LDAP и PAM/NSS

В этом разделе описывается материал по теме 305.1 экзамена на професионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 2.

Из этого раздела вы узнаете, как:

- Настроить NSS на получение информации из каталога LDAP
- Настроить PAM на использование LDAP для аутентификации
- Настроить модули PAM в различных средах UNIX

В традиционном для UNIX стиле средства PAM и NSS (Name Service Switch) абстрагируют многочисленные компоненты систем аутентификации и поиска от их реализации, что позволяет администратору менять конечные хранилища данных без перекомпилирования приложений. Например, переход от традиционной аутентификации на основе файла /etc/passwd к службе NIS (Network Information Service) прозрачен, поскольку механизм NSS реализован как часть библиотеки C. Для поиска пользователей приложения используют стандартные библиотечные вызовы, такие как `getpwent(3)`, однако при помощи некоторых хитростей в конфигурации данные перенаправляются в другое хранилище, например, NIS.

PAM работает немного иначе, поскольку приложения должны быть специально написаны с учетом того, что им предстоит работать с PAM. Администраторы могут использовать богатый набор библиотек, позволяющих выполнять необходимые настройки ориентированного на работу с PAM приложения; например, чтобы пользователь мог успешно пройти аутентификацию, можно потребовать, чтобы он являлся членом определенной группы и выполнял вход в систему только в определенное время.

При аутентификации пользователей PAM и NSS могут работать в связке. Приложения, ориентированные на работу с PAM, дают указания PAM выполнять проверку учетных данных пользователя. Помимо других ограничений, администратор может настроить PAM на проверку пароля через механизм NSS. PAM используется только для работы с базами password и shadow, и не может работать с другими базами, такими как groups и hosts.

Поддержку PAM и NSS со стороны LDAP обеспечивает Open Source-пакет от PADL Software.

### Настройка NSS на использование LDAP

Функционал NSS реализован в библиотеке C в виде перехватчика традиционных библиотечных вызовов, служащего для получения информации. Библиотека C содержит такие функции, как `getpwent` – для получения информации о пользователе и `gethostbyname(3)` – для получения информации о хосте. Традиционно эти функции были реализованы в виде процедур просмотра файлов etc/passwd и /etc/hosts соответственно. Путем настройки NSS администратор может заставить функцию просмотра информации о хосте также использовать службу DNS (Domain Name Service), при этом приложение даже не будет знать об этих изменениях.

## Понимание NSS

В таблице 3 перечислены базы данных, с которыми работает NSS. Большинство из них имеют соответствующий файл в директории /etc, в которой обычно хранятся данные.

**Таблица 3. Базы данных NSS**

Имя базы данных	Описание
aliases	Почтовые псевдонимы для sendmail, используемые для перенаправления одного локального адреса на другой.
ethers	Сопоставляет Ethernet-адреса IP-адресам. Редко встречается в настоящее время, поскольку теперь для этого существует протокол ARP (Address Resolution Protocol).
group	Содержит список групп и пользователей, являющихся их участниками.
hosts	Сопоставляет IP-адреса именам хостов.
netgroup	Используется для группировки серверов. Наиболее часто используется для безопасности NIS и NFS (Network File System).
networks	Сопоставляет имена сетей их номерам. Используется не часто, поскольку информация об имени сети не представляет большого значения.
passwd	Хранит информацию об учетных записях пользователей, содержащую имя, ID, главную группу, домашний каталог и иногда пароль пользователя.
protocols	Сопоставляет IP-протоколы их именам.
publickey	Используется для распространения ключей NFS и NIS+.
rpc	Сопоставляет имена RPC-функций (Remote Procedure Call) их номерам.
services	Сопоставляет имена служб TCP и UDP номерам портов.
shadow	Защищенный, зашифрованный файл паролей. Обычно в этом файле хранится поле password из файла /etc/passwd, чтобы обеспечить конфиденциальность пароля.

Настройки NSS хранятся в файле /etc/nsswitch.conf, содержащем по одной строке для каждой базы данных из таблицы 3.

**Листинг 1. Пример файла nsswitch.conf**

```
passwd:      files nis
shadow:     files nis
group:      files nis
hosts:      files nis dns
```

В листинге 1 приведен пример настройки четырех таблиц сопоставления: passwd, shadow, group и hosts. За именем таблицы следует знак двоеточия (:) и упорядоченный список методов доступа к данным. Первые три строки листинга 1 одинаковы: первая строка выполняет поиск необходимой информации в файлах, а затем обращается к службе NIS, иногда называемой службой "желтых страниц" (Yellow Pages). Поиск данных в NIS выполняется только в том случае, если ничего не было найдено в файлах. Последняя строка листинга использует для поиска информации о хостах файлы (/etc/hosts), службу NIS, а затем службу DNS.

Для каждого метода, доступного для использования в nsswitch.conf, в директории /lib имеется

соответствующая библиотека, имя которой начинается с **libnss\_**. Например, функционал для работы с файлами содержится в библиотеке /lib/libnss\_files-2.5.so (номер версии не имеет значения, поскольку он распознается и обрабатывается программой динамической линковки ld-linux.so).

## Использование LDAP совместно с NSS

С учетом предыдущего рассмотрения динамических библиотек и формата файла nsswitch.conf вы не должны удивиться тому, что интеграция LDAP и NSS выполняется через совместно используемую библиотеку под названием libnss\_ldap, и что для этого используется ключевое слово **ldap** в файле /etc/nsswitch.conf. Совместно используемая библиотека загружает свою конфигурацию из файла /etc/ldap.conf (не перепутайте с файлом конфигурации OpenLDAP для клиентов командной строки, /etc/openldap/ldap.conf). В листинге 2 показан пример файла ldap.conf.

### Листинг 2. Пример файла ldap.conf для настройки libnss\_ldap

```
# IP-адрес сервера (или адреса, разделенные пробелами)
host 192.168.1.138
# База поиска
base dc=ertw,dc=com
# необязательные учетные данные для привязки
binddn: cn=nssldap,dc=ertw,dc=com
bindpw: letmein
# Если запрос выполняет пользователь root, то вместо этого используйте это имя dn
# Пароль хранится в файле /etc/ldap.secret и доступен для чтения только пользователю root
rootbinddn cn=root,dc=ertw,dc=com
# Указание баз данных passwd, shadow и group в имени DN
# Параметр ?one определяет область
nss_base_passwd ou=People,dc=ertw,dc=com?one
nss_base_shadow ou=People,dc=ertw,dc=com?one
nss_base_group      ou=Group,dc=ertw,dc=com?one
# Отключение поиска вторичных групп для любого из этих пользователей
nss_initgroups_ignoreusers root,ldap,named,avahi,haldaemon,dbus,radvd,tomcat,radiusd
```

В дополнение к содержимому файла /etc/ldap.conf, показанному в листинге 2, вам также необходимо добавить ключевое слово **ldap** в строки passwd, shadow и group файла /etc/nsswitch.conf. Всегда проверяйте, чтобы первым элементом являлось значение **files**; в противном случае может получиться так, что вы будете ожидать ответа от выключенных серверов вплоть до истечения времени ожидания или же можете быть отключенными от вашей системы (если вы были отключены из-за проблемы с файлом nsswitch.conf, загрузитесь в однопользовательском режиме, верните в файле nsswitch.conf обратно значение **files** и выполните перезагрузку).

Можно использовать LDAP для всех баз данных, но только три базы, приведенные в этом листинге, входят в число полезных. Другие базы изменяются редко, и ими следует управлять отдельно. Исключением является база данных **hosts**, которая может использовать LDAP, хотя намного предпочтительнее использовать DNS.

## Проверка

Если вы правильно настроили файлы nsswitch.conf и ldap.conf, то вы должны суметь выполнить вход в систему под учетными данными пользователя LDAP (необходимо, чтобы были доступны следующие атрибуты):

- **uid**: имя для входа в систему

- `uidNumber`: числовой код пользователя (`userid`)
- `gidNumber`: числовой код первичной группы (`groupid`)
- `homeDirectory`: домашний каталог пользователя
- `userPassword`: пароль пользователя, зашифрованный при помощи процедуры `{crypt}` (для генерации пароля используйте `slappasswd`)

Эти и другие атрибуты добавляются через класс объекта `posixAccount`.

Для проверки попробуйте войти в систему с учетными данными пользователя, который есть в вашем каталоге LDAP, но не в локальных файлах паролей. Также вы можете использовать команду `getent passwd` для просмотра всех данных пользователя, о которых знает NSS. Если команда `getent` работает, но вы не можете войти в систему, скорее всего ваш атрибут `userPassword` задан неправильно.

Если вы проверили конфигурацию клиента, но NSS и LDAP все еще не работают вместе, установите уровень регистрации событий `stats` на сервере OpenLDAP и посмотрите, видит ли сервер ваши запросы и разрешены ли они.

## Настройка PAM на использование LDAP

Модуль PAM похож на NSS тем, что он абстрагирует набор библиотечных вызовов от фактической реализации. В отличие от NSS, PAM не замещает существующие вызовы UNIX; вместо этого он предоставляет ряд новых вызовов, которые могут использоваться приложениями.

### Понимание PAM

PAM реализован в виде библиотеки, которую используют приложения. Приложения обращаются к этой библиотеке для использования функций PAM, выполняющих проверку аутентификации, управление учетными записями, сессиями и паролями.

Проверка аутентификации является основной задачей PAM. Обращаясь к библиотекам PAM, приложения узнают, аутентифицирован ли пользователь. Библиотеки PAM, в свою очередь, в соответствии с правилами, установленными системным администратором, запрашивают пароли пользователей или выполняют любые другие проверки.

Управление учетной записью вступает в силу после того, как пользователь укажет действующие учетные данные, и отвечает за проверку того, разрешена ли эта учетная запись. Учетная запись может быть не разрешена в определенные промежутки времени или для некоторых приложений.

Управление сеансом предоставляет приложению возможность настройки среды пользователя после его успешного входа. Часто нужно предоставить вошедшему в консоль пользователю некоторые дополнительные разрешения, такие как использование локального дисковода компакт-дисков или других устройств; это достигается на уровне управления сеансом.

Наконец, управление паролем предоставляет гибкие способы изменения паролей. Как вы скоро увидите, эта функциональная возможность позволяет пользователям менять их пароли LDAP при помощи знакомой программы `passwd(1)`. Управление паролями PAM также позволяет вам задавать политики надежности паролей, которые работают независимо от того, какое хранилище паролей используется.

Чтобы настроить PAM на совместную работу с какой-либо службой, вы должны создать файл с именем этой службы в каталоге `/etc/pam.d`, например, `/etc/pam.d/sshd` – для службы `sshd`. Это не твердое правило, поскольку каждое приложение определяет свое собственное имя для службы PAM. Если вы не уверены, какое имя использовать, используйте имя бинарного файла и проверьте журналы регистрации на предмет ошибок.

Каждый файл конфигурации в каталоге `/etc/pam.d` задает упорядоченный список инструкций

для каждой управляющей функции PAM. Каждая строка этого файла имеет следующий формат: **function control module arguments**. Поле **function** содержит управляющую функцию, определяемую ключевыми словами **auth**, **account**, **session** и **password**.

Поле **control** задает способ использования полученного в результате проверки значения, и определяется следующими ключевыми словами:

- **required** – чтобы функция оказалась успешной, необходимо, чтобы проверка также была выполнена успешно. Если выполнение этой проверки завершается неудачей, PAM продолжает проверять остальные инструкции для данной функции, но полученные результаты ни на что не влияют.
- **requisite** – чтобы функция оказалась успешной, необходимо, чтобы проверка также была выполнена успешно. Если выполнение этой проверки завершается неудачей, PAM прекращает дальнейшую проверку остальных инструкций и возвращает статус неудачи.
- **sufficient** – если эта проверка оказывается успешной, обработка прекращается и функция возвращает статус успеха при условии, что все предыдущие элементы с уровнем "required" успешно прошли проверку. Если выполнение этой проверки завершается неудачей, ее результат игнорируется, и обработка продолжается.
- **optional** – результаты проверки игнорируются.

Поля **module** и **arguments** определяют способ выполнения самой проверки. Один и тот же модуль может реализовывать одну или несколько вышеописанных функций, поэтому он может перечисляться несколько раз. Одним из модулей, который вы будете видеть очень часто, является модуль **pam\_stack**, позволяющий вам вызвать стеки команд из других файлов. В листинге 3 показан PAM-файл, в котором используется **pam\_stack**.

### Листинг 3. Использование модуля **pam\_stack** для вызова других стеков команд

```
auth      required      pam_nologin.so
auth      required      pam_stack.so service=system-auth
account   required      pam_stack.so service=system-auth
session   required      pam_stack.so service=system-auth
password  required      pam_stack.so service=system-auth
```

В листинге 3 приведен формат PAM-файла. Функция **auth** содержит две строки, обе из которых имеют статус "required" и поэтому обязаны успешно пройти проверку, чтобы аутентификация была успешной. Первая строка **auth** обращается к модулю **pam\_nologin**, задачей которого является возврат статуса неудачи в том случае, если вход в систему пытается выполнить не-root пользователь и существует файл `/etc/nologin`. Следующая строка **auth** обращается к модулю **pam\_stack** и передает ему аргумент **service=system-auth**. Затем модуль **pam\_stack.so** считывает содержимое файла `/etc/pam.d/system-auth` и проверяет все инструкции, соответствующие функции **auth**. Если все проверки выполняются успешно, **pam\_stack** возвращает статус успеха назад в файл из листинга 3.

Остальные три функции — **account**, **session** и **password** — ссылаются только на модуль **pam\_stack** и службу **system-auth**. Если соответствующие функции **system-auth** возвращают статус успеха, результат проверки также считается успешным.

Многие системы используют общие методы аутентификации, поэтому модуль **pam\_stack** используется в большинстве файлов со службой **system-auth** (или эквивалентной),

содержащей все интересующие детали. В оставшейся части этого раздела файл system-auth будет одним из файлов, использующихся для интеграции LDAP в PAM-процесс.

## Использование LDAP совместно с PAM

Для настройки как модуля NSS, так и модуля PAM используется файл /etc/ldap.conf, поэтому, если вы следите этому руководству, вы уже на полпути к работающей системе PAM-LDAP. Можно совместно использовать NSS и PAM таким образом, что выполнять аутентификацию в LDAP смогут как приложения, ориентированные на работу с PAM, так и старые приложения. В дополнение к функционалу NSS, PAM предоставляет некоторые новые возможности, включая следующие:

- Изменение пароля пользователями
- Более детальная конфигурация требований аутентификации
- Поддержка большего числа типов шифрования
- Централизованное администрирование учетных записей пользователей

Убедитесь, что в файле /etc/ldap.conf присутствует строка `pam_password md5`, и удалите все остальные строки `pam_password`, если они есть. Это укажет библиотеке pam\_ldap на то, что при изменении паролей необходимо выполнять их локальное хэширование по алгоритму MD5 (Message Digest 5), прежде чем отправлять их на сервер LDAP.

Отредактируйте файл /etc/pam.d/system-auth (или его эквивалент) и добавьте ссылки на модуль pam\_ldap, как показано в листинге 4. Строки следует добавлять после каждого упоминания модуля pam\_unix (чтобы локальные учетные записи имели более высокий приоритет по сравнению с учетными записями LDAP), но перед любыми упоминаниями модулей pam\_allow и pam\_deny (которые предоставляют разрешения и запреты по умолчанию).

### Листинг 4. Новая конфигурация system-auth, использующая pam\_ldap

```
auth      sufficient  pam_unix.so nullok try_first_pass
auth      sufficient  pam_ldap.so use_first_pass
auth      required    pam_deny.so

account   required    pam_unix.so broken_shadow
          account      sufficient  pam_ldap.so
account   required    pam_permit.so

password  requisite   pam_cracklib.so try_first_pass retry=3
password  sufficient  pam_unix.so md5 shadow nullok try_first_pass use_authtok
password  sufficient  pam_ldap.so use_authtok
password  required   pam_deny.so

session   required   pam_limits.so
session   required   pam_unix.so
session   optional   pam_ldap.so
```

Строки, выделенные **жирным шрифтом**, являются дополнением к файлу конфигурации PAM. Обратите внимание на добавление аргумента `broken_shadow` для функции `account` модуля `pam_unix`. Благодаря этому модуль `pam_unix.so` не возвращает статус неудачи в случае, если пользователь не имеет записи в базе `shadow` (этой записи не существует потому, что учетная запись хранится в LDAP).

Аргумент `use_first_pass` для функции `auth` модуля `pam_ldap` заставляет `pam_ldap.so`

использовать пароль, полученный от `pam_unix.so`, вместо того чтобы запрашивать новый пароль. Аргумент `use_authok` делает то же самое для функции `password`.

Новая конфигурация позволяет использовать для авторизации как пароли UNIX, так и пароли LDAP, то есть, первый найденный действующий пароль позволяет пользователю войти в систему. Если никакой из паролей не подходит (возвращается либо ошибка, либо сообщение "no such user"), `pam_deny` возвращает ошибку.

## Проверка

Попытайтесь изменить пароль пользователя с помощью команды `passwd`, а затем проверьте, что пароль изменился в каталоге LDAP. Наконец, убедитесь в том, что пользователь, как и прежде, может войти в систему.

Если вы смогли заставить работать NSS, PAM также должен работать. Наиболее вероятными причинами ошибок могут стать опечатки в строках конфигурации PAM, помещение строк в неправильное место файла или не в тот файл.

## Миграция с NIS на LDAP

В этом разделе описывается материал по теме 305.2 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 1.

Из этого раздела вы узнаете, как:

- Анализировать структуру NIS перед миграцией на LDAP
- Анализировать структуру NIS перед интеграцией с LDAP
- Автоматизировать миграцию из NIS на LDAP
- Создавать шлюз с NIS на LDAP

Служба NIS является традиционным методом централизованной аутентификации UNIX-машин; она просто настраивается и хорошо работает. Хотя аутентификация LDAP является более сложной, она имеет следующие преимущества по сравнению с NIS:

- LDAP безопаснее, чем NIS, поскольку позволяет шифровать трафик и ограничивать доступ к базе данных.
- LDAP может хранить больше чем просто данные для аутентификации, тогда как возможности NIS ограничены.
- LDAP поддерживает большее количество клиентов, чем NIS.

Вы можете полностью заменить NIS на LDAP или использовать их одновременно. В последнем случае LDAP будет являться каноническим источником данных, а сервер NIS будет использовать данные LDAP вместо локальных файлов. Этот метод хорошо подходит для долговременной миграции или для поддержки старых операционных систем, которые не смогут работать с LDAP.

## Первый подход: миграция на LDAP

Основные действия при выполнении миграции с NIS на LDAP заключаются в следующем:

1. Определить, какие базы данных NIS необходимо заменить.
2. Загрузить данные NIS в LDAP.
3. Перенастроить клиенты на использование LDAP вместо NIS.

В промежуток времени между началом шага 2 и завершением шага 3 вы будете иметь две активные базы данных, к которым не выполнено подключений. Любые изменения, такие как добавление пользователя или изменения его пароля, должны производиться в обеих базах данных, иначе в ваших данных могут возникнуть противоречия. Вы можете "заморозить" все изменения или же воспользоваться стратегией интеграции, описанной в следующем разделе.

## Анализ существующей структуры NIS

Перед тем как начать миграцию, вы должны определить, какие базы данных управляются службой NIS. Подключитесь к главному серверу NIS и просмотрите каталог базы данных. В большинстве операционных систем файлы хранятся в каталоге `/var/yp/имя_домена`.

### Листинг 5. Определение баз данных, обслуживаемых NIS

```
# ls /var/yp/`domainname`  
group.bygid  
groupbyname  
hosts.byaddr  
hostsbyname  
mail.aliases  
netidbyname  
passwdbyname  
passwdbyuid  
protocolsbyname  
protocolsbynumber  
rpcbyname  
rpcbynumber  
servicesbyname  
servicesbyservicename  
ypservers
```

В листинге 5 используется команда `domainname` для отображения имени домена. Когда команда заключена в обратные кавычки (`), результат ее выполнения вставляется в командную строку. За исключением файла `ypservers`, каждый файл в этом каталоге представляет собой базу данных NIS. Составьте список уникальных имен баз данных, чтобы определить, какие из них должны быть перемещены в LDAP. NIS хранит одни и те же данные с различными ключами поиска, такими как поиск по имени и поиск по идентификатору пользователя (UID), как, например, в случае с файлом паролей; в данном случае обе базы представляют собой базу данных паролей. Другие случаи могут быть неочевидными: например, `mail.aliases` - это таблица псевдонимов `aliases`. Если вы сомневаетесь, загляните в файл `/var/yp/Makefile`, чтобы определить источник базы данных.

После того как вы проверите сервер, вы можете захотеть проверить некоторые из ваших NIS-клиентов, чтобы определить, какие таблицы сопоставлений они используют. Для этого ищите ключевое слово `nis` в файле `/etc/nsswitch.conf`. Вероятно, вы обнаружите, что ваш сервер хранит больше таблиц сопоставления по сравнению с используемыми.

## Использование инструментов миграции

Самыми популярными инструментами для миграции данных NIS в LDAP являются инструменты, разработанные программистами компании PADL software – авторами `ram_ldap`, `nss_ldap` и шлюза NIS-LDAP, речь о котором пойдет позднее. С большой вероятностью эти инструменты включены в ваш дистрибутив; если же нет, то ссылки на них вы можете найти в разделе [Ресурсы](#). Инструменты миграции от PADL могут получать данные из локальных файлов, NIS или NIS+ и загружать их на ваш сервер LDAP.

Перед началом использования инструментов PADL у вас должен быть работающий сервер LDAP, не содержащий никаких данных. Все необходимые записи будут сгенерированы автоматически, а вы избежите дублирования записей.

Инструменты миграции состоят из набора shell- и perl-скриптов. В системах RedHat скрипты являются частью пакета `openldap-servers`, и их можно найти в каталоге

/usr/share/openldap/migration. Пользователи Debian также могут захотеть получить пакет **migrationtools**. Ищите файл под названием `migrate_base.pl` или загрузите последнюю версию с Web-сайта PADL.

Эти скрипты получают данные из различных источников, конвертируют их в LDIF-формат и затем добавляют их на ваш сервер. Данные добавляются с помощью команды `ldapadd` в онлайновом режиме, а также с помощью команды `slapadd` – в автономном режиме, поэтому в первом случае вам понадобятся административные учетные данные, а во втором случае вам придется остановить сервер LDAP.

Предварительно может быть полезно настроить некоторые переменные среды, задав имена базового домена (DN) дерева и учетной записи администратора (root DN). В листинге 6 показаны команды оболочки bash для подготовки к миграции домена `ertw.com`.

#### Листинг 6. Настройка переменных среды для подготовки к миграции в LDAP

```
export LDAP_BASEDN="dc=ertw,dc=com"
export LDAP_BINDDN="cn=root,dc=ertw,dc=com"
export LDAP_DEFAULT_MAIL_DOMAIN=ertw.com
```

Первая строка в листинге 6 – это отличительное имя базового объекта (base DN) дерева LDAP, которое позже будет использоваться для генерации всех отличительных имен. Вторая строка – это имя административной учетной записи root (root DN). Пароль вам понадобится только в том случае, если вы используете онлайновый режим. Последняя строка задает имя домена по умолчанию для адресов электронной почты. Некоторые из утилит не будут запрашивать у вас эту информацию, поэтому, указав все необходимые сведения здесь, вы оградите себя от проблем, которые могут возникнуть позже.

Утилиты разделены на две категории. Названия файлов первой из них начинаются с `migrate_all_`. Во вторую категорию включены все оставшиеся файлы, имена которых начинаются на `migrate_` и содержат название файла или базы данных. Скрипты, входящие в первую категорию, используются для сбора данных в единое целое, а скрипты из второй категории предназначены для преобразования данных из исходного формата в LDIF-формат.

У вас есть два варианта. Вы можете использовать один из скриптов `migrate_all_`, который автоматически соберет данные всех общих баз данных из указанного местоположения (NIS, файлы, NIS+m, и так далее), или же вы можете самостоятельно собрать только нужные вам данные и использовать отдельные скрипты для их преобразования в LDIF. Первый способ, если он работает, является более простым. В листинге 7 показано использование скрипта `migrate_all_nis_online.sh` для выполнения миграции всех данных из NIS в каталог LDAP в онлайновом режиме.

#### Листинг 7. Выполнение миграции данных с NIS на LDAP с использованием скрипта `migrate_all_nis_online.sh`

```
[root@server1 migration]# ./migrate_all_nis_online.sh
Enter the NIS domain to import from (optional):
No such map networks.byaddr. Reason: Internal NIS error
Enter the hostname of your LDAP server [ldap]: localhost
Enter the credentials to bind with: mypassword
Do you wish to generate a DUAConfigProfile [yes|no]? no

Importing into dc=ertw,dc=com...
```

```

Creating naming context entries...
Migrating groups...
Migrating hosts...
Migrating networks...
Migrating users...
Migrating protocols...
Migrating rpcs...
Migrating services...
Migrating netgroups...
Migrating netgroups (by user)...
sh: /etc/netgroup: No such file or directory
Migrating netgroups (by host)...
sh: /etc/netgroup: No such file or directory
adding new entry "dc=ertw,dc=com"

Importing into LDAP...
adding new entry "ou=Hosts,dc=ertw,dc=com"
..... output omitted ...
adding new entry "cn=rquotad,ou=Rpc,dc=ertw,dc=com"

adding new entry "cn=rquotad,ou=Rpc,dc=ertw,dc=com"
ldap_add: Already exists (68)

/usr/bin/ldapadd: returned non-zero exit status: saving failed LDIF to
/tmp/nis.ldif.X17515

```

Листинг 7 начинается с запуска скрипта `migrate_all_nis_onlinesh`, который собирает данные NIS, преобразует их в LDIF-формат и затем использует команду `ldapadd` для импорта данных. Сначала скрипт запрашивает имя NIS-домена; вы можете нажать клавишу **Enter**, если используете в системе NIS-домен по умолчанию. Затем скрипт импортирует данные NIS (в этой системе была возвращена некритическая ошибка, поскольку таблица `networks` не использовалась). Далее скрипт предлагает ввести информацию о сервере LDAP, такую как имя хоста и пароль (имя для привязки (bind DN) и имя базового объекта (base DN) получены из переменных среды, которые были указаны в листинге 6). Не следует соглашаться на выполнение импорта объекта `DUAConfigProfile`, если у вас нет поддерживающей эту функцию схемы, что маловероятно.

Если на этом этапе вы начнете получать ошибки, сообщающие о неправильном синтаксисе отличительных имен, убедитесь, что вы импортировали файл `nis.schema` внутри `slapd.conf`.

Если ваша схема не содержит ошибок, будет выполнен импорт данных в ваш каталог LDAP. Существует вероятность того, что выполнение скрипта закончится с ошибкой, подобной той, что показана в конце листинга 7. Из-за способа хранения данных в NIS вы можете получить задублированные записи в некоторых базах данных. Это нормально для NIS, но не для LDAP. Ниже представлено несколько способов решения этой проблемы в зависимости от ваших задач:

- Отредактируйте LDIF-файл (в данном случае `/tmp/nis.ldif.X17515`), чтобы удалить задублированные данные, а затем удалите вашу базу данных LDAP и импортируйте файл.
- Укажите команде `ldapadd` игнорировать ошибки, используя для этого параметр `-C: export LDAPADD="/usr/bin/ldapadd -c"` (заметьте, что скрипт все равно сообщит об ошибке, но данные будут импортированы).
- Отредактируйте скрипт `migrate_all_nis_online.sh` и установите переменные

`ETC_SERVICES`, `ETC_PROTOCOLS` и `ETC_RPC` равными `/dev/null` вместо использования временного файла. Это приведет к тому, что база данных будет исключена из обработки (обратите внимание на то, что переменные некоторых из скриптов `migrate_all_` могут быть переопределены переменными среды, но не вариантом NIS).

- Не запускайте скрипт `migrate_all_nis_online.sh` и выполните миграцию вручную.

Первые три способа говорят сами за себя и эффективны в тех случаях, когда вас устраивают результаты (такие как отсутствие в каталоге LDAP протоколов, RPC и служб, как в третьем случае). Четвертый способ требует некоторых разъяснений.

Если все, что вам нужно – это переместить в LDAP пользователей и группы, вы можете самостоятельно скопировать файлы и сгенерировать LDIF с помощью других скриптов, и использовать `ypcat` для извлечения данных из NIS. Этот процесс показан в листинге 8.

### Листинг 8. Выполнение миграции пользователей и групп вручную

```
[root@server1 migration]# ypcat passwd > /tmp/passwd.tmp
[root@server1 migration]# ypcat group > /tmp/group.tmp
[root@server1 migration]# ./migrate_base.pl > /tmp/ldif
[root@server1 migration]# ./migrate_passwd.pl /tmp/passwd.tmp >> /tmp/ldif
[root@server1 migration]# ./migrate_group.pl /tmp/group.tmp >> /tmp/ldif
[root@server1 migration]# ldapadd -x -D "cn=root,dc=ertw,dc=com" \
    -w "mypassword" -f /tmp/ldif
adding new entry "dc=ertw,dc=com"
adding new entry "ou=Hosts,dc=ertw,dc=com"
.... дальнейший вывод опущен ...
```

В первых двух строках листинга 8 используется команда `ypcat` для выгрузки данных NIS в файлы, расположенные в каталоге `/tmp`. Следующие три строки генерируют LDIF. Команда `migrate_base` создает некоторые основные элементы дерева каталога, а следующие две строки преобразуют файлы `password` и `group` в LDIF-формат. Обратите внимание на использование оператора `>>`, благодаря чему, результирующий файл будет содержать вывод всех трех скриптов миграции. Наконец, происходит вызов команды `ldapadd` для импортирования данных.

Независимо от того, каким способом миграции вы воспользуетесь, сделайте несколько простых запросов, чтобы убедиться, что вы можете видеть все данные. Убедитесь, что вы можете видеть хэши паролей (для этого используйте учетную запись администратора `root` DN, поскольку ваш список контроля доступа может не позволять просматривать пароли).

На этом этапе все данные NIS находятся в каталоге LDAP. До тех пор, пока будут использоваться ваши NIS-клиенты, все изменения в NIS должны реплицироваться в LDAP и наоборот.

### Перемещение клиентов и проверка результатов

Перенастройка клиентов – довольно простая задача, заключающаяся в настройке NSS и PAM на клиенте. Об этом детально рассказывалось в [предыдущем разделе](#). Вкратце, вы указываете в файле `/etc/ldap.conf` информацию о вашем сервере и редактируете файл `/etc/nsswitch.conf`, заменяя в нем `nis` на `ldap`. Если вы настраиваете PAM, вам понадобится отредактировать соответствующие файлы в директории `/etc/pam.d` и добавить в них ссылки на модуль `pam_ldap.so`.

На каждом клиентском компьютере войдите в систему под учетной записью обычного

пользователя и выполните команду `getent` для баз данных, которые вы переместили в LDAP.

## Второй подход: интеграция с LDAP

Второй подход основан на совместном использовании NIS и LDAP. Это может оказаться полезным, если у вас имеются клиенты, которые не понимают LDAP (не имеют собственного модуля LDAP или не поддерживают PAM), или если вы хотите осуществлять переход на LDAP в течение длительного срока. Основные действия в случае совместного использования NIS/LDAP почти такие же, что и в случае перехода от NIS к LDAP:

1. Определить, какие базы данных NIS используются.
2. Загрузить данные NIS в LDAP.
3. Заменить ваши серверы NIS на `ypldapd`.
4. Перенастроить клиенты на использование LDAP.

Для клиентов, которые будут продолжать использовать NIS, не требуется никаких изменений, поскольку `ypldapd` является полнофункциональным сервером NIS. Единственное отличие между ним и стандартным сервером `ypserv`, входящим в состав вашей операционной системы, заключается в том, что `ypldapd` получает данные из каталога LDAP, а не из локальных файлов.

Первые два шага аналогичны тем, что были описаны в первом случае, поэтому начнем с шага 3.

### Замена серверов NIS на `ypldapd`

`ypldapd` представляет собой демон сервера NIS, использующий получения данных LDAP, а не файлы баз данных из каталога `/var/yp`. Это коммерческий программный продукт от компании PADL, но вы можете получить 30-дневную пробную лицензию, связавшись с PADL по электронной почте (обратитесь к разделу [Ресурсы](#)). Установить `ypldapd` достаточно просто:

1. Распакуйте tar-архив в каталог `/opt/ypldapd`.
2. Скопируйте лицензию в файл `/opt/ypldapd/etc/padlock.ldif`.
3. Отредактируйте конфигурационный файл `/opt/ypldapd/etc/ypldapd.conf`.
4. Остановите существующий сервер NIS.
5. Запустите `ypldapd`.

Сначала выполните команду `mkdir -p /opt/ypldapd` от имени пользователя `root`, чтобы создать каталог `ypldapd` (а также каталог `/opt`, если он не существует). Перейдите в этот каталог (`cd /opt/ypldapd`) и распакуйте tar-файл дистрибутива `ypldapd` с помощью команды `tar -xzf /tmp/ypldapd_linux-i386.tar.gz`. В результате файлы `ypldapd` будут помещены в соответствующий каталог.

Вам будет предоставлена лицензия, которую вы поместите в файл `/opt/ypldapd/etc/padlock.ldif`. Если вы получили ее по электронной почте, убедитесь, что ваш почтовый клиент не разрывает длинные строки: ключ должен содержать четыре строки, состоящих из пар **атрибут:значение**.

Конфигурационный файл `ypldapd` должен располагаться в каталоге `/opt/ypldapd/etc/ypldapd.conf`. В этом каталоге вы обнаружите файл `ypldapd.conf.sample`, который вы можете скопировать и использовать как основу. Как и в случае с другими утилитами, с которыми вы имели дело до сих пор, вам потребуется указать информацию о вашем сервере LDAP. В листинге 9 показан пример файла `ypldapd.conf`.

### Листинг 9. Пример файла `ypldapd.conf`

```
# Имя NIS-домена
ypdomain ertw
# Сервер LDAP и отличительное имя базового объекта (base DN)
ldaphost localhost
basedn dc=ertw,dc=com
# Учетные данные... Пользователь должен иметь разрешение на чтение атрибута userPassword
binddn cn=ypldapd,dc=ertw,dc=com
bindcred mypassword
# Сопоставление баз данных NIS с отличительными именами DN (относительно basedn)
# Если вы использовали инструменты миграции, у вас не должно возникнуть
# необходимости что-либо менять
namingcontexts namingcontexts.conf
# Должен ли ypldapd кэшировать данные?
caching on
# Время жизни кэша в минутах
cache_dump_interval 15
# Должен ли пароль быть скрыт?
hide_passwords off
# Сколько серверов ypldapd могут работать одновременно?
maxchildren 5
```

Когда ваш файл `ypldapd.conf` будет готов, вы можете остановить все экземпляры `ypserv` и выполнить команду `sbin/ypldapd`, которая запустит `ypldapd` в фоновом процессе.

### Перемещение клиентов и проверка результатов

Для проверки вашего нового сервера NIS выполните команду `ypwhich`, которая покажет, к какому серверу NIS вы привязаны. Если вы получите ошибку, убедитесь, что не запущены никакие другие экземпляры `ypserv`, и что выполняется только один демон `ypldapd`. После этого попытайтесь получить таблицу сопоставления, выполнив команду `ypcat passwd` (при условии, что на сервере также запущен и клиент).

Клиенты, которые будут продолжать использовать NIS, должны также уметь выполнять на новом сервере команды `ypwhich` и `ypcat`. Команды для клиентов, которые полностью перейдут на использование LDAP, приведены в разделе о миграции.

### Интеграция LDAP и служб UNIX

В этом разделе описывается материал по теме 305.3 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 1.

Из этого раздела вы узнаете, как:

- Интегрировать SSH и LDAP
- Интегрировать FTP и LDAP
- Интегрировать HTTP и LDAP
- Интегрировать FreeRADIUS и LDAP
- Интегрировать службы печати и LDAP

Большинство приложений будут корректно работать с LDAP, если вы настроили NSS и PAM. Некоторым приложениям необходимо указать на использование PAM или обеспечить дополнительный функционал для правильного доступа к LDAP. В этом разделе мы сосредоточимся на рассмотрении распространенных служб UNIX, а также того, каким образом эти службы поддерживают интеграцию с LDAP.

### Интеграция SSH и LDAP

OpenSSH интегрируется с LDAP через PAM, если дистрибутив был скомпилирован с поддержкой необходимого функционала. Запустите команду `ldd /usr/sbin/sshd |`

`grep pam`, чтобы проверить, были ли выполнена компоновка с поддержкой совместно используемых библиотек PAM. Если нет, необходимо перекомпилировать `sshd` с параметром `--with-pam`.

Для использования PAM убедитесь, что у вас имеется конфигурационный PAM-файл с именем `/etc/pam.d/sshd`. В листинге 10 показан пример PAM-файла, использующего стек `system-auth`.

### Листинг 10. Пример файла `/etc/pam.d/system-auth`

```
auth      required      pam_stack.so service=system-auth
account   required      pam_stack.so service=system-auth
password  required      pam_stack.so service=system-auth
session   required      pam_stack.so service=system-auth
```

Когда ваш конфигурационный PAM-файл будет готов, вы можете настроить `sshd` для работы с PAM. В файле `/etc/ssh/sshd_config` добавьте строку `UsePAM yes` и перезапустите `sshd`.

### Интеграция FTP и LDAP

Существует множество демонов FTP, и неясно, какие из них фигурируют в экзамене LPIC 3.

Простейшим методом интеграции является метод, основанный на использовании интеграции NSS. Когда FTP-сервер выполняет проверку пароля, функционал NSS использует LDAP.

Как правило, современные FTP-серверы поддерживают PAM. В этом случае вы создаете ваш конфигурационный PAM-файл в каталоге `/etc/pam.d`. Обычно этот файл называется `ftp`, но в зависимости от конкретной программы и от дистрибутива он может называться иначе. Например, RedHat указывает демону `vsftpd` использовать файл `/etc/pam.d/vsftpd` вместо файла по умолчанию `/etc/pam.d/ftp`.

Как только демон `ftp` находит свой конфигурационный PAM-файл, он обрабатывает его так же, как и любой другой PAM-клиент. Конфигурации, приведенной в листинге 10, достаточно для запуска демона. Также можно использовать параметры `pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed` и `pam_shells` на стадии `auth`, чтобы ограничить круг пользователей, которые могут подключаться, и доступные оболочки, наподобие того, как это делали старые FTP-серверы.

### Интеграция HTTP и LDAP

В состав Web-сервера Apache входят модули, выполняющие базовую HTTP-аутентификацию с использованием хранилища LDAP вместо традиционного файлового хранилища `htpasswd`. Этот функционал реализуется через модули `mod_authnz_ldap` и `mod_ldap`. Первый модуль предоставляет механизмы для аутентификации Web-пользователя с использованием данных LDAP; второй модуль, `mod_ldap`, предоставляет интерфейс доступа к LDAP для модуля `mod_authnz_ldap` (или для всех будущих модулей на основе LDAP), включая объединение и кэширование подключений.

Все инструкции в этом разделе относятся к Apache версии 2.2. Если вы используете Apache 2.0, вместо модуля `mod_authnz_ldap` используется модуль `mod_auth_ldap`.

Конфигурации этих двух модулей одинаковы.

Оба модуля – `mod_ldap` и `mod_authnz_ldap` являются частью дистрибутива Apache. Если вы компилируете Web-сервер вручную, необходимо указать параметры `--enable-authnz-ldap --enable-ldap` при выполнении команды `configure`. Если вы используете версию Apache из своего дистрибутива, установите соответствующий модуль (в дистрибутивах Red Hat модули являются частью базового пакета `httpd`).

Когда пользователь выполняет запрос к защищенному ресурсу, Apache возвращает код ошибки 401 (unauthorized). В этот момент Web-обозреватель должен предложить пользователю ввести имя и пароль. После этого обозреватель выполняет повторный запрос, содержащий полученную информацию, зашифрованную в заголовке **Authorization**. Если имя пользователя и пароль принимаются Web-сервером, пользователь получает доступ к странице, в противном случае сервер снова возвращает код ошибки 401.

Когда Apache настроен на проверку паролей, хранящихся в LDAP, он сначала привязывается к серверу, используя заранее определенную учетную запись, и ищет полученное отличительное имя DN пользователя. Затем сервер выполняет повторную привязку в качестве этого пользователя, используя полученный пароль. Если сервер успешно привязывается к серверу с использованием учетных данных этого пользователя, аутентификация считается успешной.

После аутентификации пользователя сервер может выполнять дополнительные задачи, такие как проверка имени DN или атрибута, или результатов прохождения пользователем поискового фильтра. Если настроена какая-либо из таких проверок, то она должна быть успешно пройдена для того, чтобы авторизация также считалась успешной.

Настройка модуля **mod\_authnz\_ldap** аналогична стандартному методу аутентификации с использованием текстовых файлов. В листинге 11 показан простейший пример настройки LDAP-аутентификации без использования авторизации.

### Листинг 11. Конфигурация Web-сервера Apache для выполнения LDAP-аутентификации

```
LoadModule ldap_module modules/mod_ldap.so
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so

<Location /protected>
    AuthType basic
    AuthName ProtectedByLDAP
    AuthBasicProvider ldap

    AuthLDAPUrl ldap://192.168.1138/ou=People,dc=ertw,dc=com?uid
    # Anon bind for first phase
    #AuthLDAPBindDN
    #AuthLDAPBindPassword

    AuthzLDAPAuthoritative off
    require valid-user
</Location>
```

Первые две строки листинга 11 загружают требуемые модули на Web-сервер. Оставшаяся часть конфигурации заключена в контейнер **Location**, что означает, что данная конфигурация применяется только к запросам, начинающимся с **/protected**. Сначала объявляется тип аутентификации (**basic**) и ее имя (**ProtectedByLDAP**). Это имя отображается пользователю в Web-браузере. Стока **AuthBasicProvider** указывает Apache на то, что аутентификация осуществляется посредством LDAP.

Листинг 11 продолжает строка **AuthLDAPUrl**, которая предоставляет Apache сведения о сервере LDAP. Аргумент этой строки имеет следующий вид:

**ldap://host:port/basedn?attribute?scope?filter**. Параметры **host** и **port** указывают на сервер LDAP, а параметр **basedn** является именем базового объекта (base DN), с которого выполняется начальный поиск. Параметр **attribute** указывает на атрибут,

поиск которого будет выполняться наряду с именем пользователя во время начального поиска (по умолчанию `uid`). Параметр `scope` имеет значение `one` или `sub`, определяющий соответствующий уровень поиска. Параметр `filter` является необязательным фильтром, который будет связан логической операцией "И" с условиями поиска заданной комбинации пользователь/атрибут.

Строки `AuthLDAPBindDN` и `AuthLDAPBindPassword` в листинге 11 закомментированы, вследствие чего выполняется анонимная привязка. При желании вы можете указать здесь учетные данные пользователя. В любом случае пользователь, выполняющий начальную привязку, должен иметь разрешение на чтение атрибута, указанного в команде `AuthLDAPUrl`.

Последние две строки отключают авторизацию, разрешая доступ любому проверенному пользователю. Стока `AuthzLDAPAuthoritative off` означает, что впоследствии модуль может разрешать доступ даже в тех случаях, если LDAP отклоняет авторизацию (но не аутентификацию). Стока `require valid-user` получена из другого модуля, поэтому здесь она необходима. Вместо этих двух строк вы можете использовать строки, относящиеся к LDAP, например, выполняющих проверку принадлежности к группе или LDAP-атрибута. В листинге 12 показана измененная часть конфигурации из листинга 11, разрешающая доступ только тем пользователям, чья учетная запись содержит атрибут/значение `ou=Engineering`.

## **Листинг 12. Предоставление доступа только участникам определенного организационного подразделения (OU)**

```
AuthzLDAPAuthoritative on  
require ldap-filter ou=engineering
```

В листинге 12 следует обратить внимание на две вещи. Во-первых, параметр `AuthzLDAPAuthoritative` теперь включен (это значение по умолчанию), поскольку ваши требования могут быть обработаны модулем LDAP. Во-вторых, аргумент `ldap-filter` не заключен в круглые скобки. Apache использует указанный фильтр LDAP, выполняет логическую операцию "И" для него и для атрибута `uid` (или любого другого атрибута, указанного в команде `AuthLDAPUrl`) и строит фильтр поиска, вставляя одну строку. Если ваш фильтр будет содержать дополнительные кавычки или скобки, результирующий запрос станет неправильным, и аутентификация завершится с ошибкой, которая будет зарегистрирована на сервере в журнале `error_log`.

## **Интеграция FreeRADIUS и LDAP**

FreeRADIUS – это RADIUS-сервер (Remote Authentication Dial In User Service) с открытым кодом, который часто используется для аутентификации коммутируемых или других сетевых устройств. Клиенты используют RADIUS для аутентификации пользователей, а RADIUS-сервер, в свою очередь, использует LDAP для поиска информации.

Вы можете интегрировать PAM и FreeRADIUS двумя способами: использовать PAM или включить встроенную поддержку LDAP через модуль `rlm_ldap`. Выбор зависит от того, каким образом вы планируете использовать RADIUS. Если вам необходима только аутентификация, или вы не желаете модифицировать вашу схему LDAP, используйте PAM. Если вам необходимо использовать атрибуты RADIUS, то проще настроить модуль LDAP и хранить атрибуты в LDAP (RADIUS разрешает серверу посыпать сведения о конфигурации устройству, запрашивающему аутентификацию, что позволяет вам предоставлять различные сервисы различным пользователям).

Если вы используете PAM, убедитесь, что у вас имеются настройки PAM для LDAP, аналогично настройкам для других систем. Конфигурационный PAM-файл для FreeRADIUS – /etc/pam.d/radiusd. Взяв за основу имеющиеся конфигурационные файлы FreeRADIUS, раскомментируйте ключевое слово `pam` в разделе `authenticate` файла radiusd.conf. Затем откройте для редактирования файл users и найдите в нем строку `DEFAULT Auth-Type = System`. Измените ключевое слово `System` на PAM. Перезапустите radiusd, после чего все будет готово.

Использование встроенного модуля LDAP, `rlm_ldap`, является более сложным вариантом. Сначала вы должны установить в вашей системе FreeRADIUS, скомпилированный с поддержкой модуля `rlm_ldap` (параметр `--enable-ldap`). Компиляция FreeRADIUS выполняется так же, как и для других пакетов, и здесь мы не будем описывать этот процесс. Если ваш дистрибутив Linux имеет в своем составе пакет FreeRADIUS, то, вероятнее всего, он содержит и модуль LDAP.

В состав FreeRADIUS входит схема LDAP, которая находится в файле openldap.schema. Скопируйте этот файл в каталог /etc/openldap/schema/freeradius.schema и импортируйте его в OpenLDAP через директиву `include` в файле slapd.conf. Схема содержит несколько атрибутов и два класса объектов (objectClasses). Одним из этих классов является `radiusprofile`; он используется для всех пользователей, которые будут проходить аутентификацию RADIUS. Класс `radiusprofile` является вспомогательным классом и поэтому может применяться к любой записи. Класс `radiusObjectProfile` является структурным классом, который используется для создания контейнеров профилей radius, что не является необходимым для работы.

Далее, отредактируйте имеющийся файл users, как было показано в предыдущем примере с использованием PAM, но вместо того чтобы изменить метод по умолчанию на PAM, закомментируйте весь раздел. Этот файл управляет способом аутентификации и авторизации пользователей. Если вы удалите метод по умолчанию, этого окажется достаточно для того, чтобы модуль LDAP смог принять на себя обработку аутентификации и авторизации пользователей.

Файл radiusd.conf требует большего внимания. Раскомментируйте в разделах `authenticate` и `authorize` ключевое слово `ldap`, которое включает аутентификацию и авторизацию LDAP. Также найдите раздел, похожий на `Auth-Type LDAP { ldap }`, и раскомментируйте его. Наконец, раскомментируйте раздел `ldap { ... }` и укажите адрес вашего сервера, имя базового объекта (base DN), а также при желании необязательную информацию об аутентификации. Как и в случае с другими программами, с которыми вы встречались, в процессе начальной привязки выполняется поиск отличительного имени DN пользователя; затем выполняется вторая привязка от имени этого пользователя для подтверждения пароля и получения атрибутов. Таким образом, пользователю, от имени которого выполняется начальная привязка (или анонимному пользователю, если конкретная учетная запись не была настроена), должен иметь разрешение на поиск атрибута `uid`, а все пользователи должны иметь разрешение на чтение своих собственных атрибутов.

Для пользователей, аутентификация которых должна выполняться через LDAP, необходимо использовать класс объекта `radiusProfile` и атрибут `dialupAccess` с каким-нибудь значением, например, "yes". В более сложных конфигурациях вы можете использовать это значение для применения различных параметров, но для основных задач этот атрибут может содержать любое значение.

FreeRADIUS – чрезвычайно надежный RADIUS-сервер, но для выполнения поставленных вами задач может потребоваться значительная настройка. Рассмотренные здесь две настройки сосредоточены только на том, что необходимо для работы LDAP.

## Интеграция CUPS и LDAP

Система печати CUPS (Common UNIX Printing System) на данный момент является наиболее предпочтительной службой печати благодаря простой настройке, поддержке протокола печати через Интернет (Internet Printing Protocol, IPP) и обратной совместимости с традиционными инструментами `lpr`. Служба CUPS поддерживает PAM, но ей необходимо указать, как и когда следует выполнять аутентификацию.

Сначала отредактируйте файл `/etc/pam.d/cups` для обеспечения поддержки LDAP. Затем создайте в файле `/etc/cups/cupsd.conf` контейнер для ваших принтеров, требующий выполнения аутентификации, как показано в листинге 13.

### Листинг 13. Контейнер для принтеров, требующий выполнения аутентификации

```
<Location /printers>
    AuthType Basic
</Location>
```

В листинге 13 показана конфигурация, требующая выполнения аутентификации уровня Basic для любых URL, начинающихся с `/printers`. Конфигурация CUPS почти идентична конфигурации Apache, поэтому она должна напомнить вам листинг 11. Однако вместо встроенного модуля LDAP служба CUPS использует PAM, поэтому в настройках LDAP нет необходимости. Теперь, когда вы попытаетесь перейти по ссылке, начинающейся с `/printers`, что обеспечивает возможность печати на принтере, вам будет предложено ввести пароль. Это показано в листинге 14.

### Листинг 14. Проверка того, что CUPS работает с LDAP

```
[sean@bob LPIC-III_5]$ lpr index.xml
Password for sean on localhost? mypassword
[sean@bob LPIC-III_5]$
```

Если пароль указан неверно или PAM не работает, пользователь получит повторное приглашение на ввод пароля. Тем не менее, в листинге 14 PAM успешно работает, поэтому документ был распечатан и пользователь вернулся в командную строку.

## Интеграция LDAP и Samba

В этом разделе описывается материал по теме 305.4 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 1.

Из этого раздела вы узнаете:

- Как осуществить миграцию из `smbpasswd` в LDAP
- Как понимать схему OpenLDAP Samba
- Как работает LDAP в качестве хранилища паролей Samba

Samba – это способ интеграции с сетями Microsoft Windows, используемый сообществом UNIX. С помощью этого программного обеспечения вы можете обмениваться файлами в сетях Microsoft (как в качестве клиента, так и в качестве сервера) и обеспечивать доступность компьютеров под управлением UNIX для других Windows-клиентов так же, как если бы это

были компьютеры под управлением Windows.

### Понимание аутентификации Samba

Поскольку служба Samba предназначена для интеграции с сетями Windows, она должна использовать механизмы аутентификации, которые используются в Windows. Если вы выполняете аутентификацию на Windows-сервере, это замечательно, однако часто в роли хранилища учетных данных выступает Samba-сервер. Таким образом, необходимы две копии хэшей пароля – одна для обычных паролей UNIX, а другая – для хэшей Microsoft.

Пароли Microsoft похожи на пароли UNIX тем, что в обоих случаях они являются хэшами реального пароля. *Хэш-функция* – это односторонняя функция, которая принимает на вход данные переменной длины (такие как пароль) и формирует на выходе хэш фиксированной длины (строку). Имея хэш, невозможно восстановить оригинальный пароль, хотя для того, чтобы получить данный хэш, потребуется перебрать миллиарды различных паролей.

Для паролей Microsoft хранятся два различных хэша: хэш LANManager и хэш Windows NT. Первый из них менее защищен по сравнению со вторым, поскольку перед хэшированием пароля выполняются несколько действий, призванных снизить количество возможных выходных данных. Хэш Windows NT был разработан для преодоления этих ограничений. Хотя система хранит оба хэша, вы можете отключить поддержку LANManager, если все ваши клиенты поддерживают хэши Windows NT (такая поддержка доступна в OC Windows NT SP3 и более поздних версиях).

Традиционно Samba хранит хэши паролей в файле smbpasswd и использует инструменты наподобие `smbpasswd` для управления этим файлом. Эти хэши можно легко переместить в LDAP так, чтобы несколько Samba-серверов могли выполнять аутентификацию без необходимости использования контроллеров домена и других компонентов инфраструктуры Microsoft. Хранение данных в LDAP также уменьшает дублирование информации в вашей сети.

### Как устроена схема Samba

Пароли NT отличаются от паролей UNIX и не могут храниться в атрибуте `userPassword`. По этой причине схему LDAP нужно расширить для обеспечения поддержки хранения хэшей паролей и другой информации, необходимой клиентам Microsoft.

Дистрибутив Samba содержит файл схемы `samba.schema`. Скопируйте его в каталог `/etc/openldap/schema` и используйте директиву `include` в файле `slapd.conf`, что сделает его частью схемы вашего сервера.

Схема `samba.schema` определяет несколько новых классов объектов, которые перечислены в таблице 4.

**Таблица 4. Классы объектов в схеме `samba.schema`**

Класс объекта ( <code>objectClass</code> )	Описание
<code>sambaSamAccount</code>	Содержит информацию (компьютер, пользователь и так далее), необходимую для учетной записи в среде NT.
<code>sambaGroupMapping</code>	Сопоставляет группу UNIX группе Windows.
<code>sambaTrustPassword</code>	Содержит необходимую для аутентификации информацию о доверительных отношениях между доменами.
<code>sambaDomain</code>	Содержит информацию о домене в дереве LDAP. Вы обнаружите, что один из них будет автоматически добавлен в ваше дерево LDAP после того, как вы

настройте связку Samba/LDAP.

## Настройка Samba для использования с LDAP

Настройка совместной работы Samba и LDAP заключается в следующем: редактирование файла smb.conf с целью настройки источника данных LDAP; управление записями пользователей LDAP с целью использования новых атрибутов Samba.

В файле smb.conf вы найдете строку, похожую на `passdb backend = tdb`, которая определяет механизм хранения файла smbpasswd. Замените эту строку измененным для вашей среды кодом, приведенным в листинге 15.

### Листинг 15. Использование хранилища паролей ldapsam

```
# Необходимо указать для ldapsam uri к серверу LDAP
passdb backend = ldapsam:ldap://192.168.1.138/
# Пользователь на сервере LDAP, которому разрешено выполнять
# чтение и запись новых атрибутов
# Пароль будет указан позже
ldap admin dn = cn=root,dc=ertw,dc=com
# То же, что и база поиска
ldap suffix = dc=ertw,dc=com
# Контейнеры OU для пользователей/компьютеров/групп
ldap user suffix = ou=People
ldap machine suffix = ou=Computers
ldap group suffix = ou=Group
```

По завершении редактирования файла smb.conf перезапустите Samba и выполните команду `smbpasswd -W`. Вам будет предложено ввести пароль учетной записи (DN) администратора LDAP, указанной в smb.conf. С этого момента Samba будет использовать данные LDAP для аутентификации пользователей.

## Управление пользователями Samba в LDAP

Прежде чем пользователи смогут использовать Samba, для них необходимо настроить класс объекта `sambaSamAccount`, что включает в себя настройку хэшей паролей и назначение пользователю идентификатора безопасности (SID). Это легко сделать с помощью утилиты `smbpasswd`, которая обычно используется для добавления пользователей в файл smbpasswd. `smbpasswd` будет управлять пользователями LDAP в том случае, если в файле smb.conf выполнены все необходимые настройки для использования LDAP, как, например, в листинге 15.

Чтобы выполнить настройки для нового пользователя, сначала убедитесь, что его учетная запись содержит класс объекта `posixAccount` и атрибут `uid`, которые уже должны присутствовать, если пользователь подключался через LDAP, работающий совместно с PAM или NSS. После этого выполните команду `smbpasswd -a имя_пользователя`, чтобы выполнить редактирование записи пользователя LDAP, включающее в себя указание пароля Samba. В листинге 16 показана типовая запись пользователя после ее настройки для работы с Samba.

### Листинг 16. Запись пользователя Samba

```
dn: cn=Jim Joe,ou=people,dc=ertw,dc=com
givenName: Jim
```

```
sn: Joe
cn: Jim Joe
uid: jjoe
uidNumber: 1000
sambaSID: S-1-5-21-2287037134-1443008385-640796334-
userPassword:: e01ENX1yTDBZMjB6QytGenQ3MlZQek1TazJBPT0=
sambaLMPassword: 5BFAFBEBFB6A0942AAD3B435B51404EE
sambaNTPassword: AC8E657F83DF82BEEA5D43BDAF7800CC
loginShell: /bin/bash
gidNumber: 4
homeDirectory: /home/a
sambaAcctFlags: [U]
objectClass: inetOrgPerson
objectClass: sambaSamAccount
objectClass: posixAccount
objectClass: top
```

Строки листинга 16, выделенные жирным шрифтом, были добавлены в результате выполнения команды `smbpasswd`. Начиная сверху, для учетной записи добавлен идентификатор безопасности SID. Использование `smbpasswd` освобождает вас от вычисления этого значения, поскольку `smbpasswd` самостоятельно определяет, какой SID использовать. Далее хранятся хэши паролей LanManager и NT. Атрибут `sambaAcctFlags` используется для хранения некоторых атрибутов записи. Возможными значениями этого флага являются:

- **N**: Пароль не требуется
- **D**: Учетная запись отключена
- **H**: Требуется наличие домашней директории
- **T**: Временная копия другой учетной записи
- **U**: Постоянная учетная запись пользователя
- **M**: Учетная запись пользователя для подключения к кластеру MNS (Majority Node Set)
- **W**: Учетная запись доверия рабочей станции
- **S**: Учетная запись доверия сервера
- **L**: Автоматическая блокировка
- **X**: Срок действия пароля не ограничен
- **I**: Учетная запись доверия домена

Наконец, класс объекта `sambaSamAccount` позволяет использовать все эти атрибуты.

В дополнение ко всему вышеизложенному, вы можете настраивать многие другие параметры для указания дополнительной информации, используемой Windows. Узнать о работе с информацией пользователя Samba из командной строки можно из man-руководства `pdbedit`. Samba может выступать в роли основного контроллера домена Windows (PDC), в этом случае дополнительная информация необходима для правильной работы Windows-клиентов.

## Синхронизация паролей

Теперь, когда существуют два набора паролей (`userPassword` и два хэша Samba), необходимо найти способ синхронизации паролей между собой. Если пользователь меняет свой пароль Samba из командной строки или с помощью Windows-клиента, пароль UNIX также должен измениться. И наоборот, если пользователь меняет пароль UNIX, должен измениться пароль Samba.

В первом случае дела обстоят намного проще. Добавьте строку `ldap password sync =`

`yes` в раздел **[global]** файла `smb.conf` и перезапустите Samba. После этого при изменении паролей будут меняться хэши как для `userPassword`, так и для Samba.

Чтобы при изменении пользователями своих паролей посредством UNIX-команды `passwd` также менялись пароли Samba, необходимо использовать PAM. Samba поставляется с модулем `mod_smbpasswd`, который используется для проверки и изменения паролей через систему Samba. В данный момент вам не нужна проверка паролей, поэтому будет использоваться только функция `password`. В листинге 17 показана часть конфигурационного PAM-файла, который в случае его использования изменяет в LDAP как пароль UNIX, так и пароль Samba.

### Листинг 17. Конфигурация PAM для одновременного изменения паролей в UNIX и Samba

```
password      requisite    pam_cracklib.so try_first_pass retry=3
password      optional     pam_smbpass.so use_authtok use_first_pass
password      sufficient   pam_unix.so md5 shadow nullok try_first_pass use_authtok
password      sufficient   pam_ldap.so use_authtok
password      required    pam_deny.so
```

В листинге 17 была добавлена строка, выделенная жирным шрифтом. Модуль `pam_smbpass` указан как необязательный, поэтому, если пользователь не указан в качестве пользователя Samba, этот шаг будет пропущен. Стока, изменяющая пароль Samba, поставлена перед соответствующими строками для UNIX и LDAP, поскольку последние две имеют тип `sufficient`, что означает, что первый введенный успешный пароль прекращает дальнейшую обработку файла.

После применения настроек, показанных в листинге 17, при смене пользователем своего пароля из командной строки также будет изменяться пароль Samba.

### Миграция существующих пользователей в LDAP

Когда вы переходите к использованию LDAP в качестве хранилища данных, вероятно, у вас уже имеются пользователи, чьи пароли хранятся в файлах и должны быть перенесены в LDAP. Утилита `pdbedit` умеет копировать учетные записи из одного места в другое, облегчая эту задачу.

В листинге 18 показано использование утилиты `pdbedit` для выполнения миграции пользователей. Параметр `-i` задает источник данных, а параметр `-e` – их конечное местоположение. Перед запуском команды `pdbedit` у вас должна быть настроена база данных `ldapsam` в файле `smb.conf`.

### Листинг 18. Миграция пользователей из `tdbsam` в `ldapsam`

```
[root@server1 ~]# pdbedit -e ldapsam -i tdbsam
Importing account for fred...ok
Importing account for jsmith...ok
```

Если вы используете хранилище паролей `smbpasswd`, вместо `tdbsam` укажите параметр `smbpasswd`.

## **Интеграция LDAP и Active Directory**

В этом разделе описывается материал по теме 305.5 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 2.

Из этого раздела вы узнаете:

- Об интеграции Kerberos и LDAP
- О межплатформенной аутентификации
- О принципах технологии единого входа (Single sign-on)
- Об ограничениях интеграции и совместимости между OpenLDAP и Active Directory

Операционную систему Microsoft Windows можно встретить практически в любой организации, и существует большая вероятность, что в вашей среде уже используется служба каталога от корпорации Microsoft – Active Directory. Служба Active Directory основана на двух открытых протоколах: LDAP и Kerberos. Если вы изучите эти протоколы и правильно настроите систему Linux, то ваши Linux-устройства смогут выполнять аутентификацию в службе каталога предприятия и поддерживать технологию единого входа *single sign on* (SSO). Это означает, что после того, как вы один раз выполните вход в систему на вашей машине, ваши учетные данные будут действовать во всей сети на протяжении всего сеанса работы.

### **Что такое Kerberos**

Kerberos – это протокол, названный в честь трехглавого пса, охранявшего вход в подземное царство в древнегреческой мифологии. Этот протокол позволяет пользователям и серверам подтверждать свою подлинность в сетях без доверительных отношений. Этот протокол был разработан в Массачусетском технологическом институте (МИТ) для использования в собственной сети и впоследствии распространился по всему миру. Корпорация Microsoft выбрала Kerberos для использования в составе службы Active Directory операционной системы Windows 2000.

Текущей версией протокола Kerberos является Kerberos V, хотя иногда вы можете встретиться и с предыдущей версией – Kerberos IV. Протокол Kerberos V обеспечивает обратную совместимость с системами, использующими Kerberos IV.

### **Протокол Kerberos**

Kerberos – это протокол, который позволяет службам выполнять проверку подлинности пользователя без необходимости предоставления пароля. Это достигается путем установки сервера с двусторонним доверием, называющегося службой аутентификации (Authentication Service, AS). Служба AS использует для каждого пользователя и устройства совместно используемый секретный ключ. Этот ключ используется для защиты информации, передаваемой между AS и другим участником установленного соединения, и даже позволяет службе AS передавать пользователю сообщение (называемое *мандатом*), адресованное кому-то другому. В последнем случае пользователи не могут прочесть мандат, поскольку они не имеют совместно используемого секретного ключа.

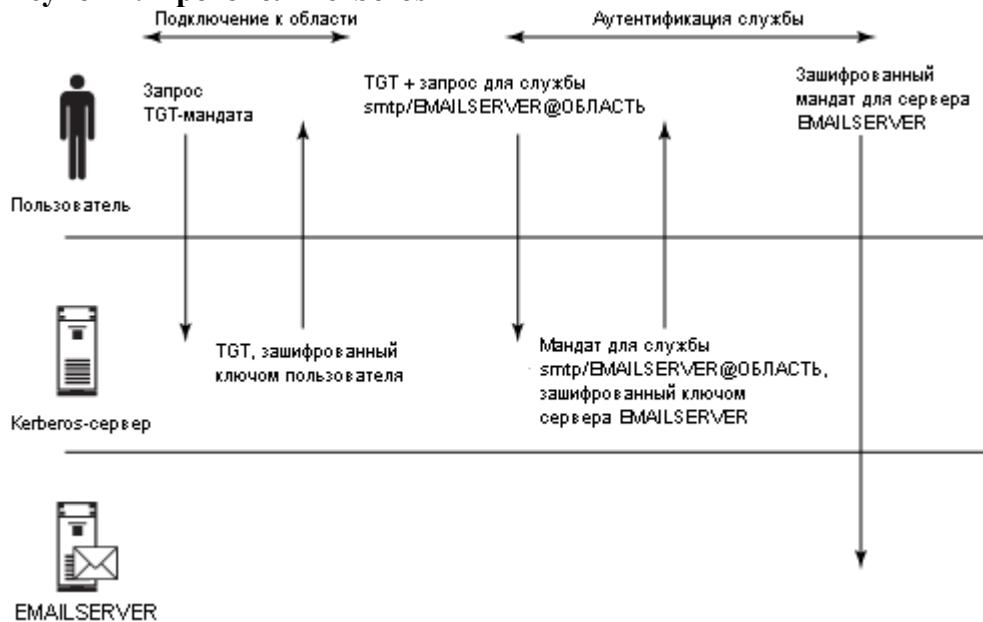
Все клиенты и серверы в совокупности образуют *область Kerberos*, которая похожа на NIS-домен или, в некотором отношении, на отличительно имя базового объекта (base DN) дерева LDAP. Область определяет все устройства и всех людей, которые выполняют аутентификацию на общих Kerberos-серверах. В общем смысле, область – это DNS-зона организации, написанная заглавными буквами, такая как ERTW.COM. В терминах Kerberos клиенты – это компьютеры, которые получают мандаты от Kerberos-сервера, а серверы – это устройства, на которых работают службы Kerberos для выдачи мандатов.

Всякий участник, который аутентифицируется в области Kerberos, имеет идентифицирующий его *принципал* Kerberos и связан с паролем или совместно используемым секретным ключом. Когда пользователь подключается к серверу, на самом

деле он подключается к службе, запущенной на сервере. Каждая служба рассматривается в отдельности и должна быть зарегистрирована в качестве принципала сервером Kerberos. Принципал службы имеет форму **имя\_службы/имя\_сервера@ОБЛАСТЬ**, а принципал пользователя имеет форму **пользователь@ОБЛАСТЬ**.

Протокол Kerberos изображен на рисунке 1.

**Рисунок 1. Протокол Kerberos**



При рассмотрении протокола Kerberos можно разделить его на две отдельных фазы: начальное подключение пользователя к области и аутентификация пользователя в службе. Преимущество Kerberos заключается в том, что начальное подключение выполняется только один раз, а все последующие процедуры аутентификации могут выполняться множество раз на различных серверах.

Первая фаза Kerberos начинается с того, что пользователь запрашивает у Kerberos-сервера (а именно, у компонента, называемого центром распределения ключей – Key Distribution Center, KDC) *мандат на получение мандата* (Ticket Granting Ticket , TGT), который впоследствии используется для обращения к службе. Сервер KDC генерирует мандат TGT, шифрует его с помощью пароля пользователя и отправляет обратно пользователю.

Мандат TGT можно сравнить с гостевым пропуском в организации. Вы предъявляете удостоверение своей личности сотруднику охраны (KDC), после чего вам выдается гостевой пропуск, действующий в течение одного дня. Этот процесс позволяет вам хранить свой собственный идентификатор ID в безопасности, а также ограничивает уязвимость организации в случае похищения гостевого пропуска. Срок действия TGT ограничен коротким отрезком времени, обычно около 8 часов.

На второй фазе пользователь решает, что ему нужен доступ к определенной службе. Он посыпает запрос службе выдачи мандатов (Ticket Granting Service, TGS), являющейся компонентом Kerberos-сервера, который содержит TGT и имя службы (принципал). Служба TGS проверяет, действителен ли мандат TGT, и затем выдает мандат, зашифрованный совместно используемым секретным ключом службы. Наконец, пользователь предоставляет мандат службе. Если служба может успешно расшифровать мандат, она понимает, что запрос был санкционирован системой Kerberos. При этом никакие пароли по сети не передаются.

Kerberos является преградой для атак повторного использования (когда атакующий захватывает мандат и использует его повторно), устанавливая ограниченное время жизни для

мандатов и включая в зашифрованный мандат метки времени. Выданный для службы мандат может действовать в течение 5 минут, поэтому для того, чтобы узнать, что мандат был использован повторно, служба должна помнить только о мандатах, полученных за последние 5 минут. Для успешной работы этой функции необходимо, чтобы часы на всех компьютерах были синхронизированы.

### **Куда можно приспособить LDAP?**

Kerberos предоставляет вам только инфраструктуру аутентификации, подобно тому, как это делает PAM. Пользовательская информация в базе данных Kerberos не хранится.

Секретные ключи Kerberos могут храниться в базе данных LDAP или отдельно. Выбор зависит от реализации Kerberos. В обоих случаях LDAP используется для хранения пользовательской информации, такой как домашняя директория и личные данные.

Независимо от того, где хранится база данных Kerberos, вы должны обеспечивать ее надежную защиту. Ключи Kerberos подобны паролям – они могут быть похищены и использованы для генерации TGT и мандатов. В большинстве руководств настоятельно рекомендуется использовать для Kerberos-сервера отдельный компьютер и защищать его как можно лучше.

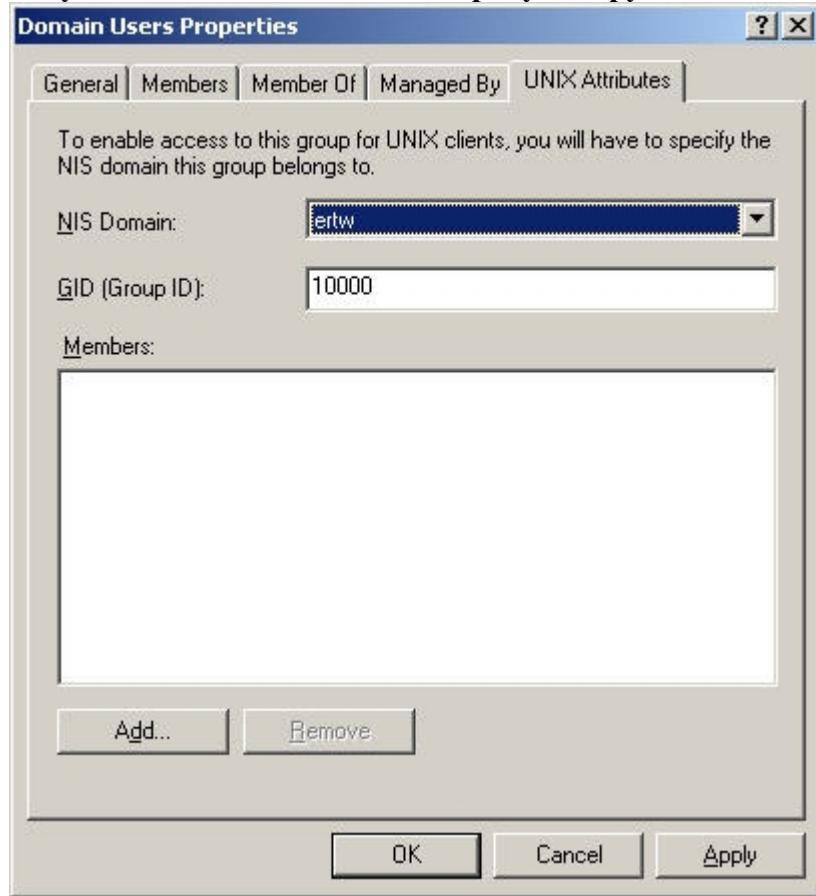
### **Настройка Microsoft Active Directory для гостевых подключений Linux**

Реализация протоколов Kerberos и LDAP в Active Directory совместима с их реализацией в Linux. Microsoft расширила протокол Kerberos для обеспечения поддержки специфичных для Windows атрибутов, но не запрещает использовать этот протокол пользователям UNIX (обратитесь к разделу [Ресурсы](#) за документацией Microsoft, относящейся к этому вопросу).

Для поддержки некоторых атрибутов UNIX необходимо расширить схему Active Directory, что в ОС Windows 2003 Server делается легко. Откройте панель управления на вашем контроллере домена и откройте оснастку **Add or Remove Programs > Add/Remove Windows Components**. Найдите компонент Active Directory Services и в его составе выберите подкомпонент Identity Management for UNIX (если вы используете более раннюю версию Windows, этот компонент иногда называется Server for NIS). Установите это программное обеспечение, и схема LDAP будет расширена; в диалоговых окнах пользователей появится вкладка UNIX Attributes, которую мы скоро будем использовать.

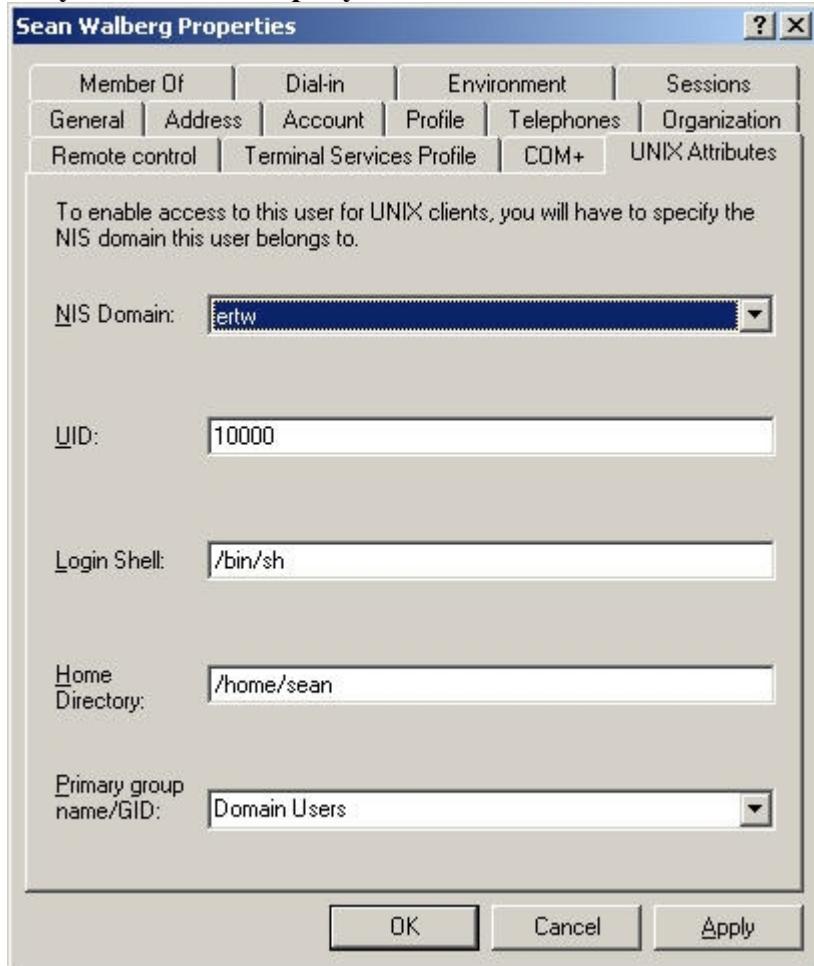
В оснастке Active Directory Users and Computers отредактируйте свойства группы безопасности Domain Users. Обратите внимание на вкладку **UNIX Attributes**. Назначьте группу и NIS-домен группе Domain Users, как показано на рисунке 2. Это сделает группу видимой для UNIX-систем. Данная группа будет являться для пользователя основной.

**Рисунок 2. Назначение UNIX-атрибутов группе**



В контейнере Users найдите пользователя, учетную запись которого вы хотите использовать на ваших UNIX-серверах. В свойствах учетной записи этого пользователя перейдите на вкладку **UNIX Attributes** и назначьте ему стандартные UNIX-атрибуты, как показано на рисунке 3.

Рисунок 3. UNIX-атрибуты пользователя



На рисунке 3 пользователю были назначены основная группа, домашняя директория, командная оболочка и идентификатор userid.

Дальше вы должны создать сервисную учетную запись, разрешающую доступ к вашему дереву LDAP, поскольку анонимный доступ по умолчанию отключен. Используйте для этого пользователя следующую конфигурацию:

- **Name** (Имя): учетное имя сервиса LDAP (или по вашему усмотрению)
- **User logon name** (Имя входа в систему): ldap (или по вашему усмотрению)
- **Password** (Пароль): по вашему усмотрению
- **User can't change password** (Запретить смену пароля пользователем): установлено
- **Password never expires** (Срок действия пароля не ограничен): установлено
- **Primary group** (Основная группа): Domain Guests

Сервисная учетная запись должна являться участником только группы Domain Guests (Гости домена). На вкладке **Member Of** учетной записи добавьте группу Domain Guests, выделите ее в списке групп и нажмите кнопку **Set Primary Group**. После изменения основной группы вы можете удалить группу Domain Users из профиля.

Если ваша политика безопасности запрещает изменять параметры паролей, вам придется выполнять дополнительные настройки Linux (они будут описаны ниже) каждый раз, когда меняется пароль. Обратите внимание на то, что LDAP используется только в качестве хранилища информации каталога, а не паролей, поэтому требования к изменению паролей снижаются.

## Настройка Linux

Настройка на стороне Linux состоит из трех этапов. Сначала вы настраиваете доступ к каталогу через файл /etc/ldap.conf. Затем вы настраиваете PAM для Kerberos-аутентификации. Наконец, вы настраиваете Samba на использование информации Active Directory для аутентификации и присоединяете ее к домену.

Прежде чем начать, вы должны убедиться, что ваша Linux-система использует службу DNS и службу сетевого времени, запущенные на сервере Microsoft. Также в DNS-зоне Microsoft для вашего домена должна содержаться хост-запись о Linux-сервере.

### Настройка LDAP

Настройка LDAP выполняется так же, как и [раньше](#), за исключением того, что необходимо задать некоторые сопоставления между атрибутами UNIX и атрибутами Microsoft. В листинге 19 показан файл конфигурации /etc/ldap.conf, который обеспечивает доступ к LDAP-каталогу Microsoft с использованием настроенной ранее учетной записи.

#### Листинг 19. Настройка файла на использование каталога Microsoft

```
# Информация о каталоге
uri ldap://192.168.1.151
binddn ldap@ertw.com
bindpw ldap
ssl no
base dc=ertw,dc=com

# Сопоставление атрибутов
nss_map_objectclass posixAccount user
nss_map_objectclass shadowAccount user
nss_map_attribute uid sAMAccountName
nss_map_attribute homeDirectory unixHomeDirectory
nss_map_attribute shadowLastChange pwdLastSet
nss_map_objectclass posixGroup group
nss_map_attribute uniqueMember member
pam_login_attribute sAMAccountName
pam_filter objectclass=User
pam_password ad
```

Конфигурация, показанная в листинге 19, указывает модулю на сервер Microsoft LDAP и на использование учетных данных, настроенных ранее. Имена атрибутов UNIX сопоставлены соответствующим именам атрибутов Microsoft, например, `sAMAccountName` и `userid`.

В заключение, добавьте строку `ldap winbind` в разделы `passwd`, `group` и `shadow` файла /etc/nsswitchconf (оставьте в нем строку `files`). Это позволит вашей системе получать информацию каталога из LDAP и Samba (Samba будет настроена позже).

После выполнения этих шагов вы можете запустить команду `getent passwd`, чтобы увидеть пользователей LDAP. Помните, что для того чтобы пользователь отображался в списке, для него должны быть настроены атрибуты UNIX в службе Active Directory.

### Настройка Kerberos

Kerberos настраивается через PAM и файл /etc krb5.conf. Если для вашей DNS-зоны вы используете DNS-сервер Microsoft, то вам нужно только указать область, поскольку имя сервера будет автоматически получено от службы DNS. В листинге 20 показано содержимое файла /etc/krb5.conf.

## Листинг 20. Файл /etc/krb5.conf для области ERTW.COM

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
default_realm = ERTW.COM
dns_lookup_realm = false
dns_lookup_kdc = true
ticket_lifetime = 24h
forwardable = yes

[realms]
ERTW.COM = {
    default_domain = ertw.com
}

[domain_realm]
.ertw.com = ERTW.COM
ertw.com = ERTW.COM

[appdefaults]
pam = {
    debug = false
    ticket_lifetime = 36000
    renew_lifetime = 36000
    forwardable = true
    krb4_convert = false
}
```

Файл krb5.conf поделен на разделы, имя каждого из которых заключено в квадратные скобки. В разделе logging указываются пути к различным журналам регистрации событий. В разделе libdefaults настраиваются библиотеки Kerberos; в частности, параметр dns\_lookup\_kdc говорит библиотеке о том, что для поиска центра распределения ключей (KDC) она должна просматривать SRV-записи в службе DNS. Эта запись выглядит примерно как `_kerberos._tcp.ERTWCOM.` и содержит имя сервера и номер порта, через который осуществляется доступ.

В разделе realms определяются области и связанные с ними зоны DNS. В разделе domain\_realm выполняется обратное сопоставление: хост определяет свою область на основе своего полного доменного имени (Fully qualified domain name, FQDN). Наконец, в разделе appdefaults перечисляются приложения, использующие Kerberos; в данном случае был настроен модуль PAM с некоторыми параметрами по умолчанию.

На практике файл krb5.conf почти не требует редактирования, поскольку конфигурация по умолчанию содержит все необходимые элементы. Все, что вам необходимо сделать - это указать имена вашей области и вашего домена. Также вы можете использовать системную утилиту настройки Kerberos, такую как `authconfig`.

Конфигурация PAM полностью аналогично предыдущим конфигурациям LDAP и smbpasswd. Когда это необходимо, вы обращаетесь к PAM-библиотеке Kerberos. В листинге 21 показан фрагмент файла system-auth операционной системы Fedora после настройки Kerberos.

## Листинг 21. Файл system-auth после настройки Kerberos

```
auth      sufficient  pam_unix.so nullok try_first_pass
auth      sufficient  pam_krb5.so use_first_pass
auth      required    pam_deny.so

account   required    pam_unix.so broken_shadow
account   [default=bad success=ok user_unknown=ignore] pam_krb5.so
account   required    pam_permit.so

password  sufficient  pam_unix.so md5 shadow nullok try_first_pass use_authok
password  sufficient  pam_krb5.so use_authok
password  required    pam_deny.so

session   required    pam_unix.so
session   optional   pam_krb5.so
```

Проверка Kerberos добавлена сразу же после проверки пароля UNIX на стадии авторизации (auth) и определена как достаточная. Это означает, что если пароль UNIX обнаружен, обращения к Kerberos не происходит. Если пароль UNIX не обнаружен, выполняется обращение к Kerberos. Если попытка обращения к Kerberos оканчивается неудачей, весь ряд проверок также завершается неудачей. Если в результате всех проверок пользователь не был найден, управление передается модулю `pam_deny`, который возвращает статус неудачи.

На стадии account используется синтаксис, альтернативный тому, который вы видели до этого момента. Каждый модуль PAM может возвращать различные значения, такие как "success" или "no such user". Квадратные скобки позволяют администратору выполнять различные действия в зависимости от возвращенного кода. В листинге 21 реализована политика, которая говорит, что если `pam_krb5` возвращает успешный результат, то обработка продолжается. Если пользователь неизвестен, модуль полностью игнорируется. Если в результате попытки аутентификации возвращается какое-либо другое значение, она считается неудачной. Такое поведение близко к использованию ключевого слова `required` за исключением того, что попытка, в результате которой был обнаружен неизвестный пользователь, не считается неудачной. Для получения дополнительной информации о синтаксисе и параметрах обратитесь к man-руководству `pam.conf(5)`.

Стадии проверки password и session содержат модуль без каких-либо специальных параметров.

На этом этапе вы должны иметь возможность подключаться к вашему Linux-серверу, используя учетные данные из Active Directory. Далее мы настроим Samba и усилим защиту подключений, создав учетную запись компьютера для сервера.

## Настройка Samba и присоединение к домену

Для настройки Samba сначала необходимо удалить существующее хранилище паролей из файла smb.conf, а также все файлы `tdb` в директориях `/etc/samba` и `/var/cache/samba`. В листинге 22 показаны директивы, которые вы должны добавить в раздел `[global]` файла smb.conf, чтобы позволить Samba использовать Active Directory.

## Листинг 22. Настройка Samba для интеграции с AD

```
# Безопасность Active Directory
security = ads
realm = ERTW.COM
```

```

use kerberos keytab = yes

# Сопоставление идентификаторов
idmap backend = ad
ldap idmap suffix = dc=ertw,dc=com

# Настройка LDAP
ldap admindn = cn=ldap,cn=users,dc=ertw,dc=com
ldap suffix = dc=ertw,dc=com

# Winbind
winbind use default domain = yes
winbind nested groups = yes

```

Листинг 22 начинается с указания того, что используется режим защиты ADS (удаленный сервер Active Directory), а также область Kerberos ERTW.COM. Вторая группа параметров настраивает idmapping – функцию сопоставления удаленных идентификаторов безопасности Microsoft (SID) локальным идентификаторам UNIX (id). Эта конфигурация определяет Active Directory в качестве источника данных. Сопоставление идентификаторов уже выполнено на стороне Microsoft, поскольку вы уже ввели идентификаторы ID пользователей и групп на соответствующих вкладках UNIX Attributes. Вашему серверу необходимо просто получить информацию из LDAP.

Конфигурация LDAP вам уже знакома; в ней указывается имя DN созданного ранее пользователя для подключения к LDAP. Заметьте, что вместо контейнера `ou=people`, который использовался до сих пор, сейчас используется контейнер `cn=users`. Пароль указывается через команду `smbpasswd`.

Последние две строки разрешают использование службы Winbind, являющейся реализацией некоторых RPC-запросов Microsoft (для получения дополнительной информации о Winbind обратитесь к разделу [Ресурсы](#)). Это позволяет вам получать больше информации от вашего сервера Active Directory, а не только списки групп и пользователей, для которых вы добавили атрибуты UNIX.

По завершении редактирования файла `smb.conf` запустите службы Samba и `winbind`.

Последние шаги по настройке Samba заключаются в задании пароля `admindn` и присоединении к вашему домену. В листинге 23 показано присоединение компьютера к домену.

### **Листинг 23. Задание пароля `admindn` и присоединение к домену**

```

[root@server1 ~]# smbpasswd -W
Setting stored password for "cn=ldap,cn=users,dc=ertw,dc=com" in secrets.tdb
New SMB password: ldap
Retype new SMB password: ldap

[root@server1 ~]# net ads join -U administrator
administrator's password: mypassword
Using short domain name -- ERTW0
Joined 'SERVER1' to realm 'ERTW.COM'

```

## **Проверка**

У вас должна появиться возможность подключаться к вашему серверу, используя учетные данные Active Directory, и просматривать общие файловые ресурсы с удаленного компьютера без необходимости ввода учетных данных. Ниже перечислены некоторые полезные команды, которые вы можете использовать для проверки результатов:

- `net ads testjoin`: выполняет проверку учетной записи компьютера
- `wbinfo -u`: выводит список пользователей Active Directory и проверяет `winbind`
- `klist`: после подключения через Kerberos показывает ваш мандат TGT и выданные мандаты всех служб
- `smbclient -k -L '\\\$ERVERNAME'`: выводит список общих файловых ресурсов сервера с именем SERVERNAME, используя регистрацию Kerberos

## **Интеграция LDAP с почтовыми службами**

В этом разделе описывается материал по теме 305.6 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 1.

Из этого раздела вы узнаете, как:

- Планировать структуру схемы LDAP для почтовых служб
- Создавать почтовые атрибуты в LDAP
- Интегрировать Postfix и LDAP
- Интегрировать sendmail и LDAP

Postfix и sendmail – это два наиболее популярных и широко используемых агента пересылки почтовых сообщений (mail transport agents, MTA). Задачей MTA является получение сообщений и отправка их конечному пользователю или следующему узлу MTA. Кроме того, агенты MTA получают сообщения от пользователей и выполняют поиск удаленного агента MTA, способного доставить сообщение.

Работа как sendmail, так и Postfix основана на использовании разных таблиц сопоставлений ключ/значение, которые обычно хранятся в плоских файлах или хэшированных базах данных, подобных BDB. Данный тип поиска хорошо подходит для использования с LDAP. Преимущества LDAP заключаются в том, что множество хостов могут использовать одну и ту же общую конфигурацию; при этом проще разрабатывать инструменты для управления данными, хранящимися в LDAP, а не в плоских файлах, которые впоследствии должны быть перестроены в хэш-таблицы. Издержки LDAP по сравнению с операциями чтения диска не должны играть заметной роли, особенно если дерево LDAP правильно проиндексировано.

## **Настройка sendmail**

Агент пересылки сообщений sendmail является достаточно сложной программой, и добавление LDAP в эту связку только усложняет ситуацию. С помощью sendmail можно сделать практически все что угодно, поскольку возможности конфигурирования этого почтового агента почти безграничны. Недостаток же его заключается в том, что простые вещи на самом деле оказываются более сложными, чем это необходимо.

Отметим, что в sendmail для обработки почты используется интерпретируемый язык, часто называемый *cf*. Cf – это язык, упрощающий синтаксический анализ для программы sendmail, но не для человека. К счастью, для генерирования результирующего кода cf люди могут использовать язык M4, синтаксис которого гораздо проще.

## **Таблицы сопоставлений sendmail**

Во многих операциях языка cf используется поиск информации в *таблицах сопоставлений*

(maps), являющихся наборами пар ключ/значение. Каждая таблица существует с определенной целью, например, таблица `aliases` используется для почтовых псевдонимов, а таблица `mailertable` – для статической маршрутизации электронной почты. Таблица сопоставлений – это объект, состоящий из двух столбцов; ключи поиска содержатся в левом столбце, а соответствующие значения – в правом.

Принцип таблицы сопоставлений не переносится в LDAP напрямую. В результате запроса `sendmail`, выполняемого в таблице по единственному ключу, может быть возвращена только одна запись из правого столбца (правый столбец может содержать несколько записей, но может существовать всего один экземпляр ключа). Для работы с этим `sendmail` определяет схему, позволяющую хранить пары ключ/значение в LDAP. Кроме того, `sendmail` преобразует каждый запрос к карте сопоставлений в фильтр запроса LDAP; это сделано для того, чтобы из одной записи можно было возвращать набор атрибутов. Для работы с данными вашего дерева каталога вы можете либо использовать схему `sendmail`, либо изменить фильтры.

Чтобы начать работу, добавьте схему `misc.schema` (поставляется в составе OpenLDAP) к схеме вашего сервера. В результате этих действий будет реализована почтовая маршрутизация на основе LDAP. После этого добавьте схему `sendmail.schema` из дистрибутива `sendmail`, которая позволит вам хранить таблицы сопоставлений в LDAP.

## Настройка почтовой маршрутизации LDAP

Для обслуживания всех своих пользователей некоторые организации имеют несколько почтовых серверов, что связано с их географическим месторасположением или возможностями управляемости. В этом случае почтовый ящик пользователя может располагаться на сервере `wpgertw.com`, хотя адресом электронной почты этого пользователя может являться `sean@ertw.com`. Почтовая маршрутизация LDAP позволяет любому серверу `sendmail` получать сообщение, выполнять поиск в LDAP и изменять имя почтового адреса на его внутреннее имя. После этого несложно изменить сервер назначения, изменив LDAP. В любом случае, адрес электронной почты пользователя остается неизменным.

`misc.schema` реализует Internet-схему для LDAP-маршрутизации. Схема содержит класс объекта `inetLocalMailRecipient` со следующими атрибутами:

- `mailLocalAddress` – задает адрес электронной почты в том виде, в котором он виден за пределами организации
- `mailRoutingAddress` – задает внутренний адрес пользователя, обычно содержащий имя сервера, на котором расположен почтовый ящик
- `mailHost` – определяет имя сервера, который обрабатывает электронную почту этого пользователя

Информация о хосте пользователя может быть указана как в атрибуте `mailRoutingAddress`, так и в атрибуте `mailHost`. Например, если значением атрибута `mailRoutingAddress` (адрес маршрутизации) является `sean@wpg.ertw.com`, а значением атрибута `mailHost` (имя хоста) – `mx.ertw.com`, возникает противоречие. Если задано имя хоста, почта будет доставляться на него независимо от адреса маршрутизации. Если атрибут `mailRoutingAddress` существует, то его значение будет использоваться в качестве адреса электронной почты. В данном примере почтовый адрес пользователя будет изменен на `sean@wpg.ertw.com`, но сообщение будет доставлено на сервер `mx.ertw.com`.

В листинге 24 приведен код M4, включающий маршрутизацию LDAP. Этот код необходимо поместить в файл `/etc/mail/sendmail.mc`, а затем вы должны пересобрать ваш файл `sendmail.cf`. Обычно это означает, что вам нужно зайти в каталог `/etc/mail/` и запустить команду `make`, или выполнить команду `m4 sendmail.mc > sendmail.cf`.

#### Листинг 24. Включение маршрутизации LDAP в файле sendmail.mc

```
define(`confLDAP_DEFAULT_SPEC', ` -h localhost -b dc=ertw,dc=com')
FEATURE(`ldap_routing')
LDAPROUTE_DOMAIN(`ertw.com')
```

Первая строка устанавливает аргументы по умолчанию для встроенного клиента LDAP – хост и базу поиска. Вторая строка включает функцию маршрутизации LDAP, а третья – включает домен ertw.com в схему маршрутизации.

Наиболее пристального внимания заслуживает дублирование атрибутов `mailLocalAddress` класса `inetLocalMailRecipient` и `mail` класса `inetOrgPerson`. Программа sendmail позволяет вам переопределять используемые ей внутренние поисковые запросы путем передачи дополнительных аргументов в функцию `ldap_routing`. Первый аргумент является фильтром, используемым для нахождения атрибута `mailHost`, а второй аргумент используется для поиска атрибута `mailRoutingAddress`. Таким образом, строка `FEATURE(`ldap_routing', `ldap -1 -T<TMPF> -v mailHost -k (&(objectClass=inetLocalMailRecipient)(mail=%0))', `ldap -1 -T<TMPF> -v mailRoutingAddress -k (&(objectClass=inetLocalMailRecipient)(mail=%0))')` обеспечивает то, что sendmail использует атрибут `mail` вместо атрибута `mailLocalAddress`. Фильтр поиска указан с помощью параметра `-k`, а возвращаемый атрибут – с помощью параметра `-V`. Остальные аргументы – это стандартные аргументы sendmail.

#### Настройка псевдонимов

Псевдонимы LDAP реализованы в sendmail в качестве набора записей в дереве LDAP, включенных в атрибут `sendmailMTAKey` класса `sendmailMTAAliasObject`. Вы можете хранить псевдонимы в их собственном контейнере. В листинге 25 показан LDIF-файл для псевдонима sendmail, который берет почту для адреса `exec@ertw.com` и отправляет ее на адреса `hair@ertwcom` и `teeth@ertw.com`.

#### Листинг 25. Псевдоним для `exec@ertw.com`

```
dn: sendmailMTAKey=execs,ou=aliases,dc=ertw,dc=com
objectClass: sendmailMTAAliasObject
sendmailMTACluster: external
sendmailMTAAliasGrouping: aliases
sendmailMTAKey: execs
sendmailMTAAliasValue: hair@ertwcom
sendmailMTAAliasValue: teeth@ertw.com
```

Первый атрибут, `sendmailMTACluster`, определяет список серверов, которые могут использовать этот псевдоним. Также вы должны указать имя кластера в файле `sendmail.mc`, например, `define(`confLDAP_CLUSTER', `external')`. Этот кластер используется как часть фильтра поиска, поэтому если вы забудете указать его, ваши псевдонимы никогда не будут использоваться. Вместо указания кластера вы можете задать значение для атрибута `sendmailMTAHost`, который применяет запись только к определенному хосту.

Атрибут `sendmailMTAAliasGrouping` должен иметь значение `aliases` – это часть фильтра поиска. Ключ (`sendmailMTAKey`) относится к имени псевдонима; наконец, одно или несколько значений используются в качестве конечных адресов.

Заключительным этапом настройки sendmail является настройка LDAP на использование файла псевдонимов с помощью директивы языка M4 `define(`ALIAS_FILE', `ldap:')`. Вообще в любом месте sendmail.mc, в котором вас просят указать файл, вы можете написать `ldap:`, и сопоставление будет выполняться через LDAP. После этого значение атрибута `sendmailMTAAliasGrouping` становится именем таблицы сопоставлений.

## Настройка Postfix

Программа Postfix менее сложна в настройке, чем sendmail, и при этом совместима с ней. Здесь также используется принцип таблиц сопоставлений, но вместо интеграции таблиц в схему вы должны определить свои собственные фильтры запросов, использующие ваши собственные атрибуты.

Для большинства таблиц в качестве источника информации указывается `ldap:/путь/к/config.cf`, где файл config.cf является конфигурационным файлом, содержащим параметры сервера LDAP, запрос и атрибуты, формирующие результат. Например, директива `local_recipient_maps` задает способ сопоставления адресов электронной почты локальным учетным записям. Задайте параметр `local_recipient_maps = $aliases`, `ldap:/etc/postfix/localrecipients.cf`, чтобы система сначала проверяла базу данных псевдонимов (`aliases`), а затем искала обычный адрес, указанный в записи пользователя. В листинге 26 показано содержимое файла localrecipients.cf.

### Листинг 26. Поиск локальных получателей в LDAP

```
# Информация о сервере LDAP
server_host = ldap://localhost
search_base = ou=people,dc=ertw,dc=com

# %s – это адрес электронной почты...
query_filter = mail=%s

# uid определяет учетную запись получателя
result_attribute = uid
```

В листинге 26 определен локальный сервер LDAP и организационное подразделение People. Postfix сверяется с фильтром поиска и замещает символ `%S` адресом электронной почты. Таким образом, адрес электронной почты пользователя `fred@ertw.com` отразится в поисковом запросе как запись (`mail=fred@ertw.com`) в контейнере People. Атрибут `uid` используется для определения почтового ящика. Чтобы выполнить проверку, вы можете запустить команду `postmap -q fred@ertw.com` `ldap:/etc/postfix/localrecipients.cf`, которая обрабатывает указанный адрес электронной почты с помощью конфигурационного файла localrecipients.cf (обратите внимание, что для получения деталей об учетной записи пользователя fred необходимо настроить NSS).

## Заключение

Из этого руководства вы узнали, как интегрировать LDAP с другими используемыми вами системами. NSS предоставляет простой способ использования LDAP для базовых служб UNIX, перенаправляя стандартные библиотечные вызовы языка C в выбранное вами хранилище данных. PAM – это еще одна абстракция, позволяющая вам выбирать способы

детальной аутентификации, используемой приложениями, ориентированными на работу с PAM. Также PAM может использоваться для ограничения учетных записей и изменения паролей. Файлы PAM располагаются в директории /etc/pam.d.

При миграции с NIS на LDAP необходимо запланировать, какие базы данных должны быть перемещены, а затем с помощью специальных инструментов извлечь данные и конвертировать их в LDIF-формат. На тот случай, если в вашей среде все еще необходимо поддерживать NIS, компания PADL разработала NIS-сервер `ypldapd`, который транслирует обращения к NIS и LDAP, предоставляя приложениям NIS-интерфейс и считывая данные из LDAP.

Многие приложения ориентированы на работу с PAM; в результате для того, чтобы выполнить миграцию на LDAP, достаточно изменить несколько файлов в директории /etc/pam.d. Некоторые приложения, такие как Apache, общаются с LDAP напрямую. Настройка Apache для работы с LDAP включает в себя использование модуля `mod_authnz_ldap` и задание фильтров поиска, которые помогают Apache искать пользователей в дереве LDAP.

С помощью Samba вы можете использовать службы Windows на платформе UNIX. Для прямого общения с Windows вы можете настроить Samba на использование информации протоколов LDAP и даже Kerberos. В последнем случае LDAP будет продолжать использоваться для хранения информации каталога, а Kerberos – для выполнения аутентификации.

Системы электронной почты естественным образом сочетаются с LDAP, поскольку LDAP схож с телефонной книгой. С LDAP может взаимодействовать как sendmail, так и Postfix.

Это руководство завершает рассмотрение служб каталога в рамках экзамена LPIC 3.

Следующее и заключительное руководство этой серии будет сфокусировано на рассмотрении мониторинга и прогнозирования производительности серверов под управлением Linux.

## Ресурсы

### Научиться

- Оригинал руководства "[LPI exam 301 prep, Topic 305: Integration and migration](#)" (EN).
- Изучите предыдущее руководство в серии 301 - "[Подготовка к экзамену LPI 301: Тема 304. Использование](#)" (EN) (developerWorks, март 2008) или [все руководства в серии 301](#).
- Чтобы познакомиться с основами Linux и подготовиться к сертификации в качестве системного администратора, ознакомьтесь со всей [серий руководств для подготовки к экзаменам LPI](#).
- В [программе LPIC](#) (EN) вы можете найти перечни заданий, примеры вопросов и подробные цели для трех уровней сертификации системных администраторов Linux института Linux Professional Institute.
- Если вы не знакомы с Kerberos, начните знакомство с [объяснений в игровой форме](#) (EN), ознакомьтесь с [часто задаваемыми вопросами по Kerberos](#) (EN).
- Прочтите документацию Microsoft по [работе с Kerberos](#) (EN) и по [устранению неполадок Kerberos](#) (EN).
- Презентация PowerPoint о том, [как работает Kerberos](#) (EN), содержит анимированные кадры о движении пакетов и сообщений. Презентация рассчитана на администраторов Windows, но если вы опустите детали, специфичные для Windows, то получите

замечательное описание этого протокола.

- Руководства [Использование Samba и Kerberos](#) (EN) и [Встроенный LDAP, встроенный Kerberos, а также службы и схема Windows Server AD для управления межплатформенной проверкой подлинности](#) (EN) содержат пошаговые инструкции по интеграции служб Linux в Active Directory.
- Документация Samba, представленная разделом [IDMAP](#) (EN), показывает, как Samba выполняет сопоставления идентификаторов безопасности Microsoft (SIDs) и идентификаторов UNIX (userids).
- Статья о [Winbind](#) (EN) раскрывает альтернативный способ интеграции Samba и AD без использования Kerberos.
- Описание модуля [mod\\_authnz\\_ldap](#) (EN) содержит все детали, касающиеся настройки Apache и LDAP.
- О том, [как работает аутентификация HTTP](#) (EN), рассказывает Wikipedia.
- Прочитайте документацию FreeRADIUS по [модулю LDAP](#) (EN). Если все ваши попытки отыскать схему не увенчались успехом, попробуйте использовать [RADIUS-LDAPv3.schema.gz](#).
- Перед началом работы с Postfix и LDAP прочитайте [обучающее руководство Postfix LDAP](#) (EN) и справочное man-руководство [ldap\\_table\(5\)](#).
- В [руководстве по конфигурации sendmail](#) (EN) описаны все способы использования LDAP, включая способы генерации поисковых фильтров из конфигурационного файла. Эта [публикация об альтернативном способе построения псевдонимов sendmail](#) (EN) многое разъясняет не только потому, что описанный способ проще, но также потому, что он дает заглянуть за кулисы того, что происходит.
- Очень советую онлайновую книгу [LDAP для больших ученых](#) (EN), несмотря на то, что работа над ней ещё не закончена.
- В [разделе Linux сайта developerWorks](#) можно найти дополнительные ресурсы для разработчиков Linux, а также [самые популярные среди наших читателей статьи и руководства](#) (EN).
- Посмотрите все [советы по Linux](#) и [руководства Linux](#) на сайте developerWorks.
- Следите на последними новостями на портале [Web-трансляций и технических мероприятий developerWorks](#) (EN).

## Получить продукты и технологии

- Утилита [Firewall Builder](#) упрощает задачу создания правил iptables, предоставляя в ваше распоряжение графический интерфейс и набор инструментов для развертывания обновлений на ваших брандмауэрах.
- Загрузите [pam\\_ldap](#) и [nss\\_ldap](#), если ваш дистрибутив не содержит LDAP-библиотек PAM и NSS компании PADL.
- Загрузите [ypldapd](#), если вы планируете выполнить демонстрационные примеры по настройке шлюза NIS-LDAP. Лицензия действительна в течение 30 дней.
- Загрузите [инструменты миграции LDAP](#) с Web-сайта компании PADL.
- Корпорация Microsoft разработала утилиту [gssMonger](#) для проверки совместимости Kerberos-аутентификации между Windows и другими платформами.
- Загрузите [OpenLDAP](#).

- [phpLDAPadmin](#) - инструмент администрирования LDAP на базе Web. Если вам больше нравится графический интерфейс, вам стоит посмотреть на [Luma](#) (EN).
- Используйте в своем следующем проекте разработки для Linux [ознакомительные версии программного обеспечения IBM](#), которые можно скачать непосредственно с developerWorks (EN).

# Подготовка к экзамену LPI 301: Тема 306.

## Планирование пропускной способности

*Профессионал Linux высокого уровня (LPIC-3)*

Шон Уолберг, старший сетевой инженер, независимый писатель

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux Professional Institute на квалификацию профессионала Linux® высокого уровня (LPIC-3). В этом руководстве, последнем из серии из шести руководств, Шон расскажет о мониторинге ресурсов вашей системы, об устранении проблем, связанных с использованием системных ресурсов, а также о том, как оценивать пропускную способность системы.

[Больше статей из этой серии](#)

**Дата:** 24.03.2009

**Уровень сложности:** средний

### Предисловие

Узнайте, чему могут научить вас эти руководства, и как получить от них больше пользы.

### Об этой серии руководств

Институт Linux Professional Institute (LPI) сертифицирует системных администраторов Linux® по трём уровням: *младший уровень* (также называемый "уровень сертификации 1"), *углубленный уровень* (также называемый "уровень сертификации 2") и *высший уровень* (также называемый "уровень сертификации 3"). Чтобы получить сертификацию на уровне 1, нужно сдать экзамены 101 и 102. Чтобы получить сертификацию на уровне 2, нужно сдать экзамены 201 и 202. Чтобы получить сертификацию на уровне 3, у вас должна быть действующая сертификация на углубленном уровне и сдан экзамен 301 ("основной"). Кроме того, на высоком уровне от вас может потребоваться сдача дополнительных экзаменов.

Сайт developerWorks предлагает руководства, которые помогут вам подготовиться к пяти экзаменам для младшего, углубленного и высшего уровня. В каждом экзамене охватывается несколько тем, и для каждой темы на developerWorks есть соответствующий учебник для самостоятельного изучения. В таблице 1 перечислены шесть тем и соответствующие им руководства developerWorks для экзамена LPI 301.

**Таблица 1. Экзамен LPI 301: руководства и темы**

Тема экзамена	Руководство developerWorks	Краткое описание руководства
Тема 301	<a href="#">Подготовка к экзамену LPI 301: понятия, архитектура и модель (EN)</a>	Узнайте о понятиях и архитектуре LDAP, о том, как проектировать и внедрять каталог LDAP, а также о схемах.
Тема 302	<a href="#">Подготовка к экзамену LPI 301: установка и разработка</a>	Узнайте, как устанавливать, настраивать и использовать программное обеспечение OpenLDAP.

Тема 303	<a href="#"><u>Подготовка к экзамену LPI 301: конфигурирование</u></a>	Узнайте более подробно о том, как настраивать программное обеспечение OpenLDAP.
Тема 304	<a href="#"><u>Подготовка к экзамену LPI 301: использование</u></a>	Узнайте, как следует выполнять поиск по дереву каталога LDAP и использовать утилиты OpenLDAP.
Тема 305	<a href="#"><u>Подготовка к экзамену LPI 301: интеграция и миграция</u></a>	Узнайте, как использовать LDAP в качестве источника данных для ваших системных приложений.
Тема 306	Подготовка к экзамену LPI 301: планирование пропускной способности	(Это руководство) Узнайте, как измерять степень загрузки системных ресурсов, решать связанные с ними проблемы и планировать дальнейшее масштабирование системы. См. подробные <a href="#"><u>цели</u></a> .

Чтобы сдать экзамен 301 (и получить сертификацию третьего уровня), вы должны:

- обладать несколькими годами опыта установки и поддержки Linux на большом числе компьютеров, используемых в различных целях
- обладать опытом интеграции с различными технологиями и операционными системами
- обладать профессиональным опытом или пройти профессиональную подготовку специалиста Linux корпоративного уровня (включая опыт, полученный при работе в другой роли)
- знать администрирование Linux на углубленном и высоком уровне, включая установку, управление, обеспечение безопасности, решение возникающих проблем и техническое обслуживание
- уметь использовать инструменты с открытым исходным кодом для проведения измерений, необходимых для планирования пропускной способности и решения проблем с ресурсами
- иметь профессиональный опыт применения LDAP для интеграции с сервисами UNIX® и Microsoft® Windows®, в том числе Samba, Pluggable Authentication Modules (PAM), электронной почтой и Active Directory
- уметь планировать, проектировать, разрабатывать, строить и реализовывать полную среду с использованием Samba и LDAP, а также проводить измерения для планирования производительности и оценки безопасности служб
- уметь создавать сценарии на Bash или Perl или знать как минимум один язык системного программирования (например, C)

Для дальнейшей подготовки к сертификации уровня 3 ознакомьтесь с [серийей руководств для подготовки к экзамену 301 Института LPI](#) (EN), а также со всей [серийей руководств developerWorks для подготовки к экзаменам LPI](#) (EN).

Институт Linux Professional Institute не дает рекомендаций по каким-либо конкретным материалам и методикам для подготовки к экзаменам, разработанным сторонними лицами.

## Об этом руководстве

Добро пожаловать в последнее из шести руководств, призванных помочь вам подготовиться к сдаче экзамена LPI 301, - "Планирование пропускной способности". Из этого руководства вы узнаете об измерении степени загрузки и оценке требований к ресурсам UNIX, а также о том, как определить, какие ресурсы потребуются в будущем.

Это руководство организовано в соответствии с целями LPI по этой теме. Условно говоря, чем выше вес цели, тем больше вопросов по этой теме будет на экзамене.

## Цели

В таблице 2 подробно перечислены цели этого руководства.

**Таблица 2. Планирование пропускной способности: цели экзамена, описанные в этом руководстве**

Цель экзамена LPI	Вес цели	Краткое описание цели
306.1 <u>Измерение степени загрузки ресурсов</u>	4	Оцените степень загрузки аппаратного обеспечения и пропускной способности сети.
306.2 <u>Решение проблем, связанных с использованием ресурсов</u>	4	Выполните диагностирование и устранение неполадок, связанных с использованием системных ресурсов.
306.3 <u>Оценка потребностей в ресурсах</u>	2	Оцените потребности ваших приложений в пропускной способности системы.
306.4 <u>Определение будущих потребностей в ресурсах</u>	1	Запланируйте масштабирование системы, анализируя текущее использование ресурсов и определяя будущие потребности ваших приложений.

## Необходимые условия

Чтобы извлечь максимум пользы из этого руководства, вы должны обладать глубокими знаниями Linux и иметь работающую Linux-систему, на которой вы сможете практиковаться в выполнении рассматриваемых задач.

Если ваши базовые знания Linux немного устарели, вы можете сначала ознакомиться [с руководствами для экзаменов LPIC-1 и LPIC-2](#).

Различные версии программ могут выводить данные в различных форматах, поэтому результаты, полученные вами, могут отличаться от листингов и рисунков, приведенных в этом руководстве.

## Требования к системе

Чтобы выполнить примеры, приведенные в этом руководстве, вам потребуется рабочая станция под управлением Linux с пакетом OpenLDAP и поддержкой РАМ. Большинство современных дистрибутивов удовлетворяют этим требованиям.

# Подготовка к экзамену LPI 301: Тема 306. Планирование пропускной способности

*Профессионал Linux высокого уровня (LPIC-3)*

[Шон Уолберг](#), старший сетевой инженер, независимый писатель

**Описание:** В этом руководстве Шон Уолберг поможет вам подготовиться к экзамену института Linux Professional Institute на квалификацию професионала Linux® высокого уровня (LPIC-3). В этом руководстве, последнем из [серии из шести руководств](#), Шон расскажет о мониторинге ресурсов вашей системы, об устранении проблем, связанных с использованием системных ресурсов, а также о том, как оценивать пропускную способность системы.

## Измерение степени загрузки ресурсов

В этом разделе описывается материал по теме 306.1 экзамена на професионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 4.

Из этого раздела вы узнаете, как:

- Измерить степень загрузки центрального процессора
- Измерить степень загрузки памяти
- Измерить степень загрузки средств дискового ввода/вывода
- Измерить степень загрузки средств сетевого ввода/вывода
- Измерить пропускную способность функций межсетевой защиты и маршрутизации
- Получить карту использования пропускной способности клиентами

Работа компьютера основана на использовании следующих аппаратных ресурсов: центральный процессор (CPU, ЦП), оперативная память, дисковая и сетевая подсистемы. Вы измеряете эти ресурсы, чтобы получить представление о текущем состоянии системы, а также о проблемах, которые могут возникнуть в будущем. Посмотрев на результаты измерений за определенный промежуток времени, например, за несколько месяцев, можно получить более полную картину. Часто полученные результаты можно экстраполировать на будущие периоды, что поможет вам определить время, через которое какой-либо из ресурсов исчерпает себя. Помимо этого, используя накопленную информацию, вы можете разработать математическую модель вашей системы, с помощью которой сможете более точно предсказывать работу системы в будущем.

Для выполнения задач серверы всегда используют несколько аппаратных компонентов. Для успешного выполнения задачи может потребоваться доступ к диску для получения данных, а также некоторый объем свободной памяти для хранения этих данных во время их обработки центральным процессором. Если системе не будет хватать какого-нибудь ресурса, производительности системы пострадает. Центральный процессор не может обрабатывать информацию, пока она не будет считана с диска, а информация, в свою очередь, не может быть помещена в память, если вся память заполнена. Таким образом, между всеми компонентами существует тесная взаимосвязь. При заполнении памяти операционная система начинает выгружать ее фрагменты на диск. Кроме того, дополнительная память забирается из буферов, которые используются для ускорения дисковых операций.

## Что представляют собой ресурсы

Прежде чем вы сможете извлечь пользу из измерения ресурсов, вы должны понять, что вы измеряете. После этого вы сможете получать полезную информацию, такую как текущее состояние и история работы системы, а также предсказывать ее работу в будущем.

## Центральный процессор

Центральный процессор компьютера выполняет все необходимые приложениям вычисления, вызывает на исполнение команды дисковой подсистемы и другого периферийного оборудования, а также следит за работой ядра операционной системы. В каждый момент времени центральный процессор выполняет только одну задачу независимо от того, является ли эта задача ядром или простым приложением. Текущая задача может быть прервана

аппаратным сигналом, называемым *прерыванием* (interrupt). Прерывания возникают в результате внешних событий, таких как получение сетевого пакета, или внутренних событий, таких как импульс системного тактового генератора (в Linux это называется *tick*). Когда возникает прерывание, обработка текущей задачи приостанавливается и запускается процедура определения следующего действия, которое должна выполнить система.

Когда время, отведенное на выполнение текущего процесса, истекает, ядро может переключиться на выполнение другого процесса, используя процедуру, называемую *переключением контекста* (context switch). Переключение на выполнение другого процесса может произойти досрочно, если процесс вызывает какую-либо команду ввода/вывода, такую как чтение с диска. Скорость работы компьютера намного больше скорости работы диска, поэтому в то время, когда центральный процессор ожидает завершения дисковой операции приостановленного процесса, он может выполнять другие задачи.

Когда мы говорим о ЦП в системе Linux, необходимо учитывать несколько факторов. Во-первых, процент времени, в течение которого процессор находится в режиме простоя, в сравнении со временем, в которое процессор выполняет полезную работу (на самом деле центральный процессор всегда *что-то* делает – считается, что процессор простояивает, если ни одна задача не ожидает выполнения). Производительность работы процессора максимальна, когда процент его простоя равен нулю. Та часть работы ЦП, которая отлична от простоя, делится на системное время и время пользователя; *системное время* означает время, потраченное на работу ядра, а *время пользователя* – время, потраченное на обработку запросов пользователя. Время простоя делится на время простоя ядра по причине отсутствия каких-либо задач и на время простоя ядра по причине ожидания каких-либо сигналов от операций ввода/вывода.

Измерение показаний этих счетчиков является непростой задачей, поскольку для получения точных значений потребовалось бы, чтобы процессор тратил все свое время на определение того, что он делает! Ядро выполняет проверку текущего статуса (system, user, iowait, idle) около 100 раз в секунду и использует полученные значения для вычисления процентных соотношений.

Другой величиной для определения загрузки центрального процессора в Linux является *средняя загрузка* (load average). Эта величина не привязана непосредственно к загрузке ЦП, а представляет собой экспоненциальное взвешивание количества задач, находящихся в очереди на исполнение, за последнюю минуту, 5 и 15 минут. Более подробно эта величина будет рассмотрена позже.

Другими характеристиками работы ядра являются интенсивность генерации прерываний (interrupt load) и переключения контекста (context switches); в их отношении не существует верхних границ, но чем больше выполняется прерываний и переключений контекста, тем меньше времени остается у центрального процессора на выполнения задач пользователя.

## Память

В системе существует два вида памяти: реальная память и область подкачки. *Реальная память* – это емкость модулей ОЗУ, установленных на материнской плате. *Область подкачки* – это пространство для временного хранения данных, используемое в тех случаях, когда система пытается выделить больше ОЗУ, чем физически установлено в системе. В этой ситуации страницы ОЗУ выгружаются на диск с целью освободить место для размещения текущих данных. Когда выгруженные на диск данные требуются снова, они загружаются обратно в ОЗУ.

ОЗУ может использоваться приложениями или системой, либо не использоваться вообще. Система использует ОЗУ двумя способами: в качестве буфера для линейных дисковых блоков (входящих или исходящих) и в качестве файлового кэша. Размеры буферов и кэша являются динамическими, поэтому при необходимости эта память может быть предоставлена

приложениям. Вот почему в большинстве ситуаций создается впечатление, что в Linux-системах нет свободной памяти – система выделяет всю неиспользованную память под буферы и кэш.

Область свопинга располагается на диске. Большое количество операций свопинга замедляет работу системы и служит признаком нехватки ОЗУ.

## Диск

Диском называют устройство длительного хранения данных. Это может быть жесткий диск, флэш-диск или лента (все эти устройства называются *блочными устройствами*).

Исключением является RAM-диск, который ведет себя как блочное устройство, но располагается в ОЗУ (RAM); при выключении системы все данные на этом диске теряются. Наиболее распространенным типом диска является жесткий диск, поэтому обсуждение в этом руководстве будет сфокусировано именно на этом носителе информации.

Для описания диска используются две категории параметров: объем и скорость. *Свободное пространство* диска – это число байтов на диске, доступных для использования. *Занятое пространство* – это любая область, используемая файловой системой или же недоступная для использования по другим причинам. Помните, что многие производители указывают объем диска в гигабайтах, состоящих из 1 000 000 000 байтов, тогда как операционная система понимает один гигабайт как степень двойки – 1 073 741 824; таким образом, для покупателя реальный объем составляет 93% от заявленного. Эти "потери" не учитываются в качестве занятого пространства, но если вы не обратите на это внимание, ваши расчеты окажутся неверными.

Второй величиной, характеризующей диск, является *скорость*, которая показывает, насколько быстро происходит получение данных с диска. Чтобы центральный процессор смог получить данные с диска в результате своего запроса, выполняются следующие действия:

1. Запрос помещается в очередь ядром и ожидает своего перенаправления на диск (время ожидания).
2. Команда направляется контроллеру диска.
3. Диск перемещает головки к требуемому блоку (время позиционирования).
4. Головки диска считывают данные.
5. Данные возвращаются центральному процессору.

Каждый из этих шагов измеряется по-разному, а иногда не измеряется вовсе. *Время обслуживания* включает в себя последние три шага и показывает, как долго выполняется запрос после его получения. *Время ожидания* включает в себя всю процедуру – как время нахождения запроса в очереди, так и время обслуживания.

Частью действий по оптимизации, которую выполняет ядро, является переупорядочивание и объединение на шаге 1 запросов, находящихся в очереди, с целью минимизации количества обращений к диску. Эта функция называется *элеватором*, и на протяжении прошедших лет для нее использовались различные алгоритмы.

## Сеть

Linux-система выполняет две основные сетевые роли: роль клиента, когда приложение на сервере получают и отправляют пакеты, и роль маршрутизатора (или брандмауэра, или моста). Во втором случае пакеты приходят на один сетевой интерфейс и отправляются с другого (возможно, после выполнения некоторой фильтрации или дополнительной проверки).

Производительность сети чаще всего характеризуется такими величинами, как количество битов (килобитов, мегабитов или гигабитов) в секунду и количество пакетов в секунду.

Измерение количества пакетов в секунду часто оказывается не очень полезным, поскольку

имеются фиксированные издержки для каждого пакета, выражющиеся в более низкой пропускной способности для пакетов более мелкого размера. Не путайте скорость сетевого интерфейса (100 Мбит/с или 1 Гбит/с) с предполагаемой скоростью прохождения или передачи данных от компьютера. Здесь играют роль несколько внешних факторов, включая задержки и производительность удаленной стороны подключения, не говоря уже о настройках сервера.

## Очереди

Очереди не очень вписываются в одну категорию с другими ресурсами, но они так часто фигурируют в мониторинге производительности, что о них следует упомянуть. *Очередь* – это последовательность, в которой запросы ожидают своего выполнения. Очереди используются ядром множеством способов, начиная от очереди на выполнение, содержащей список процессов, которые необходимо выполнить, и заканчивая дисковыми очередями, сетевыми очередями и аппаратными очередями. Обычно очередью называют участок памяти, который ядро использует для отслеживания конкретного набора задач, но это также может быть область памяти аппаратного компонента, управляемая этим компонентом.

Очереди влияют на производительность следующим образом. Во-первых, когда в очередь поставлено слишком много заданий, любое новое задание теряется. Например, если на сетевой интерфейс приходит слишком много пакетов, некоторые из них отбрасываются (в сетевых терминах это называется *tail drop* - отбрасывание конца очереди). Во-вторых, если очередь чрезмерно (а иногда недостаточно) используется, то другие компоненты не работают так, как это необходимо. Частое нахождение большого числа процессов в очереди выполнения может означать, что центральный процессор перегружен.

## Оценка производительности

Для оценки производительности Linux-систем существует несколько инструментов. Некоторые из них непосредственно измеряют загрузку процессора, дисковой подсистемы, памяти и сети, а другие отображают индикаторы, такие как использование очереди, создание процессов и возникающие ошибки. Одни инструменты отображают моментальные значения, другие – значения, усредненные за некоторый период времени. Однаково важно понимать как то, каким образом выполнялись измерения, так и то, что было измерено.

### vmstat

**vmstat** – это утилита для оценки наиболее часто используемых величин в реальном времени. Самое главное, что необходимо знать о **vmstat**, – это то, что первоначальные показания этой утилиты представляют собой значения, усредненные за время работы системы с момента ее загрузки. Обычно их можно смело игнорировать. Чтобы утилита **vmstat** отображала текущую информацию, укажите в командной строке интервал обновления (в секундах). В листинге 1 показан результат выполнения команды **vmstat 5**.

### Листинг 1. Вывод данных утилиты **vmstat 5**

```
# vmstat 5
procs -----memory----- -----swap-----io-----system-----cpu-----
 r b    swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0 3   17780  10304  18108 586076    0    0   2779   332    1    1  3  4 76 17  0
 1 2   17780  10088  19796 556172    0    0   7803  3940  2257 4093 25 28 14 34  0
 0 2   17780   9568  19848 577496    0    0  18060  1217 1610  910  0  3 48 49  0
 0 0   17780  51696  20804 582396    0    0   9237  3490 1736  839  0  3 55 41  0
```

В листинге 1 показан вывод данных команды **vmstat** с интервалом обновления в 5 секунд.

Первая строка содержит значения, усредненные за время работы системы с момента загрузки, поэтому ее следует проигнорировать. Первые два столбца относятся к процессам. Числа под заголовком **r** представляют собой количество процессов в очереди выполнения на момент измерения. Процессы в очереди на выполнение ожидают своей обработки центральным процессором. В следующем столбце отображено количество процессов, заблокированных операциями ввода/вывода; это означает, что данные процессы ожидают возвращения некоторой части данных ввода/вывода и не могут быть прерваны.

Столбцы в группе **memory** содержат результаты моментальных измерений системной памяти, выраженные в килобайтах (1024 байтов). В столбце **swpd** отображено количество памяти, выгруженной на диск. В столбце **free** отображено количество памяти, не используемой приложениями, буферами или кэшем. Не удивляйтесь, если это количество невелико (дополнительную информацию о том, что на самом деле представляет собой свободная память, см. в обсуждении команды [free](#)). Столбцы **buff** и **cache** показывают, какое количество памяти выделено под буфера и кэш. В буферах хранятся линейные дисковые блоки, а в кэше – файлы.

Первые две категории – это результаты моментальных измерений. Возможна ситуация, когда на короткий период времени вся свободная память окажется занятой, но будет вновь освобождена до наступления следующего интервала. Остальные значения являются усредненными за период выборки.

В столбцах группы **swap** отображено усредненное количество памяти (в килобайтах), загруженной (**si**) и выгруженной (**so**) на диск в секунду. В столбцах группы **io** отображено количество дисковых блоков в секунду, считанных со всех блочных устройств и отосланных им.

В столбцах группы **system** отображено количество прерываний (**in**) и переключений контекста (**cs**) в секунду. Прерывания генерируются устройствами (например, сетевой адаптер сообщает ядру о том, что имеется ожидающий своей очереди пакет) и системным таймером. В некоторых ядрах системный таймер генерирует 1 000 тактов в секунду, поэтому количество прерываний может быть достаточно велико.

Последняя группа величин показывает, что происходит с центральным процессором; значения выражены в процентах от его общего времени работы. Сумма этих пяти значений должна быть равна 100. В столбце **us** отображено среднее время, затраченное центральным процессором на обработку задач пользователей за период выборки, а в столбце **sy** – среднее время, затраченное на обработку системных задач. Столбец **id** показывает время простоя ЦП, а столбец **wa** – время ожидания процессором данных ввода/вывода (листинг 1 был получен на машине с большим количеством операций ввода/вывода, и вы можете видеть, что 34-49% времени работы процессора ушло на ожидание получения данных с диска). Последний столбец, **st** (steal time - украденное время), предназначен для серверов, на которых запущены гипервизор и виртуальные машины. Данное значение представляет собой время, в течение которого гипервизор мог бы работать с виртуальной машиной, но ему приходилось выполнять какие-то другие задачи.

Как видно из [листиングа 1](#), утилита **vmstat** позволяет получить большое количество информации о самых различных параметрах работы системы. Если во время вашей работы в системе возникнут какие-то неполадки, утилита **vmstat** является прекрасным способом установить их причины.

С помощью утилиты **vmstat** можно также получить некоторые полезные сведения об использовании диска каждым устройством, что позволяет более подробно узнать о параметрах категорий подкачки и ввода/вывода, представленных в листинге 1. Параметр **-d** выводит статистику о дисках, включая суммарное количество операций чтения и записи, выполненных для каждого диска. В листинге 2 показан фрагмент вывода, полученного при

выполнении команды `vmstat -d 5` (без вывода информации о неиспользуемых устройствах).

## Листинг 2. Использование `vmstat` для получения информации об использовании диска

```
disk-----reads-----writes-----IO-----
      total merged sectors      ms   total merged sectors      ms   cur    sec
hda    186212  28646 3721794  737428 246503 4549745 38981340 8456728      0   2583
hdd    181471  27062 3582080  789856 246450 4549829 38981624 8855516      0   2652
```

Каждый диск представлен отдельной строкой, а данные об операциях записи и чтении сгруппированы в соответствующих разделах, каждый из которых, в свою очередь, разделен на четыре группы: общее количество запросов, количество запросов, объединенных в дисковом элеваторе, количество считанных или записанных секторов и общее время обслуживания. Все эти значения являются счетчиками и в отличие от средних значений, полученных без использования параметра `-d`, увеличиваются вплоть до следующей перезагрузки системы.

Последняя группа величин в листинге 2, **IO**, отображает текущее количество операций ввода/вывода, выполняющихся в текущий момент, и общее количество секунд, затраченное на выполнение операций ввода/вывода с момента загрузки системы.

Из листинга 2 видно, что объем считанных данных практически одинаков для двух дисков, так же как и объем записанных данных. Эти два диска объединены в программный массив RAID1, поэтому такое поведение вполне ожидаемо. Информацию из листинга 2 можно использовать для выявления более медленных дисков или дисков с более высокой загрузкой по сравнению с остальными.

## iostat

Команда `iostat` очень похожа на пример утилиты `vmstat -d`, приведенный в листинге 2. Данная команда выводит подробную информацию об использовании каждого диска. Утилита `iostat` позволяет получить больше информации, чем `vmstat -d`. Так же, как и в случае с `vmstat`, вы можете передать команде `iostat` параметр, задающий интервал обновления. И опять же, как и в случае с `vmstat`, первая строка вывода содержит значения за все время работы системы с момента ее загрузки и поэтому обычно игнорируется. В листинге 3 показан вывод данных команды `iostat` с 5-секундным интервалом обновления.

## Листинг 3. Вывод данных команды `iostat`

```
$ iostat 5
Linux 26.20-1.3002.fc6xen (bob.ertw.com)        02/08/2008

avg-cpu: %user  %nice  %system %iowait  %steal  %idle
          0.85    0.13    0.35     0.75    0.01   97.90

Device:    tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda        1.86      15.24      13351     4740568    41539964
hdd        1.85      14.69      133.51    4570088    41540256
```

Для каждого интервала обновления в первой части вывода данных указана загрузка центрального процессора, которую также показывает команда `vmstat`. Однако здесь выводятся два разряда после запятой. Вторая часть вывода данных содержит все блочные

устройства, присутствующие в системе (чтобы ограничить набор выводимых устройств, укажите имена требуемых устройств в командной строке утилиты, например, `iostat 5 hda sda`). В первом столбце, **tps**, отображается количество передач в секунду, поступивших устройству после того, как запросы были объединены элеватором. Размеры передач не указываются. Последние четыре столбца содержат значения, выраженные в блоках по 512 байтов, и показывают количество считанных и записанных блоков в секунду, а также общее количество считанных и записанных блоков соответственно. Если вы хотите видеть эти значения в килобайтах или мегабайтах, используйте параметр командной строки `-k` или `-m` соответственно. При необходимости вы можете использовать параметр `-p`, который отображает информацию на уровне дисковых разделов.

Вы можете получить намного больше информации, используя параметр `-x`, что показано в листинге 4. Вывод данных в листинге 4 ограничен одним диском. Мы изменили форматирование результатов, чтобы они поместились на ширине страницы.

#### Листинг 4. Расширенная информация, полученная в результате выполнения `iostat`

```
# iostat -x 5 hda
..... CPU information removed ...
Device:      rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s  avgrq-sz
hda          16669.31    1.49  756.93   1.49 139287.13    27.72   18369
              avgqu-sz   await   svctm %util
                  1.58     208    1.28  96.83
```

Первые шесть значений касаются операций чтения и записи в секунду. Столбцы **rrqm/s** и **wrqm/s** содержат количество запросов на чтение и запись, которые были объединены. Столбцы **r/s** и **w/s**, наоборот, содержат количество операций записи и чтения, переданных на диск. Таким образом, процент объединенных дисковых запросов составляет  $16669 / (16669 + 757) = 95\%$  от общего количества. Столбцы **rsec/s** и **wsec/s** показывают скорости чтения и записи, выраженные в секторах в секунду.

В следующих четырех столбцах содержится информация о дисковых очередях и временных характеристиках. Столбец **avgrq-sz** содержит средний размер запросов (выражается в секторах) к устройству. Столбец **avgqu-sz** содержит среднюю длину дисковой очереди за период выборки. Столбец **await** содержит среднее время ожидания (в миллисекундах), то есть среднее время, прошедшее от момента отправки запроса ядру до его обратного возвращения. Столбец **svctm** содержит среднее время обслуживания (в миллисекундах), то есть, время, прошедшее с момента, когда дисковый запрос покинул очереди и был отправлен на диск, до его возвращения.

Последний столбец, **%util**, содержит процент времени, затраченного системой на выполнение операций ввода/вывода для данного устройства, также называемый *насыщением*. Значение 96.83% в листинге 4 означает, что за этот промежуток времени диск работал почти на полную мощность.

#### `mpstat`

Утилита `mpstat` сообщает детальную информацию о центральном процессоре (или обо всех процессорах в многопроцессорных системах). Большую часть этой информации в той или иной форме можно получить из утилит `iostat` и `vmstat`, однако `mpstat` предоставляет данные для каждого отдельного процессора. В листинге 5 показан результат выполнения команды `mpstat` с 5-секундным интервалом обновления. В отличие от случаев утилитам `iostat` и `vmstat`, здесь вы игнорировать первую строку не следует.

## Листинг 5. Информация о центральном процессоре, полученная в результате выполнения **mpstat**

```
# mpstat -P 0 5
Linux 2.620-1.3002.fc6xen (bob.ertw.com)        02/09/2008

09:45:23 PM CPU %user %nice %sys %iowait %irq %soft %steal %idle intr/s
09:45:25 PM    0 77.61 21.89 0.00 0.00 0.50 0.00 0.00 0.00 0.00 155.22
09:45:27 PM    0 68.16 30.85 1.00 0.00 0.00 0.00 0.00 0.00 0.00 154.73
```

Параметр командной строки **-P 0** указывает, что необходимо вывести информацию для первого ЦП (нумерация начинается с 0). Также вы можете указать параметр **-P ALL**, чтобы вывести информацию обо всех процессорах по отдельности. Утилита **mpstat** возвращает следующие значения:

- **%user**: процент времени, затраченного на обработку задач пользователей, за исключением nice-задач
- **%nice**: процент времени, затраченного на обработку задач пользователей в режиме nice (с пониженным приоритетом)
- **%sys**: процент времени, затраченного на обработку задач ядра
- **%iowait**: процент времени, затраченного на ожидание операций ввода/вывода во время простоя
- **%irq**: процент времени, затраченного на обработку аппаратных прерываний
- **%soft**: процент времени, затраченного на обработку программных прерываний
- **%steal**: процент времени, "украденного" гипервизором у виртуальной машины
- **intr/s**: среднее количество прерываний в секунду

## **pstree**

Информация о том, какие из процессов являются родительскими для других процессов, полезна в тех случаях, когда вы отслеживаете использование системных ресурсов. Одним из способов обнаружить такую взаимосвязь является использование команды **ps -ef** и полученного родительского идентификатора процесса для отслеживания всей цепочки вплоть до PID 1 (**init**). Также вы можете использовать параметр **ps -efjH**, который сортирует результаты в виде дерева родитель-потомок и содержит данные о времени использования центрального процессора.

Утилита **pstree** выводит на экран дерево процессов в более наглядном графическом формате, а также группирует несколько экземпляров одного и того же процесса в одну строку. В листинге 6 показаны результаты выполнения утилиты **pstree**, которой был передан идентификатор PID демона Postfix.

## Листинг 6. Вывод данных утилиты **pstree**

```
[root@sergeant ~]# pstree 7988
master--anvil
          |---cleanup
          |---local
          |---pickup
          |---proxymap
          |---qmgr
          |---2*[smtpd]
```

```
└─2*[trivial-rewrite]
```

Главный процесс, разумно называющийся **master**, породил несколько других процессов, среди которых процессы **anvil**, **cleanup** и **local**. Последние две строки вывода данных отображены в формате  $N^*$  [процесс], где *процесс* – это имя процесса, а  $N$  – количество дочерних процессов с этим именем. Если в дополнение к квадратным скобкам ([])) *процесс* заключен в фигурные скобки ({}), это может означать, что выполняются  $N$  потоков (обычно команда **ps** не показывает потоки, если только вы не используете параметр -L).

### w, uptime и top

Эти утилиты собраны в одну группу, поскольку именно к ним обычно обращаются в первую очередь при диагностировании проблем. В листинге 7 приведен вывод данных команды **w**.

#### Листинг 7. Вывод данных команды w

```
# w
12:14:15 up 33 days, 15:09,  2 users,  load average: 0.06, 0.12, 0.09
USER     TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
root     tty2      -           17Jan08  18days  0.29s  0.04s login -- root
root     pts/0      bob        Sat22    0.00s  0.57s  0.56s -bash
```

Первая строка вывода данных команды **w** содержит основную информацию. Первая часть, "12:14:15 up 33 days, 15:09", сообщает текущее время, после которого указано время непрерывной работы системы – 33 дня, 15 часов и 9 минут. Вторая часть, "2 users", сообщает количество работающих в системе пользователей. Заключительная часть – это средняя загрузка системы за периоды 1, 5 и 15 минут.

Средняя загрузка является средневзвешенным значением количества процессов в очереди на выполнение за указанный период времени. Чем выше средняя загрузка, тем больше процессов пытаются обратиться к центральным процессорам. Средние загрузки не нормализованы относительно количества процессоров; это означает, что средняя загрузка и количество процессоров не взаимосвязаны.

Для использования средней загрузки вы также должны знать о взвешивании. Средняя загрузка обновляется каждые 5 секунд, при этом используется более старая информация, играющая менее значимую роль в вычислениях. Если в вашей системе произошел непосредственный переход от 0 процессов, находящихся в очереди на выполнении, к 1 процессу, график средней минутной загрузки за последующую минуту не будет представлять собой прямую линию, а будет являться кривой, сначала резко возрастающей, а затем спадающей до следующей 60-секундной отметки. Для получения более подробной информации о том, как рассчитывается средняя загрузка, обратитесь к разделу [Ресурсы](#).

Практический результат взвешивания средней загрузки заключается в том, что изменения фактической загрузки во время измерения сглаживаются, но текущее состояние отражается более точно, в особенности в ежеминутных средних значениях.

За первой строкой следует список работающих в системе пользователей, включающий время входа в систему, их местоположение, а также информацию об использовании центрального процессора. Первый пользователь, **root**, вошел в систему с терминала **tty2** (локальная консоль) и простоял уже 18 дней. Второй пользователь тоже **root**, но подключился по сети и в данное время находится в командной оболочке. Столбцы **JCPU** и **PCPU** позволяют понять, сколько процессорного времени потратил пользователь; первый столбец содержит

прошлые задания, тогда как столбец **PCPU** показывает процесс, который использует пользователь в текущий момент.

Вывод данных утилиты **uptime** содержит ту же информацию, что и первая строка вывода данных утилиты **w**, но без информации о пользователях. С практической точки зрения более полезной является утилита **w**, поскольку она показывает дополнительную информацию о пользователях, и ее проще набрать в командной строке!

Другой популярной командой является команда **top**, показывающая в дополнение к некоторым другим величинам непрерывно обновляющийся список самых значимых процессов (отсортированных по степени использования памяти или ресурсов центрального процессора). На рисунке 1 показан снимок экрана, на котором команда **top** запечатлена в действии.

**Рисунок 1. Команда top в действии**

The screenshot shows a terminal window titled "sean@bob:~" displaying the output of the "top" command. The top part of the screen shows system statistics: "top - 20:14:44 up 5 days, 13:29, 10 users, load average: 0.47, 0.35, 0.14", "Tasks: 220 total, 3 running, 215 sleeping, 0 stopped, 2 zombie", "Cpu(s): 5.6%us, 3.0%sy, 0.0%ni, 91.0%id, 0.0%wa, 0.3%hi, 0.0%si, 0.0%st", "Mem: 961780k total, 955052k used, 6728k free, 30196k buffers", and "Swap: 2915776k total, 144k used, 2915632k free, 456700k cached". Below this is a table listing processes:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
13435	root	15	0	164m	76m	6696	S	3.7	8.1	15:36.17	X
13485	sean	15	0	17800	9164	6704	S	1.7	1.0	0:30.46	metacity
32137	sean	15	0	41008	13m	9156	S	1.7	1.5	0:01.71	gnome-terminal
32122	sean	15	0	134m	35m	28m	R	1.3	3.8	0:04.92	audacious
32229	sean	15	0	2348	1060	776	R	0.7	0.1	0:00.09	top
7348	sean	18	0	4716	968	464	S	0.3	0.1	0:35.10	nxnode
23201	sean	15	0	14816	4332	3740	S	0.3	0.5	0:00.33	pam-panel-icon
1	root	15	0	2060	584	500	S	0.0	0.1	0:00.42	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.03	ksoftirqd/0
4	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.32	events/0
6	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khelper
7	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
9	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	xenwatch
10	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	xenbus
50	root	10	-5	0	0	0	S	0.0	0.0	0:00.24	kblockd/0

Первая строка показывает ту же информацию, что и утилиты **uptime** – время непрерывной работы и среднюю загрузку системы. Вторая строка содержит ряд процессов.

Выполняющиеся (*running*) процессы находятся в очереди на выполнение, спящие (*sleeping*) процессы ожидают, пока кто-нибудь выведет их из этого состояния, обратившись к ним.

Остановленный (*stopped*) процесс был временно приостановлен, вероятно, потому, что выполняется его трассировка или отладка. Зомби-процесс (*zombie*) завершил работу, но родительский процесс не опознал этого.

### Числа не сходятся

Величина **VIRT** = **RES** + **SWAP**, значит, величина **SWAP** = **VIRT** - **RES**. Посмотрев на процесс с идентификатором PID 13435, вы можете увидеть, что величина **VIRT** имеет значение 164m, а **RES** - 76m, значит величина **SWAP** должна иметь значение 88m. Тем не менее, статистика в верхней части экрана показывает, что используется только 144 КБ swap-пространства! Это можно проверить, включив с помощью клавиши **f** дополнительные поля (например, **swap**), находясь в окне **top**.

Как оказывается, swap-пространство относится не только к выгруженным на диск страницам памяти. Двоичный код и библиотеки приложения не обязательно должны оставаться в памяти все время. Ядро может пометить определенные страницы памяти как неиспользуемые в текущий момент, но поскольку двоичный код располагается по известному адресу на диске,

нет необходимости использовать для него swap-файл. Тем не менее это дисковое пространство рассматривается как swap-область, поскольку эта часть кода не является резидентной. Кроме того, память может быть выгружена в дисковый файл приложением. Поскольку размер всего приложения (VIRT) включает в себя распределенную память, но она не является резидентной (RES), она считается swap-областью.

Третья строка содержит статистику использования центрального процессора: в порядке очередности вы видите время, затраченное на обработку задач пользователей, системных задач и задач в nice-режиме, время простоя ЦП, время ожидания операций ввода/вывода, время обработки аппаратных и программных прерываний, и, наконец, украденное время (steal time). Эти значения являются процентными соотношениями времени за последний интервал обновления (по умолчанию 3 секунды).

Последние две строки верхнего раздела содержат статистику использования памяти. Первая строка показывает информацию о реальной памяти; из [рисунка 1](#) вы можете видеть, что в системе имеется 961 780 КБ ОЗУ (не считая памяти, выделенной ядру). Используется вся память, за исключением 6 728 КБ, при этом около 30 МБ выделено под буферы и 456 МБ – под кэш (размер кэша отображается в конце второй строки). Вторая строка показывает информацию об использовании swap-области: в системе имеется почти 3 ГБ swap-пространства, и лишь 144 КБ из этого объема используется.

В остальной части экрана отображается информация о выполняющихся в настоящее время процессах. Утилита `top` отображает максимальное количество процессов, которое умещается на экране. Для каждого процесса имеется отдельная строка, и этот список обновляется с заданным интервалом, отображая процессы по убыванию в соответствии с их ресурсоемкостью. Параметры каждого процесса содержатся в следующих столбцах:

- **PID**: идентификатор процесса
- **USER**: действующее имя пользователя процесса (если программа сменит пользователя командой `setuid(2)`, будет отображаться новое значение)
- **PR**: приоритет задачи, используемый ядром для определения очередности использования центрального процессора той или иной задачей
- **NI**: nice-уровень задачи, заданный системным администратором, чтобы повлиять на очередность использования центрального процессора той или иной задачей
- **VIRT**: размер виртуального образа процесса, являющийся суммой размера используемого ОЗУ (резидентная часть) и размера данных, находящихся в swap-области (размер области подкачки)
- **RES**: резидентная часть процесса, представляющая собой количество физического ОЗУ, используемого процессом
- **SHR**: количество разделяемой памяти приложения, например, разделяемая память SysV или динамические библиотеки (\*.so)
- **S**: состояние процесса, например, ожидание, выполнение или зомби
- **%CPU**: процент загрузки центрального процессора за последний интервал обновления
- **%MEM**: процент загрузки ОЗУ (за исключением swap-области) за последний интервал обновления
- **TIME+**: время в формате минуты:секунды:сотые, использованное процессом
- **COMMAND**: имя выполняющейся команды

Утилита `top` позволяет быстро увидеть, какими задачами центральный процессор используется больше всего, а также дает удобный обзор данных о центральном процессоре и памяти. Вы можете отсортировать выводимый список задач по степени загрузки памяти, нажав клавишу **M** в окне утилиты `top`.

**free**

После того, как вы познакомились с утилитой **top**, вы сможете разобраться с утилитой **free** без особых проблем. В листинге 8 показан вывод данных утилиты **free** с использованием параметра **-m**, указывающего утилите выводить значения в мегабайтах.

### Листинг 8. Использование команды **free**

```
# free -m
      total        used        free      shared      buffers      cached
Mem:       939         904          34          0         107        310
 -/+ buffers/cache:     486        452
Swap:      2847          0        2847
```

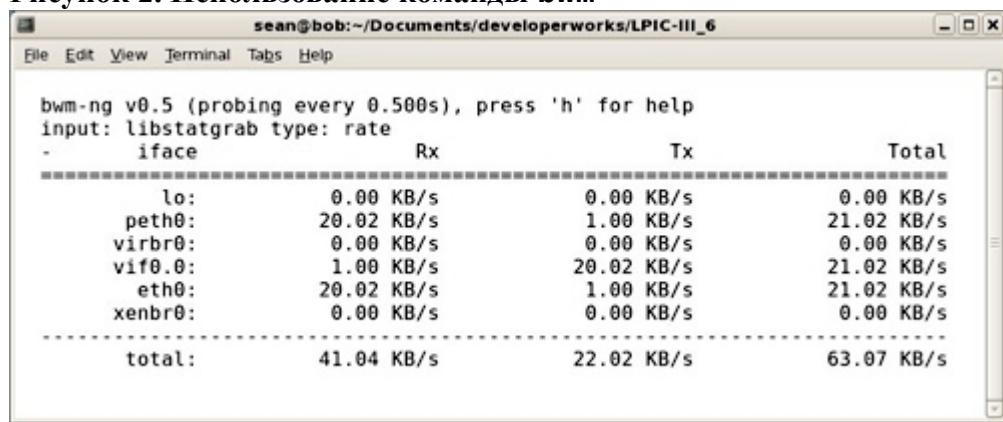
Утилита **free** выводит сведения об использовании памяти по нескольким направлениям. Первая строка показывает информацию, которую вы уже видели при запуске утилиты **top**. Вторая строка содержит сведения об используемой и свободной памяти, не принимая во внимание буферы и кэш. Из листинга 8 видно, что приложениям доступно для использования 452 МБ памяти; этот объем будет взят из свободной памяти (34 МБ), буферов (107 МБ) или кэша (310 МБ).

Последняя строка показывает ту же статистику для swap-области, что и утилита **top**.

### Получение сетевой статистики

Для получения сетевой статистики используются не такие прямые методы, как для центрального процессора, памяти и дисков. Основным способом является чтение счетчиков из **/proc/net/dev**, сообщающих о количестве передач через интерфейс, как в терминах пакетов, так и в терминах байтов. Если вы хотите получить значение за секунду, вы должны вычислить его самостоятельно, разделив разницу между двумя успешно измеренными величинами на интервал измерения. Другим способом является использование инструментов наподобие **bwm** для автоматизации сбора и отображения общей пропускной способности. На рисунке 2 показана работа утилиты **bwm**.

### Рисунок 2. Использование команды **bwm**



Утилита **bwm** показывает различные характеристики использования сетевых интерфейсов. На рисунке 2 показаны мгновенные значения, измеряемые каждые полсекунды, хотя можно посмотреть средние значения за 30 секунд, максимальную пропускную способность и счетчики байтов. Из рисунка 2 вы можете видеть, что интерфейс **eth0** получает трафик со скоростью около 20 КБ/с, который, вероятно, приходит с интерфейса **vif0.0**. Если вместо количества байтов в секунду вы хотите увидеть другие значения, вы можете переключаться между количеством битов, пакетов и ошибок с помощью клавиши **u**.

Для получения более детальной информации о том, какие хосты генерируют трафик, вам понадобится утилита **iftop**, которая имеет интерфейс, подобный интерфейсу утилиты **top**, отображающий сетевой трафик. Ядро не предоставляет эту информацию напрямую, поэтому для проверки пакетов, передаваемых по сетевому кабелю, **iftop** использует библиотеку **pcap**, для чего необходимы полномочия пользователя **root**. На рисунке 3 показана работа утилиты **iftop** при подключении к сетевому интерфейсу **eth2**.

**Рисунок 3. Вывод данных команды **iftop -i eth2****

	195Kb	391Kb	586Kb	781Kb	977Kb	
mybox	=> pub1.kernel.org			23.3Kb	19.0Kb	10.7Kb
mybox	<=			622Kb	518Kb	286Kb
mybox	=> sbproxy01.voip.les.net			78.1Kb	78.1Kb	78.2Kb
mybox	<=			77.3Kb	75.2Kb	75.8Kb
mybox	=> scfire-chi-aa05.stream.aol.com			4.62Kb	5.02Kb	5.27Kb
mybox	<=			124Kb	129Kb	130Kb
mybox	=> ool-4356bfb2.dyn.ptonline.net			0b	75.8Kb	18.9Kb
mybox	<=			0b	9.55Kb	2.39Kb
mybox	=> lists.groupstudy.com			0b	690b	201b
mybox	<=			0b	4.17Kb	1.07Kb
mybox	=> 501060010b57beae6.cn.shawcable.ne			6.33Kb	1.27Kb	324b
mybox	<=			6.00Kb	1.20Kb	307b
mybox	=> qb-in-f17.google.com			0b	1.81Kb	911b
mybox	<=			0b	354b	209b
mybox	=> crawl-66-249-70-75.googlebot.com			0b	583b	146b
mybox	<=			0b	506b	127b
255.255.255.255	=> 24.78.140.1			0b	0b	0b
	<=			1.32Kb	826b	816b
<b>TX:</b>	<b>cumm:</b> 1.44MB	<b>peak:</b> 303Kb		<b>rates:</b> 113Kb	183Kb	122Kb
<b>RX:</b>	<b>4.78MB</b>	<b>914Kb</b>		<b>831Kb</b>	<b>739Kb</b>	<b>497Kb</b>
<b>TOTAL:</b>	<b>6.22MB</b>	<b>1.00Mb</b>		<b>943Kb</b>	<b>922Kb</b>	<b>619Kb</b>

Утилита **iftop** показывает хосты, генерирующие самый большой трафик в вашей сети. По умолчанию для каждого сетевого взаимодействия используются две линии: одна линия используется для отправки данных, вторая – для получения. Если посмотреть на первый диалог между хостами **mybox** и **pub1.kernel.org**, то верхняя строка показывает трафик, переданный хостом **mybox**, а вторая строка – трафик, полученный им. Числа справа показывают средний трафик за последние 2, 10 и 40 секунд соответственно. Также вы видите черную полосу поверх имен хостов, которая является наглядным индикатором среднего значения за 10 секунд (шкала отображается в верхней части экрана).

Если проанализировать рисунок 3 более внимательно, то можно увидеть, что в первой передаче, вероятно, происходит скачивание данных, поскольку получено большое количество трафика (со средней скоростью около половины мегабита в секунду за последние 10 секунд) при небольшом количестве переданных данных. Исходящий трафик во второй передаче примерно равен входящему трафику, скорость постоянна и составляет около 75-78 кБ/сек. Это голосовой вызов G.711 через хост **les.net** моего VoIP-провайдера. Третья передача содержит входящий трафик со скоростью 128 КБ и небольшой исходящий трафик – это поток Интернет-радио.

Важен выбор интерфейса, к которому вы подключаетесь. На рисунке 3 используется внешний интерфейс на брандмауэре, который видит все пакеты после того, как они проходят IP-маскирование. Это приводит к тому, что внутренний адрес теряется. При использовании другого интерфейса, например, внутреннего, эта информация может сохраняться.

### **sar**

Утилите **sar** посвящена целая отдельная статья (обратитесь к разделу [Ресурсы](#)). **sar** измеряет множество ключевых величин каждые 10 минут и позволяет вам получить эти данные. Вышеописанные инструменты дают ответ на вопрос "что происходит в данный момент?". Утилита **sar** отвечает на вопрос "что произошло на этой неделе?". Обратите внимание на то, что **sar** хранит данные только за последние 7 дней.

Чтобы настроить сбор данных, необходимо добавить две строки в файл **crontab** пользователя **root**. В листинге 9 приведен типовой пример файла **crontab** для использования **sar**.

### Листинг 9. Конфигурация файла crontab для сбора данных утилитой sar

```
# Сбор статистики каждые 10 минут
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1 -d 1 1
# Создание ежедневных отчетов и очистка старых файлов
0 * * * * /usr/lib/sa/sa2 -A
```

Первая строка выполняет команду **sa1** для сбора данных каждые 10 минут; эта команда запускает команду **sadc**, выполняющую фактический сбор данных. Эта задача является самостоятельной – она знает, в какой файл необходимо записывать информацию, и не требует дополнительного конфигурирования. Вторая строка выполняет команду **sa2** в полночь, чтобы очистить старые файлы данных и собрать данные за прошедший день в удобочитаемый текстовый файл.

Важно проверить, как работает утилита **sar** в вашей системе, прежде чем использовать ее данные. В некоторых системах отключен сбор статистики дисков, и для его включения необходимо добавить параметр **-d** в командную строку вызова **sa1** (в листинге 9 этот параметр добавлен).

Когда у вас имеются накопленные данные, вы можете запустить утилиту **sar** безо всяких дополнительных параметров, чтобы посмотреть использование центрального процессора за день. В листинге 10 приведен фрагмент вывода данных.

### Листинг 10. Пример вывода данных команды sar

```
[root@bob cron.d]# sar | head
Linux 2.6.20-1.3002.fc6xen (bob.ertw.com)          02/11/2008

12:00:01 AM      CPU      %user      %nice      %system      %iowait      %steal      %idle
12:10:01 AM      all       0.18       0.00       0.18        3.67       0.01      95.97
12:20:01 AM      all       0.08       0.00       0.04       0.02       0.01      99.85
12:30:01 AM      all       0.11       0.00       0.03       0.02       0.01      99.82
12:40:01 AM      all       0.12       0.00       0.02       0.02       0.01      99.83
12:50:01 AM      all       0.11       0.00       0.03       0.05       0.01      99.81
01:00:01 AM      all       0.12       0.00       0.02       0.02       0.01      99.83
01:10:01 AM      all       0.11       0.00       0.02       0.03       0.01      99.83
```

К этому моменту вам должны быть знакомы значения, приведенные в листинге 10: они представляют собой различные характеристики работы центрального процессора, которые выводятся утилитами **top**, **vmstat** и **mpstat**. Вы можете получить более подробную информацию, используя один или несколько параметров командной строки, приведенных в таблице 3.

Таблица 3. Краткий обзор параметров утилиты sar

Параметр	Пример	Описание
-A	<b>sar -A</b>	Выводит <i>всю информацию</i> . Этот параметр, вероятно, не понадобится вам до тех пор, пока вы не захотите сохранить эти результаты в текстовом файле. Если же вам это нужно, данный процесс запускается каждую ночь как часть команды <b>sa2</b> .
-b	<b>sar -b</b>	Показывает транзакции и блоки, переданные и

		считанные с блочных устройств, подобно утилите <b>iostat</b> .
-B	<b>sar -B</b>	Показывает статистику подкачки, ту же, что и утилита <b>vmstat</b> .
-d	<b>sar -d</b>	Показывает активность диска, подобно команде <b>iostat -x</b> , включая время ожидания и обслуживания, а также длину очереди.
		Показывает активность сетевого интерфейса (подобно команде <b>bwm</b> ), если используется ключевое слово <b>DEV</b> , или статистику NFS-клиентов, если используется ключевое слово <b>NFS</b> (используйте ключевое слово <b>NFSD</b> для сбора статистики работы демона NFS-сервера).
-n	<b>sar -n</b> <b>DEV or</b> <b>sar -n</b> <b>NFS</b>	Ключевое слово <b>EDEV</b> показывает информацию об ошибках на сетевом интерфейсе.
-q	<b>sar -q</b>	Показывает информацию об очереди на выполнение, суммарных размерах списков обработки и средних загрузках. Эти значения соответствуют значениям, выводимым утилитами <b>vmstat</b> и <b>uptime</b> .
-r	<b>sar -r</b>	Показывает информацию об использовании памяти, swap-области, кэша и буферов (как и утилита <b>free</b> ).
-f	<b>sar -f</b> <b>/var/log/</b> <b>sa/sa11</b>	Считывает информацию из указанного файла. Имена файлов начинаются со дня месяца.
-s	<b>sar -s</b> <b>08:59:00</b>	Указывает время начала интересующего интервала вывода информации, начиная с момента первого измерения после указанного времени. Если вы укажете <b>09:00:00</b> , первое измерение будет выполнено в <b>09:10</b> , поэтому отнимите одну минуту от желаемого времени.
-e	<b>sar -e</b> <b>10:01:00</b>	Указывает время окончания интересующего интервала вывода информации. Следует добавить одну минуту к желаемому времени, чтобы убедиться, что все сработает.

Вы можете комбинировать различные параметры для получения нескольких отчетов или отдельного файла со временем начала и окончания наблюдения.

## df

Ваш жесткий диск является ограниченным ресурсом. Если в разделе заканчивается место, приготовьтесь к проблемам. Команда **df** показывает информацию о состоянии диска. В листинге 11 показан вывод команды **df -h**, которая выводит результаты в более дружественном формате.

### Листинг 11. Проверка использования дискового пространства с помощью команды **df -h**

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                    225G  169G   44G  80% /
/dev/hda1        99M   30M   64M  32% /boot
```

tmpfs	474M	0	474M	0%	/dev/shm
-------	------	---	------	----	----------

В листинге 11 показана одна файловая система корневого раздела размером 225 ГБ, 44 из которых свободны. Раздел /boot имеет размер 99 МБ, 64 из которых свободны. tmpfs – это специальная файловая система, не относящаяся к какому-либо конкретному устройству. Если раздел заполнен, вы увидите, что в нем не осталось свободного пространства, и что он используется на 100%.

## Решение проблем, связанных с использованием ресурсов

В этом разделе описывается материал по теме 306.2 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 4.

Из этого раздела вы узнаете, как:

- Определять наиболее вероятные источники проблем в зависимости от симптомов
- Определять узкие места в работе системы

В предыдущем разделе было рассказано, как можно посмотреть различные счетчики производительности в Linux-системе. Теперь настало время применить рассмотренные команды для решения проблем, связанных с использованием ресурсов, возникающих в вашей системе.

## Методология диагностирования неисправностей

Прежде чем приступить к рассмотрению ограничений на количество ресурсов и детальному изучению работы вашей системы, спланируйте стратегию решения проблемы. Большинство стратегий сводятся к четырем шагам:

1. Выявление симптомов.
2. Определение первопричины неисправности.
3. Внесение исправлений.
4. Оценка результатов.

### Выявление симптомов

Первым шагом в решении проблемы является выявление симптомов. Примером такого симптома может быть ситуация типа «электронная почта работает слишком медленно» или «у меня закончилось место на диске». Симптомы приводят к жалобам пользователей, которые не будут довольны до тех пор, пока эти симптомы не прекратятся. Однако не путайте симптомы с проблемой: очень часто проблема заключается совсем не в том, о чем сообщают ваши пользователи, хотя она и является причиной возникновения симптомов.

После того как вы выявили все симптомы, попытайтесь подсчитать количество случаев возникновения неполадок и выяснить условия, при которых это происходит. Слишком медленная работа системы электронной почты может заключаться в том, что получение электронных писем после их отправки, занимавшее секунды, теперь занимает несколько часов. А пользователь, у которого закончилось место на диске, должен был при этом выполнять какие-то действия, например, сохранять файл или выполнять пакетное задание.

Заключительный шаг - количественная оценка неполадок - преследует две цели. Первая – это позволить вам воспроизвести проблему, что позже поможет вам определить, решена ли проблема или нет. Вторая цель – получить больше подробностей о действиях пользователя, что поможет вам определить первопричину. Например, после того, как вы узнали, что задача, которая раньше выполнялась 5 минут, теперь выполняется час, вы можете осведомиться о характере этой задачи. В результате может выясниться, что задача получает информацию из

базы данных, расположенной на другом сервере, который необходимо будет включить в сферу поиска первопричины неисправности.

## Определение первопричины неисправности

Определение первопричины включает в себя использование команд, изученных в разделе [Измерение степени загрузки ресурсов](#), позволяющих найти источник проблемы. Для этого вы должны исследовать ресурсы, такие как центральный процессор, память, диск и сеть. В идеальном случае вы сможете собирать данные непосредственно в момент наступления проблемы с помощью инструментов реального времени, таких как `vmstat`, `iostat` и `top`. Если это невозможно, вам может помочь какая-нибудь утилита, собирающая статистику за прошедшие периоды, например утилита `sar`.

Если проблема связана с ресурсами, то возможны два варианта: либо один или более ресурсов системы будет загружен на 100%, и в этом случае причина должна быть очевидна, либо вы не обнаружите никакой явной перегрузки.

В последнем случае вы должны обратиться к *базовому уровню*. Базовый уровень – это набор эталонных данных, который вы можете использовать для сравнения наблюдаемых значений с "нормальными" значениями. Базовым уровнем для вашей системы может являться набор диаграмм или заархивированные отчеты утилиты `sar`, которые были созданы в период нормальной работы системы. Базовый уровень пригодится и позже, когда вы будете изучать прогнозирование потребностей в системных ресурсах.

По мере использования административных инструментов у вас начнет вырисовываться картина первопричины неисправности. Может оказаться, что на вашем почтовом сервере застряло сообщение, заблокировав обработку остальных сообщений. А возможно, какое-либо пакетное задание полностью загружает центральный процессор.

На этом этапе необходимо точно убедиться, что вы правильно определили первопричину возникшей проблемы. Приложение, генерирующее огромный файл журнала событий, могло привести к тому, что на диске закончилось свободное место. Если вы неправильно решите, что первопричиной оказался файл журнала событий, и решите удалить его, диск снова может оказаться заполнен спустя некоторое время.

## Внесение исправлений

Часто имеется несколько путей решения проблемы. Возьмем, например, пакетное задание, потребляющее все ресурсы центрального процессора. Если вы удалите задачу с помощью команды `kill`, пользователь, запустивший ее, вероятно, потеряет все свои данные, хотя остальные пользователи и получат в свое распоряжение ресурсы сервера. Вместо этого можно понизить приоритет процесса, чтобы высвободить часть ресурсов процессора для других задач. Как правило, выбор решения делается на ваше собственное усмотрение в зависимости от потребностей бизнеса и важности ситуации.

## Оценка результатов

После того как вы предприняли действия по исправлению ситуации, необходимо вернуться и проверить, решена ли проблема. Доставляются ли электронные письма незамедлительно? Могут ли пользователи войти в систему? Если нет, нужно вернуться на шаг назад и посмотреть на первопричину снова – возможно, вы найдете дополнительный способ решения проблемы, который можно использовать. Если ваши действия не решили проблему, также необходимо убедиться, что в результате ситуация не усугубилась!

После того как проблема решена, определите, нужно ли предпринимать какие-либо дополнительные долгосрочные меры. Возможно, стоит ли рассмотреть возможность установки диска большего объема или переноса пакетного задания пользователя на другую машину. Если на машине выполняются неопознанные процессы, стоит выполнить более

тщательную проверку безопасности сервера с целью убедиться, что он не используется злоумышленником.

## Более сложные проблемы

Некоторые проблемы производительности очевидны. Пользователь жалуется на то, что что-то работает медленно, вы запускаете команду `top`, видите ненужный процесс, который загружает процессор, завершает его выполнение, и система приходит в нормальное состояние. Ваш босс хвалит вас за хорошую работу, повышает вам зарплату и отпускает вас в этот день домой пораньше (ну, может быть, последняя часть выдумана).

Но что если проблема не столь очевидна? Иногда проблемы обусловлены не единственным фактором, а симптом может быть вызван чем-то таким, что на первый взгляд может показаться не относящимся к делу.

### Swap-спираль

Память работает быстро, и, как правило, в системе установлено достаточное ее количество. Тем не менее иногда приложению требуется больше памяти, чем имеется в системе, или же несколько процессов сообща расходуют всю имеющуюся память. В таких случаях используется виртуальная память. Ядро выделяет область на диске и выгружает в нее страницы резидентной памяти так, чтобы активные приложения могли использовать ее. Когда выгруженная на диск память требуется приложению, она загружается обратно в ОЗУ; при этом из ОЗУ на диск могут выгружаться другие страницы памяти, чтобы освободить место для загружаемых данных.

Проблема этого механизма заключается в том, что скорость диска очень мала. Если вы обращаетесь к виртуальной памяти кратковременно, вы можете не заметить этого. Однако когда система начинает активно выгружать память на диск с целью освободить ОЗУ под нужды других задач, у вас появляются проблемы. Вы обнаруживаете, что количество дисковых операций ввода/вывода стремительно увеличивается, и кажется, что система не реагирует на ваши запросы. На самом деле, система не отвечает потому, что ваши приложения ожидают, пока их память будет перемещена с диска в ОЗУ.

Администраторы UNIX называют эту ситуацию *swap-спиралью* (или иногда более мрачно – *смертельной swap-спиралью*). В итоге система застопоривается, так как диски работают на пределе, пытаясь выгружать память из ОЗУ на диск и обратно. Если устройство для подкачки расположено на том же физическом диске, что и данные, дела обстоят еще хуже. После того как ваше приложение обращается к центральному процессору и отправляет запрос ввода/вывода, ему приходится ожидать завершения обработки подкачки еще дольше.

Явным признаком swap-спирали является абсурдно долгое время отклика системы на любое действие, даже на выполнение простейшей команды `uptime`. Кроме того, вы также увидите высокое значение средней загрузки, поскольку множество процессов находятся в очереди на выполнение вследствие затора в работе системы. Чтобы отличить эту проблему от проблемы, связанной с высокой загруженностью центрального процессора, вы можете запустить команду `top` и посмотреть, насколько сильно загружен процессор, или запустить команду `vmstat` и посмотреть, насколько активно используется подкачка. Как правило, решение заключается в том, чтобы поочередно завершать процессы до тех пор, пока система не придет в нормальное состояние; иногда в зависимости от характера проблемы ее можно просто переждать.

### Нехватка места на диске

От приложений не требуется выполнения проверки на предмет возникающих ошибок. Многие приложения созданы в предположении, что все обращения к диску выполняются быстро и точно. Переполнение дискового раздела часто приводит к тому, что приложения начинают работать непредсказуемым образом. Например, приложение может

занять все доступные ресурсы центрального процессора, пытаясь выполнить операцию снова и снова, не осознавая при этом, что продолжение работы невозможно. Чтобы посмотреть, что делает приложение в данный момент (если оно использует системные вызовы), можно воспользоваться командой `strace`.

В других случаях приложения просто перестают работать. Если Web-приложение не может получить доступ к базе данных, оно может возвращать пустые страницы.

Самый быстрый способ проверить, не заключается ли проблема в нехватке дискового пространства – это войти в систему и проверить доступное место с помощью команды `du`.

### Заблокированные операции ввода/вывода

Когда процесс посылает запрос на выполнение какой-либо операции ввода/вывода, ядро переводит процесс в состояние сна (sleep) до тех пор, пока запрос ввода/вывода не будет возвращен. Если с диском происходит что-то неординарное (например, вследствие возникновения swap-спирали, сбоя в работе диска или сетевого сбоя в сетевых файловых системах), в режим сна одновременно переводится множество приложений.

В состоянии сна процесс может быть прерываемым (interruptible) или неинтерпретируемым (uninterpretable). В первом случае можно завершить работу процесса, послав сигнал на завершение его работы, во втором случае это сделать невозможно. Состояние можно посмотреть с помощью команды `ps aux`. В листинге 12 показаны два процесса, один из которых находится в состоянии прерываемого сна, а другой – в состоянии неинтерпретируемого сна.

### Листинг 12. Два процесса в состоянии сна

```
apache  26575  0.2 19.6 132572 50104 ?          S    Feb13   3:43 /usr/sbin/httpd
root    8381  57.8  0.2   3844    532 pts/1      D    20:46   0:37 dd
```

Первый процесс в листинге 12, `httpd`, находится в состоянии прерываемого сна, о чем говорит буква `S` после знака вопроса. Второй процесс, `dd`, находится в состоянии неинтерпретируемого сна. Состояния неинтерпретируемого сна чаще всего связаны с доступом к жесткому диску, тогда как состояния прерываемого сна относятся к относительно длительным операциям, например операциям NFS и сокетов.

Если вы обнаружили высокую среднюю загрузку (означающую, что в очереди на выполнение находится множество процессов) и множество процессов в состоянии неинтерпретируемого сна, возможно, проблема связана с вводом/выводом жесткого диска, - в результате либо сбоя в работе устройства, либо слишком большого количества одновременных операций записи/чтения на диск.

### Оценка потребностей в ресурсах

В этом разделе описывается материал по теме 306.3 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 2.

Из этого раздела вы узнаете, как:

- Устанавливать потребности в ресурсах
- Детализировать потребности приложений
- Определять потребности приложений в памяти и ресурсах центрального процессора
- Проводить полный анализ, исходя из потребностей отдельных приложений

Устранение текущих проблем является ключевой задачей системного администратора. Другая задача – это анализ текущей работы системы с целью предсказания ограничений на количество ресурсов в будущем и разрешения этой ситуации прежде, чем она выльется в проблему. В этом разделе рассматривается анализ текущих потребностей, а в следующем – прогнозирование использования ресурсов в будущем, исходя из материалов текущего раздела.

Вы можете использовать два подхода к анализу текущих потребностей: измерять текущие потребности в течение некоторого периода времени (наподобие базового уровня), или моделировать систему и задавать ряд параметров, которые приводят эту модель в состояние, соответствующее текущему состоянию системы. Первый подход проще и позволяет получить достаточно правдоподобную информацию. Второй подход точнее, но более трудоемок. Настоящее преимущество моделирования проявляется тогда, когда вам необходимо спрогнозировать поведение системы в будущем. Когда у вас имеется модель вашей системы, вы можете изменить определенные параметры, отражающие ее будущее развитие, и посмотреть, как изменится производительность.

На практике оба этих подхода используются совместно. В некоторых случаях построение модели конкретной системы оказывается слишком сложным, поэтому результаты измерений являются единственной основой для прогнозирования требований к ресурсам в будущем. С другой стороны, для построения моделей все-равно необходимы измерения.

### **Моделирование поведения системы**

Активность системы может быть смоделирована в виде серии *очередей*. Очередь – это конструкция, которая принимает на вход запросы и удерживает их до тех пор, пока ресурс не станет доступен. Как только ресурс становится доступен, задание выполняется и покидает очередь.

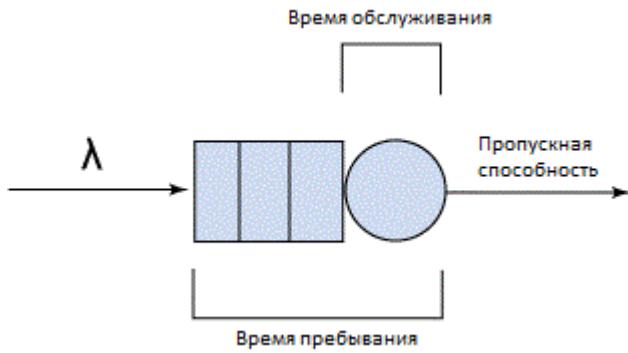
Несколько очередей можно объединить вместе в более объемную систему. Диск может быть смоделирован как очередь, в которой запросы поступают в буфер. Когда обслуживание запроса может быть выполнено, запрос передается к диску. Обычно запрос поступает от центрального процессора, который является одним простым ресурсом с несколькими задачами, конкурирующими за его использование. Изучением очередей и их приложений занимается *теория очередей*.

Книга *Analyzing Computer System Performance with Perl::PDQ* (ссылку вы можете найти в разделе [Ресурсы](#)) знакомит вас с теорией очередей и показывает, как смоделировать компьютерную систему в виде серии очередей. Далее в этой книге описывается библиотека языка C, которая называется PDQ, и сопутствующий интерфейс языка Perl, позволяющие вам определять и использовать очереди для получения оценки производительности.

### **Знакомство с очередями**

На рисунке 4 изображена простая очередь. Запрос поступает слева и входит в очередь. Как только запросы обрабатываются кругом, они покидают очередь. Прямоугольные блоки слева от круга означают объекты, поставленные в очередь.

**Рисунок 4. Простая очередь**



Поведение очереди характеризуется параметрами времени, частоты и размеров. *Частота поступления* обозначена буквой *лямбда* ( $\lambda$ ) и обычно выражается количеством запросов в секунду. Значение  $\lambda$  можно определить, выполнив наблюдения за вашей системой в течение подходящего интервала времени и посчитав поступления запросов. Подходящий интервал должен по меньшей мере в сто раз превышать *время обслуживания*, которое представляет собой продолжительность обработки запроса. *Время пребывания* – это общее время, в течение которого запрос находился в очереди, включая время, затраченное на его обработку.

Частота поступления описывает частоту, с которой объекты поступают в очередь, а *пропускная способность* определяет частоту, с которой объекты покидают очередь. В более сложной системе *узловая пропускная способность* определяет пропускную способность отдельного узла очереди, а *системная пропускная способность* относится ко всей системе в целом.

Размер буфера в большинстве случаев не имеет значения, поскольку буфер будет иметь ограниченный и предсказуемый размер, пока выполняются следующие условия:

- Размер буфера достаточно большой для обработки поставленных в очередь объектов.
- Очередь не увеличивается безгранично.

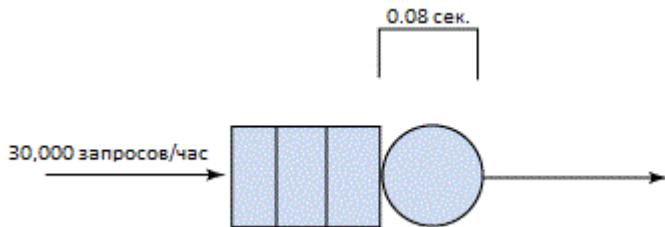
Второе условие является наиболее важным. Если очередь может отправлять запросы с частотой один запрос в секунду, но запросов, поступающих за одну секунду в очередь, больше, то очередь будет увеличиваться безгранично. В реальности частота поступления будет колебаться, но при анализе производительности нас интересует *устойчивое состояние*, поэтому используются средние значения. Возможно, что в какой-то момент в очередь попадут 10 запросов в секунду, а в другой момент – ни одного. До тех пор, пока среднее значение меньше, чем один запрос в секунду, очередь будет иметь конечную длину. Если среднее значение частоты поступления превышает частоту, с которой запросы выходят из очереди, длина очереди будет продолжать расти и никогда не достигнет устойчивого состояния.

Очередь, изображенная на рисунке 4, называется *открытой очередью*, поскольку в нее входит неограниченное число запросов, и им не обязательно возвращаться назад после своей обработки. *Закрытая очередь* возвращает информацию на вход; в системе находится ограниченное число запросов. После того, как запросы были обработаны, они возвращаются в очередь поступления.

Классическим примером очереди является продовольственный магазин. Количество людей, встающих в очередь к кассе, поделенное на период измерения – это частота поступления. Количество рассчитавшихся и покидающих очередь людей, поделенное на период измерения – это пропускная способность. Среднее время, которое затрачивает кассир на обслуживание одного покупателя – это время обслуживания. Среднее время, в течение которого покупатель стоит в очереди и ждет расчета с кассиром, плюс время обслуживания этого покупателя кассиром – это время пребывания.

Чтобы перейти к рассмотрению PDQ, рассмотрим следующий сценарий. Web-служба получает 30,000 запросов за один час. Наблюдая некоторое время за ненагруженной системой, мы установили, что время обслуживания составляет 0.08 секунды. На рисунке 5 это изображено в виде очереди.

**Рисунок 5. Web-служба, смоделированная в виде очереди**



Какую информацию может предоставлять PDQ? В листинге 13 приведена требуемая программа PDQ и ее вывод данных.

**Листинг 13. Программа PDQ и ее вывод данных**

```

#!/usr/bin/perl
use strict;
use pdq;
# Observations
my $arrivals = 30000; # requests
my $period = 3600; # seconds
my $serviceTime = 0.08; # seconds

# Derived
my $arrivalRate = $arrivals / $period;
my $throughput = 1 / $serviceTime;
# Sanity check -- make sure arrival rate < throughput
if ($arrivalRate > $throughput) {
    die "Arrival rate $arrivalRate > throughput $throughput";
}

# Create the PDQ model and define some units

pdq::Init("Web Service");
pdq::SetWUnit("Requests");
pdq::SetTUnit("Seconds");
# The queuing node
pdq::CreateNode("webservice", $pdq::CEN, $pdq::FCFS);

# The circuit
pdq::CreateOpen("system", $arrivalRate);

# Set the service demand

pdq::SetDemand("webservice", "system", $serviceTime);

# Run the report
pdq::Solve($pdq::CANON);
pdq::Report();

..... output ..
*****
***** Pretty Damn Quick REPORT *****
*****
*** of : Sat Feb 16 11:24:54 2008 ***

```

```
*** for: Web Service ***  
*** Ver: PDQ Analyzer v4.2 20070228***  
*****  
*****
```

```
*****  
***** PDQ Model INPUTS *****  
*****
```

Node	Sched	Resource	Workload	Class	Demand
-----	-----	-----	-----	-----	-----
CEN	FCFS	webservice	system	TRANS	0.0800

#### Queueing Circuit Totals:

Streams:	1
Nodes:	1

#### WORKLOAD Parameters

Source	per Sec	Demand
-----	-----	-----
system	8.3333	0.0800

```
*****  
***** PDQ Model OUTPUTS *****  
*****
```

#### Solution Method: CANON

```
***** SYSTEM Performance *****
```

Metric	Value	Unit
-----	-----	-----
Workload: "system"		
Mean Throughput	8.3333	Requests/Seconds
Response Time	0.2400	Seconds

#### Bounds Analysis:

Max Demand	12.5000	Requests/Seconds
Max Throughput	12.5000	Requests/Seconds

```
***** RESOURCE Performance *****
```

Metric	Resource	Work	Value	Unit
-----	-----	-----	-----	-----
Throughput	webservice	system	8.3333	Requests/Seconds

Utilization	webservice	system	66.6667	Percent
Queue Length	webservice	system	2.0000	Requests
Residence Time	webservice	system	0.2400	Seconds

Листинг 13 начинается с типичной для UNIX строки, определяющей интерпретатор для оставшейся части программы. Первые две строки Perl-кода указывают на использование модулей `pdq` и `strict`. Модуль `pdq` предоставляет функции PDQ, тогда как модуль `strict` налагает жесткие ограничения на синтаксис Perl, позволяя избежать ряда ошибок при написании кода.

В следующем разделе листинга 13 определены переменные, связанные с наблюдениями за системой. С помощью этой информации в следующем разделе вычисляются частота поступления и пропускная способность. Последняя величина является обратной к времени обслуживания – если вы можете обработать один запрос за  $N$  секунд, значит, вы можете обработать  $1/N$  запросов в секунду.

## Установка PDQ

Дистрибутив PDQ можно загрузить с Web-сайта автора (обратитесь к разделу [Ресурсы](#)). Распакуйте его во временную папку с помощью команды `tar -xzf pdq.tar.gz` и перейдите во вновь созданную папку, выполнив команду `cd pdq42`. После этого запустите команду `make`, чтобы выполнить компиляцию исходного кода на языке C и модуля Perl. Наконец, выполните команду `cd perl5` и из этой папки запустите команду `./setup.sh`, чтобы завершить компоновку модуля Perl и установить его в вашу системную директорию.

Первый тест на работоспособность проверяет, что длина очереди ограничена. Хотя большинство функций PDQ и сообщают об ошибках, автор модуля рекомендует выполнять явную проверку. Если количество поступающих в секунду запросов больше, чем количество запросов, покидающих очередь за это же время, программа завершается с ошибкой.

В оставшейся части программы выполняются прямые вызовы функций PDQ. Сначала выполняется инициализация модуля и указывается название модели. Затем временной модуль и рабочие модули настраиваются таким образом, чтобы информация в отчетах выводилась в нужном вам виде.

Каждая очередь создается с помощью функции `CreateNode`. В листинге 13 создана очередь типа CEN (центр организации очереди, в отличие от узла задержки, который не выполняет никакой работы) с именем `webservice` (эти имена являются метками и помогают вам разобраться в итоговом отчете). Это стандартная очередь типа FIFO (first in, first out – первый вошел, первый вышел), которая в терминах PDQ называется *first-come first-served* (первый вошел, первый обработан).

Затем происходит вызов функции `CreateOpen` с целью определения *цепи* (совокупность очередей). Частота поступления в цепь уже рассчитана. Наконец, с помощью функции `SetDemand` для очереди задается нагрузка. Функция `SetDemand` определяет время, необходимое на завершение конкретной рабочей нагрузки (очередь в составе цепи).

Наконец, цепь решается с помощью функции `Solve`, и с помощью функции `Report` создается отчет. Обратите внимание на то, что PDQ берет вашу модель, преобразует ее в ряд уравнений и затем решает их. PDQ не имитирует модель каким-либо образом.

Теперь рассмотрим, как интерпретировать полученные результаты. Отчет начинается с заголовка и сводной информации о модели. Раздел **WORKLOAD Parameters** содержит более интересную для нас информацию. Время обслуживания в цепи составляет 0.08

секунды, в соответствии с заданным ранее значением. Значение per second - это частота поступления.

В разделе **SYSTEM performance** рассчитывается производительность системы в целом. Цепь выдержала частоту поступления, равную 8.3333 запросам в секунду. Время отклика (response time) составило 0.24 секунды (более подробно об этом позже); это значение включает в себя время обслуживания, равное 0.08 секундам, и время нахождения запроса в очереди. Максимальная производительность цепи оценивается в 12.5 запросов в секунду.

Посмотрев на очередь более внимательно, вы можете увидеть, что она используется на 66.6667%. Средняя длина очереди равна двум запросам. Это означает, что перед входящим запросом в очереди может содержаться еще два запроса, плюс запрос, обрабатываемый в данный момент. С учетом времени обработки одного запроса, равного 0.08 секундам, среднее время ожидания составляет 0.24 секунды, о чем было сообщено ранее.

Эту модель можно детализировать до компонентов Web-службы. Вместо использования одной очереди, представляющей собой Web-службу, вы могли бы использовать отдельную очередь для обработки запроса, отдельную очередь для доступа к базе данных и отдельную очередь для подготовки ответа. Если ваша модель правильно построена, производительность системы должна оставаться той же самой, просто в этом случае вы получите более подробную информацию о внутренней работе Web-службы. С этого уровня вы можете использовать подход "что если" и смоделировать применение более быстрой базы данных или нескольких Web-серверов, чтобы посмотреть, какой эффект получится в результате внесенных изменений. Разделение модели на отдельные ресурсы поможет вам понять, какая конкретная очередь является узким местом, и сколько ресурсов у вас имеется в запасе.

В листинге 13 приведен элементарный пример использования библиотек PDQ. Чтобы узнать, как строить более сложные модели, прочитайте книгу *Analyzing Computer System Performance with Perl::PDQ* (ссылку вы можете найти в разделе [Ресурсы](#)).

## Определение будущих потребностей в ресурсах

В этом разделе описывается материал по теме 306.4 экзамена на профессионала Linux высокого уровня (LPIC-3) 301. Эта тема обладает весом 1.

Из этого раздела вы узнаете, как:

- Прогнозировать момент исчерпания пропускной способности конфигурации
- Отслеживать темпы роста использования пропускной способности
- Строить графики тенденций использования пропускной способности

В [предыдущем разделе](#) вы познакомились с библиотекой PDQ и примером отчета. В отчете были показаны рассчитанные значения использования и максимальной загрузки очереди, а также системы в целом. Вы можете использовать этот же метод для прогнозирования момента исчерпания пропускной способности конфигурации. Кроме того, вы можете использовать графики для отображения темпов роста системы с течением времени, а также для расчета того момента, когда пропускная способность достигнет предела.

## Еще о PDQ

В листинге 14 показана та же Web-служба, что и в [листинге 13](#), но здесь она разделена на две очереди: одна очередь отображает время центрального процессора Web-сервера, затрачиваемое на обработку запроса и формирование ответа, а другая – время ожидания ответа от базы данных на поступивший запрос.

#### Листинг 14. Новая программа PDQ для примера Web-службы

```
#!/usr/bin/perl
use strict;
use pdq;
# Observations
my $arrivals = 30000; # requests
my $period = 3600; # seconds

# Derived
my $arrivalRate = $arrivals / $period;

# Create the PDQ model and define some units

pdq::Init("Web Service");
pdq::SetWUnit("Requests");
pdq::SetTUnit("Seconds");

# The queuing nodes
pdq::CreateNode("dblookup", $pdq::CEN, $pdq::FCFS);
pdq::CreateNode("process", $pdq::CEN, $pdq::FCFS);

# The circuit
pdq::CreateOpen("system", $arrivalRate);

# Set the service demand

pdq::SetDemand("dblookup", "system", 0.05);
pdq::SetDemand("process", "system", 0.03);

# Solve
pdq::Solve($pdq::CANON);
pdq::Report();
```

Код в листинге 14 добавляет в систему еще одну очередь. Общее время обслуживания составляет все те же 0.08 секунды, из которых 0.05 секунды затрачены на поиск в базе данных и 0.03 секунды – на обработку данных центральным процессором. В листинге 15 показан сгенерированный отчет.

#### Листинг 15. Отчет программы PDQ из листинга 14

```
*****
***** Pretty Damn Quick REPORT *****
*****
*** of : Sun Feb 17 11:35:35 2008 ***
*** for: Web Service ***
*** Ver: PDQ Analyzer v4.2 20070228 ***
*****
*****
```

```
*****
***** PDQ Model INPUTS *****
*****
```

Node	Sched	Resource	Workload	Class	Demand
------	-------	----------	----------	-------	--------

CEN	FCFS	dblookup	system	TRANS	0.0500
CEN	FCFS	process	system	TRANS	0.0300

#### Queueing Circuit Totals:

Streams: 1  
 Nodes: 2

#### WORKLOAD Parameters

Source	per Sec	Demand
-----	-----	-----
system	8.3333	0.0800

\*\*\*\*\*
 \*\*\*\*\* PDQ Model OUTPUTS \*\*\*\*\*
 \*\*\*\*\*

#### Solution Method: CANON

\*\*\*\*\* SYSTEM Performance \*\*\*\*\*

Metric	Value	Unit
-----	-----	-----
Workload: "system"		
Mean Throughput	8.3333	Requests/Seconds
Response Time	0.1257	Seconds

#### Bounds Analysis:

Max Demand	20.0000	Requests/Seconds
Max Throughput	20.0000	Requests/Seconds

\*\*\*\*\* RESOURCE Performance \*\*\*\*\*

Metric	Resource	Work	Value	Unit
-----	-----	-----	-----	-----
Throughput	dblookup	system	8.3333	Requests/Seconds
Utilization	dblookup	system	41.6667	Percent
Queue Length	dblookup	system	0.7143	Requests
Residence Time	dblookup	system	0.0857	Seconds
Throughput	process	system	8.3333	Requests/Seconds
Utilization	process	system	25.0000	Percent
Queue Length	process	system	0.3333	Requests
Residence Time	process	system	0.0400	Seconds

Взгляните на часть отчета с выходными данными и обратите внимание на то, что среднее

время отклика уменьшилось по сравнению с листингом 13, а максимальное количество запросов в секунду возросло с 12.5 до 20. Это произошло благодаря тому, что в новой модели предусмотрена *конвейерная обработка*. Пока происходит передача одного запроса в базу данных, другой запрос может обрабатываться центральным процессором. Это невозможно было рассчитать в предыдущей модели, поскольку в ней использовалась всего одна очередь.

Еще важнее то, что вы можете видеть, что база данных загружена на 42%, а центральный процессор – только на 25%. Таким образом, именно база данных является тем ресурсом, который будет исчерпан первым при высокой загрузке системы.

Увеличьте частоту поступления запросов до 60 000 в час, и вы увидите, что среднее время отклика возрастет до 0.36 секунд, а загрузка базы данных составит 83%. Также вы увидите, что 0.30 секунды из 0.36 уходит на ожидание ответа от базы данных. Таким образом, время обработки запросов можно улучшить, ускорив доступ к базе данных.

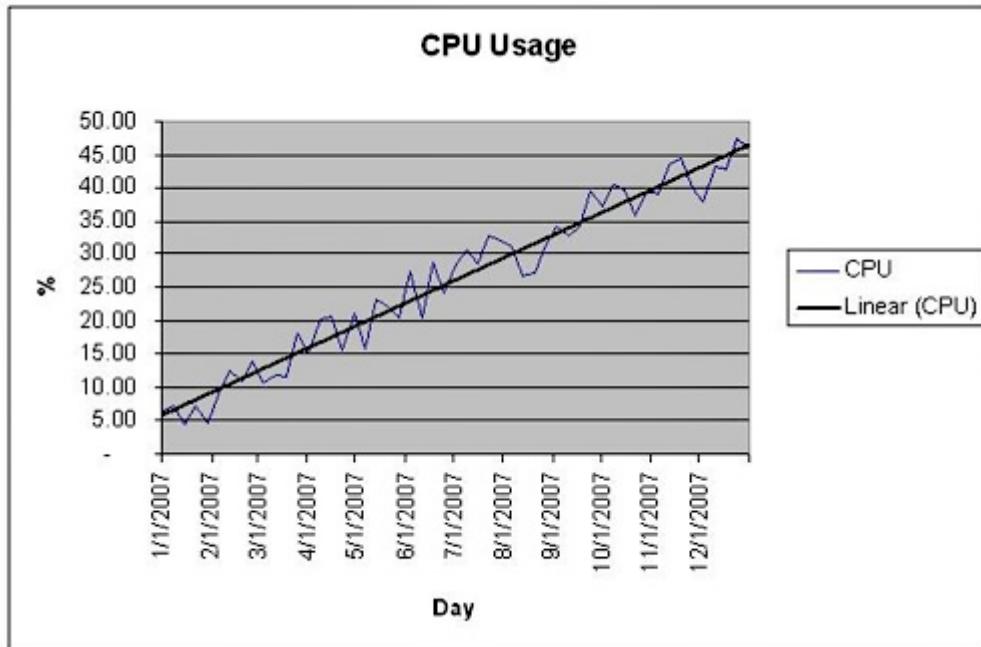
Вы можете определить максимальную производительность различными способами.

Производительность системы составляет 100% при 20 запросах в секунду (верхняя часть отчета). Вы также можете определить производительность в терминах времени среднего отклика. Приблизительно при 15 запросах в секунду время отклика превысит четверть секунды. Если ваша задача – сохранить время отклика меньше этого значения, то на этом этапе ваша система исчерпает ресурсы, хотя у вас и останется аппаратный запас мощностей.

### Использование графиков для анализа производительности

Графики предоставляют отличную возможность для отображения статистической информации. Вы можете построить график за продолжительный период времени, например, за 6 месяцев или год, и получить представление о темпах роста. На рисунке 6 представлен график использования ресурсов процессора сервером приложений в течение года. Величины средней ежедневной загрузки были загружены в электронную таблицу и представлены в виде графика. На рисунке также была добавлена линия тенденции изменения, позволяющая увидеть рост.

Рисунок 6. График использования центрального процессора сервера



На основании этого графика вы можете планировать загрузку системы в будущем (исходя из предположения, что темпы роста остаются постоянными). На рисунке 6 рост составляет приблизительно 10% каждые 3 месяца. Влияние загрузки очередей более заметно при более

высокой загрузке, поэтому может оказаться, что обновление системы целесообразно произвести, не дожидаясь, пока загрузка достигнет 100%.

## Как строить графики

Построение графиков в электронной таблице плохо подходит для анализа множества серверов, для каждого из которых имеется свой набор данных измерений. Один из методов предусматривает загрузку выходных данных **Sar** в утилиту построения графиков наподобие GUPLOT. Также можно обратить внимание на другие доступные инструменты построения графиков, многие из которых распространяются по лицензии open source. В их числе - ряд утилит, основанных на пакете RRDTool.

RRDTool – это набор программ и библиотек, которые помещают данные в кольцевую базу данных формата RRD (Round-robin database). База данных RRD непрерывно архивирует данные по мере их поступления, поэтому вы можете иметь ежечасные данные за последний год и пятиминутные средние значения за неделю. Размер базы данных формата RRD всегда остается постоянным, поскольку старые данные удаляются. Пакет RRDTool также содержит инструменты для построения графиков.

В разделе [Ресурсы](#) приведено несколько ссылок на утилиты для построения графиков.

## Что включать в графики

В графики следует включать любую информацию, которая важна для вашей службы, а также все, что потенциально может использоваться для принятия решений. Графики также помогают понять, что происходило в прошлом, так что вы можете строить графики таких величин, как, например, скорость вентиляторов. Тем не менее обычно ваше внимание будет сфокусировано на статистике загрузки центрального процессора, памяти, диска и сети. При возможности стройте графики времени отклика от работающих служб. Это поможет вам не только принимать лучшие решения для организации работы пользователей, но и разрабатывать различные модели вашей системы.

## Заключение

Из этого руководства вы узнали, как измерять и анализировать производительность системы. Также вы узнали, как использовать полученные результаты для решения возникающих проблем.

Linux предоставляет вам подробную информацию, касающуюся состояния системы. Такие утилиты, как **vmstat**, **iostat** и **ps**, показывают информацию в реальном времени. Утилиты наподобие **Sar** показывают статистику, накопленную за определенный период времени. Помните, что когда вы используете утилиты **vmstat** и **iostat**, первое выводимое значение не является значением в реальном времени!

При диагностике неисправностей первым делом следует выявить симптомы, которые помогут вам разобраться в проблеме и впоследствии проверить, решена ли она. После этого постарайтесь измерить загрузку ресурсов в тот момент, когда проблема проявляется (если это возможно), чтобы установить ее источник. После того, как вы найдете путь решения проблемы, реализуйте его и оцените результаты.

Модуль PDQ языка Perl позволяет вам решать проблемы с помощью очередей. После того, как вы смоделируете вашу систему в виде набора очередей, вы можете написать Perl-скрипт с использованием функций PDQ. Затем вы можете использовать эту модель для расчета как текущих, так и будущих потребностей в системных ресурсах.

Предсказать темпы роста можно как с помощью моделей, так и с помощью графиков. В идеальном случае следует использовать оба метода, сравнивая полученные результаты.

Это руководство завершает серию руководств по подготовке к экзамену LPIC 3. Я желаю вам успеха при сдаче экзамена и надеюсь, что эта серия руководств окажется для вас полезной.

## Ресурсы

### Научиться

- Оригинал руководства "[LPI exam 301 prep, Topic 305: Integration and migration](#)" (EN).
- Изучите предыдущее руководство в серии 301 - "[Подготовка к экзамену LPI 301: Тема 305. Интеграция и миграция](#)" (developerWorks, апрель 2008 г.) или [все руководства в серии 301](#) (EN).
- Чтобы познакомиться с основами Linux и подготовиться к сертификации в качестве системного администратора, ознакомьтесь со всей [серий руководств для подготовки к экзаменам LPI](#).
- В [программе LPIC](#) (EN) вы можете найти перечни заданий, примеры вопросов и подробные цели для трех уровней сертификации системных администраторов Linux института Linux Professional Institute.
- Прочтите статьи [UNIX Load Average Part 1: How It Works](#) и [UNIX Load Average Part 2: Not Your Average Average](#) Нейла Гюнтера (Neil Gunther), который управляет компанией [Performance Dynamics Company](#) и является автором [нескольких других статей](#) и книг, озаглавленных [Analyzing Computer System Performance with Perl::PDQ](#) и [Guerrilla Capacity Planning: A Tactical Approach to Planning for Highly Scalable Applications and Services](#) (EN).
- Статья Шона "[Easy system monitoring with SAR](#)" (EN) (developerWorks, февраль 2006 г.) знакомит вас с SAR. Эта статья была написана на примере Solaris, поэтому параметры командной строки могут в чем-то отличаться, но теория применима к любой системе.
- В руководстве Шона "[Expose Web performance problems with the RRDTool](#)" (EN) (developerWorks, март 2006 г.) показано, как следить за производительностью Web-сайта и выводить результаты в графической форме с помощью RRDTool.
- Список [часто задаваемых вопросов по sysstat](#) (EN) поможет вам найти ответы на вопросы по SAR и сопутствующим инструментам.
- В [разделе Linux сайта developerWorks](#) можно найти дополнительные ресурсы для разработчиков Linux, а также [самые популярные среди наших читателей статьи и руководства](#) (EN).
- Следите на последними новостями на портале [Web-трансляций и технических мероприятий developerWorks](#) (EN).

### Получить продукты и технологии

- Загрузите [исходные файлы PDQ](#) и инструкции по ее установке в различных операционных системах (EN).
- [RRDTool](#), основа большинства open source систем мониторинга сети, является ответвлением знаменитого пакета [Multi Router Traffic Grapher](#). RRDTool может использоваться как часть другого пакета или ваших собственных скриптов.
- [Cacti](#) – это один из лучших open source пакетов, предназначенных для мониторинга. Первоначально он был разработан для работы с сетевыми устройствами, но затем его функционал был расширен для выполнения множества системных задач. Cacti активно

поддерживается сообществом пользователей, которые всегда помогут вам на различных форумах.

- ZABBIX – это еще один о заслуживающий внимания open source пакет для мониторинга системы.
- Посетите раздел ресурса developerWorks, посвященный Tivoli, чтобы получить дополнительную информацию о предлагаемых IBM корпоративных системах и инструментах управления сетью, безопасностью и приложениями. В частности, решение Tivoli Composite Application Management помогает вам повысить производительность и готовность сложных современных бизнес-приложений, в том числе с использованием портальных и SOA-технологий.
- Используйте в своем следующем проекте разработки для Linux ознакомительные версии программного обеспечения IBM, которые можно скачать непосредственно с developerWorks (EN).

# Изучаем Linux, 302 (смешанные среды): Перечень материалов для подготовки к экзамену LPI-302

*Перечень статей developerWorks для подготовки к экзамену LPI-302*

Трейси Бост, консультант и преподаватель, Свободный писатель

Родерик Смит (Roderick Smith), автор и консультант, IBM

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Это перечень материалов IBM developerWorks, которые помогут вам освоить и освежить основы работы с Linux в смешанной среде UNIX/Microsoft. Если вы решили стать сертифицированным системным администратором Linux, то эти статьи помогут подготовиться к сдаче экзамена 302 сертификационной программы института Linux Professional Institute (LPI). Перечень материалов структурирован в соответствии с 21 заданиями экзамена LPI-302, который необходимо пройти для получения специализации "Смешанные среды" программы LPIC-3.

[Больше статей из этой серии](#)

**Дата:** 07.06.2012

**Уровень сложности:** сложный

## Об этой серии

Как и многие другие программы, [программа сертификации Linux Professional Institute \(LPIC\)](#) предусматривает различные уровни сертификации, где для получения каждого последующего уровня необходимо обладать более глубокими знаниями и практическим опытом. Экзамен LPI-302 – это факультативный экзамен третьего уровня программы LPIC, требующий продвинутых знаний в области системного администрирования Linux.

Для получения [сертификата LPIC третьего уровня \(LPIC-3\)](#) необходимо успешно сдать два экзамена первого уровня (101 и 102), два экзамена второго уровня (201 и 202), а также базовый экзамен 301 третьего уровня (LPIC-3). Если вы получили сертификат третьего уровня, вы можете сдавать факультативные экзамены по определенным специализациям. По состоянию на ноябрь 2010 года доступны следующие факультативные экзамены:

- **302. Смешанные среды** (материалы из этого перечня)
- **303. Безопасность**
- **304. Виртуализация и высокая готовность**
- **306. Web и интранет (в разработке)**
- **305. Почта и обмен сообщениями (в разработке)**

Успешно сдав любой из этих экзаменов, вы получите соответствующую специализацию в дополнение к базовой программе LPIC-3.

Эта серия статей поможет вам подготовиться к сдаче экзамена по специализации "Смешанные среды", в которой подробным образом рассматривается управление клиентами и серверами Linux в гетерогенной среде Linux и Microsoft. Поскольку эта задача почти полностью осуществляется при помощи инструментов проекта Samba, то для успешной сдачи экзамена 302 вы должны научиться в совершенстве устанавливать, настраивать и устранять неисправности в работе этой службы.

Трудно найти организацию, которая совсем не использует Windows, поэтому даже если вы не

планируете сдавать экзамен LPI-302, эта серия статей может оказаться для вас полезной.

## Экзамен 302, тема 310: архитектура, принципы и схема работы

**Таблица 1. Цели темы 310 и их вес**

Статья на сайте developerWorks	Краткое описание цели экзамена
<a href="#">Изучаем Linux, 302 (смешанные среды): основные принципы</a>	<b>310.1 Основные принципы</b> Основные принципы и концепции протоколов Server Message Block (SMB)/Common Internet File System (CIFS), технологии совместного использования файлов и служб печати в смешанной среде. После изучения этой темы вы узнаете, как работают различные компоненты и функции протоколов SMB и CIFS, и где располагаются конфигурационные файлы Samba. <i>Вес:</i> 1
<a href="#">Изучаем Linux, 302 (смешанные среды): роли Samba</a>	<b>310.2 Роли Samba</b> Режимы безопасности и ключевые роли демонов Samba. Вы научитесь управлять демонами Samba, определять их роли и понимать, чем отличаются различные режимы безопасности Samba. <i>Вес:</i> 1
<a href="#">Изучаем Linux, 302 (смешанные среды): файлы базы данных Trivial Database</a>	<b>310.3 Файлы базы данных Trivial Database</b> Структура файлов базы данных Trivial Database (TDB), в которых Samba хранит информацию, а также поиск и устранение неисправностей. Вы должны научиться архивировать и восстанавливать TDB-файлы, определять признаки повреждений TDB-файлов, а также изменять содержимое TDB, используя различные инструменты Samba. <i>Вес:</i> 2

## Экзамен 302, тема 311: компиляция и установка Samba

**Таблица 2. Цели темы 311 и их вес**

Статья на сайте developerWorks	Краткое описание цели экзамена
<a href="#">Изучаем Linux, 302 (смешанные среды): конфигурирование и компиляция Samba из исходного кода</a>	<b>311.1 Конфигурирование и компиляция Samba из исходного кода</b> Действия, необходимые для компиляции Samba из исходного кода и разрешения внешних зависимостей. Вы научитесь определять ключевые пакеты и содержимое, определять и разрешать любые внешние зависимости (например, библиотеки), описывать структуру программного обеспечения Samba и использовать распространенные параметры компиляции Samba. <i>Вес:</i> 1
<a href="#">Изучаем Linux, 302</a>	<b>311.2. Установка и обновление Samba</b>

[\(смешанные среды\): установка и обновление Samba](#)

Процедуры установки и обновления Samba из исходного кода и двоичных пакетов. Вы научитесь устанавливать Samba из исходного кода и из пакетов Red Hat и Debian, а также обновлять уже установленный экземпляр Samba с сохранением текущей конфигурации.

*Вес:* 1

**Экзамен 302, тема 312: настройка и использование Samba**

**Таблица 3. Цели темы 312 и их вес**

<b>Статья на сайте developerWorks</b>	<b>Краткое описание цели экзамена</b>
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): настройка Samba</u></a>	<b>312.1 Настройка Samba</b> Настройка демонов Samba для выполнения разнообразных задач. Вы научитесь понимать структуру конфигурационных файлов и использовать различные переменные и параметры в них. Вы также научитесь настраивать журналирование и использовать другие инструменты для поиска, отладки и устранения неисправностей Samba. Наконец, вы узнаете о наиболее важных TCP- и UDP-портах, используемых протоколами SMB и CIFS. <i>Вес:</i> 6
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): файловые службы</u></a>	<b>312.2. Файловые службы</b> Создание и настройка общих файловых ресурсов в смешанной операционной среде. Вы научитесь управлять общими файловыми ресурсами, планировать миграцию файлового сервера, создавать сценарии для управления общими файловыми ресурсами, а также использовать инструменты командной строки Samba для проверки общих файловых ресурсов и работы с ними. <i>Вес:</i> 4
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): службы печати</u></a>	<b>312.3 Службы печати</b> Создание и управление службами печати в смешанной операционной среде. Создание и настройка общих принтеров для компьютеров Windows и серверов Samba, включая инсталляцию драйверов принтеров на клиентские машины. Настройка общего ресурса <code>print\$</code> с учетом вопросов безопасности. Интеграция Samba и системы CUPS (Common UNIX Printing System) и управление статистикой печати. <i>Вес:</i> 2
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): управление доменом</u></a>	<b>312.4 Управление доменом</b> Настройка и обслуживание основного и резервного контроллеров домена и управление клиентским доступом к домену. Вы научитесь создавать и обслуживать основной и резервный контроллеры домена и добавлять клиентов в домен. Также вы научитесь настраивать сценарии входа в систему, перемещаемые профили и системные политики.

*Bec:* 4

[Изучаем Linux, 302  
\(смешанные среды\):  
настройка SWAT](#)

**312.5 Настройка SWAT**

Установка, настройка и использование Samba Web Administration Tool (SWAT). Вы научитесь устанавливать и настраивать SWAT, знать его возможности и уметь использовать его для настройки Samba.

*Bec:* 1

[Изучаем Linux, 302  
\(смешанные среды\):  
локализация](#)

**312.6 Локализация**

Локализация кодов символов и кодовых страниц. Вы научитесь понимать и использовать преимущества локализации кодов символов и кодовых страниц, изменять и создавать соответствующие библиотеки преобразования кода, управлять пользователями, группами и компьютерами в программной среде, в которой используется язык, отличный от английского.

*Bec:* 1

**Экзамен 302, тема 313: управление пользователями и группами**

**Таблица 4. Цели темы 313 и их вес**

**Статья на сайте  
developerWorks**

[Изучаем Linux, 302  
\(смешанные среды\):  
управление учетными  
записями пользователей  
и групп](#)

**Краткое описание цели экзамена**

**313.1 Управление учетными записями пользователей и групп**

Управление учетными записями пользователей и групп в смешанной операционной среде. Вы научитесь управлять учетными записями пользователей и групп с помощью различных инструментов и понимать, как работает сопоставление пользователей и групп. Также вы научитесь устанавливать права владения файлами и директориями.

*Bec:* 4

[Изучаем Linux, 302  
\(смешанные среды\):  
проверка подлинности и  
авторизация](#)

**313.2 Проверка подлинности и авторизация**

Изучение различных механизмов проверки подлинности и настройка контроля доступа. Вы научитесь настраивать локальную базу данных паролей, узнаете о формате файла smbpasswd и научитесь выполнять синхронизацию паролей между Samba и другими системами. Также вы узнаете о других внутренних системах хранения паролей, научитесь интегрировать Samba с протоколом LDAP (Lightweight Directory Access Protocol) и работать со списками контроля доступа (ACL).

*Bec:* 8

[Изучаем Linux, 302  
\(смешанные среды\):  
Winbind](#)

**313.3 Winbind**

Установка и настройка службы Winbind. Вы научитесь настраивать сервер Linux для получения информации о каталоге домена Windows, включая интеграцию с PAM-модулем (Pluggable Authentication Module) и демоном nscd, и сопоставлять идентификаторы пользователей UNIX

и идентификаторы безопасности Microsoft.

*Bec: 2*

## Экзамен 302, тема 314: работа с CIFS, NetBIOS и Active Directory

**Таблица 5. Цели темы 314 и их вес**

<b>Статья на сайте developerWorks</b>	<b>Краткое описание цели экзамена</b>
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): интеграция протокола CIFS</u></a>	<b>314.1 Интеграция протокола CIFS</b> Изучение работы протокола CIFS в смешанной среде. Вы узнаете основные принципы, возможности и преимущества SMB/CIFS, а также научитесь монтировать удаленные общие файловые ресурсы CIFS на клиенте Linux. <i>Bec: 3</i>
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): NetBIOS и WINS</u></a>	<b>314.2 NetBIOS и WINS</b> Изучение принципов работы службы Network Basic Input/Output System (NetBIOS) и Windows Internet Naming System (WINS) и просмотра сетевого окружения. Вы подробно изучите принципы работы служб WINS и NetBIOS и узнаете, как работает система разрешения имен и просмотра сетевых ресурсов. Вы научитесь описывать роли локального главного обозревателя (local master browser) и доменного главного обозревателя (domain master browser) и настраивать Samba в качестве WINS –сервера. <i>Bec: 7</i>
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): интеграция с Active Directory</u></a>	<b>314.3 Интеграция с Active Directory</b> Интеграция серверов Linux в доменную среду Active Directory Domain Services (AD DS). Вы научитесь просматривать списки удаленных пользователей AD DS через LDAP, настраивать Samba в режиме безопасности Active Directory Services и описывать требования, предъявляемые AD DS к службе системы доменных имен (DNS). <i>Bec: 2</i>
<a href="#"><u>Изучаем Linux, 302 (смешанные среды): работа с клиентами Windows</u></a>	<b>314.4 Работа с клиентами Windows</b> Обеспечение взаимодействия клиентов Linux с клиентами Windows и настройка доступа для клиентов Windows к файловым службам и службам печати, запущенным на серверах Linux. Вы узнаете, как работают клиенты Windows и как с их помощью просматривать сетевое окружение и использовать сетевые файловые ресурсы и ресурсы печати. Вы также научитесь использовать утилиты <code>net</code> (в Windows) и <code>smbclient</code> (в Linux) для выполнения различных задач по работе с общими файлами и службами печати. <i>Bec: 4</i>

## Экзамен 302, тема 315: безопасность и производительность

Таблица 6. Цели темы 315 и их вес

Статья на сайте developerWorks	Краткое описание цели экзамена
<a href="#">Изучаем Linux, 302 (смешанные среды): управление доступом к файловой системе и общим ресурсам Linux</a>	<b>315.1 Управление доступом к файловой системе и общим ресурсам Linux:</b> Управление доступом к файловой системе Linux в смешанной операционной среде. Вы узнаете обо всех аспектах управления доступом к файлам и директориям UNIX и Windows, а также уметь управлять этими разрешениями с помощью командной строки и Samba. <i>Вес: 3</i>
<a href="#">Изучаем Linux, 302 (смешанные среды): безопасность Samba</a>	<b>315.2 Безопасность Samba</b> Защита Samba на уровне брандмауэра и с помощью конфигурационных файлов. Вы научитесь управлять доступом к серверу Samba и общим ресурсам Samba, выбирая для этого наилучшие способы, а также уметь использовать параметры конфигурации Samba, отвечающие за безопасность. <i>Вес: 2</i>
<a href="#">Изучаем Linux, 302 (смешанные среды): настройка производительности</a>	<b>315.3 Настройка производительности</b> Использование кластерных служб для балансировки нагрузки и обеспечения высокой готовности и настройка параметров Samba для лучшей производительности сервера и сети. Вы научитесь оценивать производительность Samba, оптимизировать использование оперативной памяти и повышать скорость передачи файлов. Кроме того, вы научитесь работать с различными утилитами командной строки Linux и Samba для просмотра сетевой конфигурации и параметров производительности. <i>Вес: 1</i>

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): A roadmap for LPI-302 \(EN\)](#).
- На Web-сайте [программы сертификации LPIC \(EN\)](#) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены цели экзамена [LPI 302](#).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI \(EN\)](#) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- В [разделе Linux сайта developerWorks](#) можно найти сотни [пошаговых инструкций и руководств](#), загрузить программные продукты, а также получить ссылки на форумы и многие другие ресурсы, ориентированные на разработчиков и администраторов Linux.

## Об авторах



Трейси Бост - опытный разработчик программного обеспечения и проектировщик систем. Он специализируется на интеграции корпоративных приложений. Был сопредседателем рабочей группы по бизнес-правилам Организации по поддержке стандартов ипотечной отрасли (MISMO) и сопредседателем секции промышленных стандартов на симпозиуме RuleML2010. Работал в различных отраслях, в том числе в ипотеке, недвижимости и некоммерческом секторе.



**Род Смит (Rod Smith)** долгое время работает техническим консультантом и является автором более десятка книг о Linux, UNIX и сетях.

Шон Уолберг работал с Linux- и UNIX-системами с 1994 года в академических, корпоративных и "провайдерских" кругах. Он широко освещает вопросы системного администрирования в течение нескольких последних лет. С ним можно связаться по адресу [sean@ertw.com](mailto:sean@ertw.com).

# Изучаем Linux, 302 (смешанные среды): Основные принципы

*Основные принципы общего использования файлов и принтеров в смешанной среде*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Эта статья поможет вам подготовиться к сдаче экзамена 302 сертификационной программы института Linux Professional Institute (LPI). Из этой статьи вы узнаете об основных принципах и концепциях протоколов Server Message Block (SMB) и Common Internet File System (CIFS) и технологии совместного использования файлов и служб печати в смешанной среде.

[Больше статей из этой серии](#)

**Дата:** 20.03.2012

**Уровень сложности:** сложный

## О факультативном экзамене LPI-302

Как и многие другие программы, программа сертификации Linux Professional Institute (LPIC) предусматривает различные уровни сертификации, где для получения каждого последующего уровня необходимо обладать более глубокими знаниями и практическим опытом. Экзамен LPI-302 – это факультативный экзамен третьего уровня программы LPIC, требующий продвинутых знаний в области системного администрирования Linux.

Для получения сертификата LPIC третьего уровня (LPIC-3) необходимо успешно сдать два экзамена первого уровня (101 и 102), два экзамена второго уровня (201 и 202), а также базовый экзамен 301 третьего уровня (LPIC-3). Если вы получили сертификат третьего уровня, то вы можете сдавать факультативные экзамены по определенным специализациям, например, экзамен LPI-302.

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

### Краткий обзор

Из этой статьи вы узнаете о следующих концепциях:

- Протоколы Server Message Block (SMB) и Common Internet File System (CIFS).
- Совместное использование файлов.
- Службы печати.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 310.1 темы 310. Цель имеет вес 1.

### Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. Кроме этого, у вас должен быть доступ к среде Windows, которую можно использовать для проверки доступа к файлам и принтерам.

### **Общий доступ к файлам и принтерам**

Цель предоставления общего доступа к файлам и принтерам заключается в том, чтобы позволить компьютеру использовать жесткие диски и принтеры других компьютеров так, как если бы эти устройства были подключены локально. Эта функциональность позволяет большому числу пользователей использовать одни и те же ресурсы (которые могут быть дорогостоящими) и получать доступ к ним из любой точки сети. Являясь централизованными ресурсами, файлы и принтеры могут управляться и контролироваться настолько жестко, насколько это необходимо.

При реализации решений по предоставлению общего доступа к файлам и принтерам следует учитывать следующие требования:

- Решение должно работать в любой сети вне зависимости от ее типа (Ethernet, Token Ring и т. д.)
- Приложениям не нужно знать о том, обращаются ли они к локальному жесткому диску или принтеру или к удаленному.
- Необходимо предусмотреть механизм управления одновременными обращениями к одному и тому же ресурсу, например, когда двое сотрудников пытаются одновременно послать задание на один и тот же принтер.
- Необходимо реализовать некую систему аутентификации, позволяющую определить, кто именно обращается к ресурсу, а также систему авторизации, которая разрешает доступ только заранее определенным пользователям.

### **Общий доступ к файлам**

Даже если общий доступ к файлу обеспечивает рабочая станция, устройство, предоставляющее ресурс в общее пользование, называется *сервером*, а устройства, обращающиеся к этому ресурсу называются *клиентами*. Говорят, что сервер *экспортирует файловую систему* (в терминах UNIX) или *предоставляет общий доступ к диску или директории* (в терминах Windows). Клиенты монтируют файловую систему как локальный диск или подключаются к общему ресурсу.

Подключение к общим ресурсам может быть краткосрочным или долговременным. Клиент может подключиться к серверу, получить несколько файлов и затем отключиться, либо клиент может поддерживать подключение к серверу до тех пор, пока тот не будет перезагружен. Клиент может воспринимать удаленную файловую систему как локальный диск или часть локальной файловой системы, а может получать доступ к ресурсу на уровне приложения, как, например, это делают FTP-клиенты.

Другой особенностью, связанной с общим доступом к ресурсам, является концепция *просмотра сетевого окружения*. Просмотр сетевого окружения позволяет клиентам обнаруживать серверы в сети, как правило, на основе динамически обновляющихся списков.

### **Общий доступ к принтерам**

Принтер может быть не подключен к серверу напрямую, поэтому для обеспечения общего доступа могут использоваться сервисы разных уровней. Когда несколько клиентов настроены на использование одного и того же принтера, обычно они обращаются к нему через службу, которая называется *спулером* (spooler). Задача спулера заключается в управлении списком заданий печати (*очередь печати*). Когда несколько пользователей отправляют задания на принтер, то спулер сохраняет их на диск. Когда принтер свободен, спулер может послать на него задание, не опасаясь конфликтов с другими заданиями.

Даже если принтер подключен к сети, обычно в качестве спулера используется сервер, который, в свою очередь, отправляет задания на принтер по сети. Это делается по трем причинам: во-первых, сервер имеет более надежную и объемную область для спулинга, чем у принтера, во-вторых, сервер может управлять доступом к устройству, и в-третьих, сервер может передавать клиентам необходимые драйверы принтера.

## SMB и CIFS

SMB является протоколом, а не реализацией. Этот протокол реализован во многих операционных системах, начиная с Windows и заканчивая UNIX и даже майнфреймами.

### История протокола

Протокол SMB был разработан IBM и реализован компанией Microsoft, когда она начала внедрять сетевую поддержку в свои продукты в начале 1990-х. В то время для поддержки SMB требовалось использование отдельного продукта, например, LAN Manager или Windows for Workgroups, но, в конечном счете, с появлением Microsoft Windows NT возможность совместного использования файлов и принтеров стала частью операционной системы.

Microsoft продолжала внедрять поддержку SMB в новых компонентах своих операционных систем и в итоге появилась версия этого протокола под названием *CIFS*, которую Microsoft пыталась утвердить как отраслевой стандарт комитета IETF (Internet Engineering Task Force).

Приблизительно в это же время австралийский студент Эндрю Триджелл (Andrew Tridgell) приступил к обратному инженерингу реализации SMB для майнфреймов и начал разрабатывать продукт, в итоге превратившийся в Open source-реализацию SMB и протоколов Microsoft и получивший имя *Samba*. Изначально целью проекта являлась передача файлов по сети, но в итоге превратился в полноценное клиент-серверное приложение, способное работать в качестве контроллера домена Windows, а затем и в качестве сервера Active Directory.

### Сравнение SMB и CIFS

С технической точки зрения CIFS является *диалектом* SMB. Протокол SMB видоизменялся на протяжении многих лет, поэтому и для клиента, и для сервера необходимо согласовать диалект протокола, на котором они общаются. CIFS относится к семейству расширений NT LAN Manager (NTLM). По странному стечению обстоятельств этот протокол не идентифицирует себя с помощью строки **CIFS**, а использует для этого строку **NT LM 0.12**.

На практике можно использовать оба термина, SMB и CIFS, поскольку они взаимозаменяемы, а CIFS является диалектом SMB, использующимся в настоящее время. Кстати, CIFS произносится так же, как и пишется ("siffs"), тогда как название SMB произносится по буквам.

### Обзор протокола SMB

Поскольку SMB является клиент-серверным протоколом, он обеспечивает связь клиента с сервером. Изначально приложения, работающие с SMB, должны были использовать API-интерфейс под названием *сетевая базовая система ввода-вывода* (Network Basic Input/Output System, NetBIOS). Этот программный интерфейс обеспечивал поддержку нескольких служб, используемых SMB, а также нескольких служб, связанных с разрешением имен и просмотром сетевого окружения. Работая в связке с NetBIOS, SMB мог работать поверх следующих протоколов:

- "Сырой" Ethernet. В этом случае SMB использует передачу кадров NetBIOS посредством интерфейса *NetBEUI*.
- Протокол Internetwork Packet Exchange (IPX)/Sequenced Packet Exchange(SPX), разработанный Novell. В этом случае SMB использует протокол NBX (NetBIOS поверх IPX/SPX).

- Стек TCP/IP. В этом случае SMB использует протокол NBT (NetBIOS поверх TCP/IP). NetBIOS обеспечивает в сетях Microsoft работу трех ключевых служб:

- **Служба имен** для поиска компьютеров в сети.
- **Служба сеансов** для надежного взаимодействия и передачи данных между клиентом и сервером.
- **Служба распространения дейтаграмм** для передачи небольших сообщений и широковещательных рассылок

Наиболее важной службой для SMB является служба сеансов, тем не менее, две другие службы также используются, например, служба имен используется для нахождения IP-адреса сервера.

В конечном счете, широкое распространение стека протоколов TCP/IP и переход Microsoft на использование службы доменных имен Domain Name System (DNS) привели к тому, что SMB стал запускаться непосредственно поверх TCP/IP методом *прямой передачи* (direct hosting). Тогда как протокол NBT использовал TCP- и UDP-порты 137-139, прямая передача задействует TCP/UDP порт 445.

### Протокол SMB в модели OSI

На рисунке 1 показана сетевая модель взаимодействия открытых систем (Open Systems Interconnection, OSI), которая широко используется для описания взаимодействий сетевых протоколов передачи данных. Модель OSI описывает функции, необходимые для работы сетевых приложений, и разделяет их на несколько уровней. Каждый уровень использует все функции нижележащих уровней и обслуживает все вышележащие уровни.

**Рисунок 1. Сетевая модель OSI**



Уровни 1 (физический) и 2 (канальный) полностью обеспечивают работу стандарта Ethernet. На этих двух уровнях небольшие фрагменты данных (пакеты) передаются от одного узла сети другому. На следующих двух уровнях работают протоколы TCP (транспортный уровень) и IP (сетевой уровень). Сетевой уровень обеспечивает сквозную маршрутизацию, а транспортный уровень позволяет передавать сообщения большего объема, составленные более мелких сообщений (пакетов) и работать нескольким службам на одном сервере. Каждый уровень использует все функции нижележащих уровней, поэтому находясь на сетевом уровне, нет необходимости знать о том, как каждый узел сети использует физический и канальный уровни и т. д.

Если в сети работает NetBIOS, то используется *сеансовый уровень*. Сеансовый уровень отвечает за службу сеансов, которая обсуждалась ранее. Протокол SMB работает на *уровне приложений* (уровень представления здесь не используется); это означает, что SMB использует функции всех нижележащих уровней.

Если в сети не используется NetBIOS, то не используется и сеансовый уровень. Протокол TCP обладает достаточными возможностями для работы с большинством сеансовых служб, а службы имен обрабатываются протоколом DNS.

### **Универсальное соглашение об именовании**

Вам наверняка знакомы универсальные идентификаторы ресурсов (Universal Resource Identifiers, URIs) вида `http://ibm.com/developerworks`, которые используются в Интернете. Идентификаторы URI определяют, где можно найти тот или иной документ или другое содержимое. В нашем примере префикс `http` обозначает схему, которая говорит о том, что доступ к документу осуществляется по протоколу HTTP. Далее следует имя узла `ibm.com`, на котором хранится нужное нам содержимое, и заключительная часть `/developerworks`, являющаяся именем конечного запрашиваемого ресурса.

Точно так же, UNC-пути (Uniform Naming Convention – универсальное соглашение об именовании) служат для идентификации ресурсов в сетях Windows. UNC-путь имеет следующий вид: `\shorty\documents\public\photo.jpg`. UNC-путь начинается с двух обратных слешей (`\`), после которых указывается имя сервера, еще один обратный слеш и имя общей директории. Далее указывается путь к требуемому ресурсу относительно этой общей директории.

UNC-пути имеют следующие отличия от идентификаторов URI:

- Нет необходимости указывать схему, поскольку всегда используется SMB.
- Вместо прямых слешей (`/`) используются обратные, хотя в некоторых случаях могут использоваться и прямые.
- Имя общей директории используется лишь для ссылки на общий ресурс, расположенный на сервере. Объекта с именем общей директории может и не существовать в файловой системе сервера.

### **Samba**

Samba позиционируется как стандартный пакет программ для Linux и UNIX, функционально совместимый с Windows ("standard Windows interoperability suite of programs for Linux and UNIX"). Несмотря на то, что в Linux и других версиях UNIX имеются различные способы монтирования общих файловых ресурсов SMB, ни один из них не может сравниться с Samba по уровню функциональности как в области клиент-серверного взаимодействия, так и в области интеграции с сетями Windows.

Samba состоит из нескольких демонов, работающих в фоновом режиме и предоставляющих сервисы и ряд инструментов командной строки для взаимодействия со службами Windows или Samba. Все демоны и инструменты будут подробно рассмотрены в оставшихся частях этой серии, но сейчас мы отметим несколько наиболее важных исполняемых файлов:

- **smbd.** Демон, являющийся SMB-сервером файловых служб и служб печати.
- **nmbd.** Демон, предоставляющий службы имен NetBIOS.
- **mount.cifs.** Эта утилита монтирует удаленную файловую систему SMB в локальную файловую систему UNIX, подобно локальному диску или общему ресурсу NFS.
- **smblclient.** Эта утилита предоставляет доступ из командной строки к ресурсам SMB, подобно клиенту FTP. Она также позволяет получать списки общих ресурсов на удаленных серверах и просматривать сетевое окружение.
- **smb.conf.** Это конфигурационный файл, содержащий настройки для всех

инструментов Samba. Хотя это и не исполняемый файл, его необходимо упомянуть в этом разделе, поскольку он достаточно важный.

Как и любое другое Open Source-приложение, Samba можно скомпилировать из исходного кода или загрузить в составе дистрибутива с помощью установочных сценариев инициализации системы и конфигурационных файлов.

## Что дальше

Следующая статья этой серии содержит материалы цели 310.2 темы 310. В ней рассматриваются различные роли демонов Samba, а также различные режимы безопасности, в которых работают эти демоны.

Ресурсы

## Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Concepts](#) (EN).
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- Узнайте больше о [SMB](#) на страницах Википедии.
- Прочтайте бесплатную онлайновую версию книги [Implementing CIFS](#) (EN) Кристофера Хертеля (Christopher R. Hertel), в которой подробнейшим образом рассматривается протокол CIFS.
- В статье Microsoft [Прямое управление из SMB по протоколу TCP/IP](#) более подробно рассматривается прямое размещением ресурсов через SMB и даются советы по настройке этого протокола в среде Windows.
- Смотрите [демонстрационные материалы по запросу на сайте developerWorks](#) (EN), ориентированные как на новичков, так и на опытных разработчиков.

## Получить продукты и технологии

- Загрузите [Samba](#) (EN) и следите за последними новостями этого проекта.

# Изучаем Linux, 302 (смешанные среды): Роли Samba

*Архитектура, принципы и схема работы*

[Родерик Смит \(Roderick Smith\)](#), автор и консультант, IBM

**Описание:** Samba – это не просто программа, а комплекс взаимосвязанных специализированных серверных компонентов и утилит. Если вы знаете, чем отличаются эти серверы и утилиты, то сможете более эффективно управлять системой Samba. Кроме того, в Samba реализовано несколько различных моделей безопасности, о которых необходимо знать, чтобы правильно интегрировать Samba в существующую сеть или же создать новую

сеть на основе серверов Samba.

## [Больше статей из этой серии](#)

**Дата:** 22.03.2012

**Уровень сложности:** сложный

### **Об этой серии**

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

### **Краткий обзор**

Из этой статьи вы узнаете о следующих концепциях:

- Режимы безопасности Samba.
- Роли основных демонов Samba.
- Управление демонами Samba.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 310.2 темы 310. Цель имеет вес 1.

### **Предварительные требования**

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все описанные команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и в общих чертах понимает назначение Samba (о чем рассказывалось в предыдущей статье "[Изучаем Linux, 302: основные принципы](#)"). Для выполнения примеров этой статьи на вашем компьютере должно быть инсталлировано программное обеспечение Samba. Для выполнения некоторых действий потребуется сетевое окружение, в котором используется протокол Server Message Block (SMB)/Common Internet File System (CIFS).

### **Демоны Samba**

Серверные службы Linux зачастую реализованы в виде *демонов*. Это слово было позаимствовано из греческой мифологии, в которой демоны являлись полезными сверхъестественными существами. Любой демон Linux работает в фоновом режиме и выполняет определенную полезную задачу. Серверный пакет Samba содержит несколько демонов, включая `smbd`, `nmbd` и `winbinddd`. Программа `swat` также является сервером Samba, но, как правило, она запускается из-под супер-сервера и технически не является демоном.

### **Понимание `smbd`**

Программа `smbd` обеспечивает основную базовую функциональность Samba, выполняя следующие функции:

- **Обеспечение общего доступа к файлам и принтерам.** Эта функция `smbd`, вероятно, является одной из наиболее важных функций Samba.
- **Аутентификация пользователей.** Демон `smbd` авторизует пользователей, выполняя поиск в локальной базе данных или передавая запросы на другой компьютер. Если

сервер Samba настроен в качестве контроллера домена, то `smbd` также отвечает за обработку запросов авторизации, поступающих с других компьютеров (конфигурации для рабочей группы и домена будут кратко описаны в разделе [Установка режима безопасности](#)).

- **Служба времени.** Samba может сообщать другим компьютерам о текущем времени. Этим также занимается `smbd`.

По умолчанию демон `smbd` использует TCP-порты 139 и 445. Порт 139 обеспечивает работу SMB через службы NetBIOS поверх TCP (именно так с SMB/CIFS работает большинство более старых клиентов). Порт 445 обеспечивает работу SMB напрямую через TCP (через этот порт доступ к серверу Samba получает большинство более новых клиентов).

Поскольку выполняемые `smbd` функции (в частности, предоставление общего доступа к файлам и принтерам и аутентификация) имеют критический характер, то этот демон можно считать ядром Samba. Фактически, если на компьютере запущен только лишь демон `smbd`, к нему можно подключиться и обмениваться файлами, используя, например, программу `smbclient` из состава Samba. Тем не менее, работа некоторых клиентов зависит от служб, предоставляемых другими демонами Samba.

### Понимание `nmbd`

Вторым ключевым демоном Samba является `nmbd`. Его основной обязанностью является выполнение задач, связанных с использованием имен. Можно рассматривать его как собственную доменную систему имен (DNS) Samba, хотя он не так сложен в настройке и намного проще, чем DNS. `Nmbd` выполняет следующие функции:

- **Ответы на широковещательные запросы имен.** В одном из основных режимов работы клиенты посылают в сегмент сети широковещательные запросы на разрешение имен. Сервер `nmbd` отслеживает такие широковещательные запросы и отвечает на них, если система Samba настроена на использование запрашиваемых имен. Сервер `nmbd` также отвечает на запросы в тех случаях, когда он настроен в качестве прокси для другого компьютера или сети.
- **Регистрация имен NetBIOS.** Для того чтобы разрешение имен NetBIOS работало, компьютеры должны зарегистрировать свои имена либо используя центральный сервер имен NetBIOS (NBNS, известный также, как сервер Windows Internet Name Service [WINS]), либо отправляя в сеть широковещательные пакеты с именами и соответствующими им переговорными правами. `Nmbd` отвечает за этот процесс, а также за переговоры с другими компьютерами, обладающими такими правами и зарегистрировавшими свои имена.
- **Работа в качестве сервера NBNS.** Система Samba может быть настроена в качестве системы NBNS, функции которой выполняет `nmbd`.
- **Работа в качестве главного обозревателя (master browser).** Одним из действий, выполняемых пользователями в сети SMB/CIFS, является *обзор* – возможность просматривать серверы в сети наподобие просмотра директорий на жестком диске компьютера. Эта возможность зависит от наличия в сети *главного обозревателя*, который формирует и распространяет списки просмотра. Если система Samba настроена в качестве главного обозревателя, эти функции выполняет `nmbd`.

Как видно из вышеприведенного списка, `nmbd` отвечает за выполнение множества задач. Хотя эти задачи не связаны с обслуживанием файлов или принтеров, многие из них являются критическими функциями любого сервера SMB/CIFS, поэтому демон `nmbd` также следует считать критически важным компонентом Samba. Обычно этот демон запускается вместе с демоном `smbd` в сценариях запуска операционной системы.

Для большинства функций `nmbd` требуется использовать TCP-порт с номером 137, однако служба главного обозревателя использует UDP-порт с номером 138.

## **Понимание winbindd**

Третий демон Samba – это демон **winbindd**. В отличие от **smbd** и **nmbd**, **winbindd** не предоставляет никаких сервисов удаленным компьютерам, а служит интерфейсом между контроллером домена Windows® (или Samba) и собственными PAM-модулями (Pluggable Authentication Modules – подключаемые модули аутентификации) локальных компьютеров, позволяя контроллеру домена хранить информацию об учетных записях Linux.

Во многих дистрибутивах **winbindd** устанавливается отдельно от Samba в качестве самостоятельного пакета, который обычно называется **winbind** или **winbindd**. Как правило, запускается он тоже отдельно. В принципе, **winbindd** можно запускать на компьютере, на котором не установлен сервер Samba и который даже не является клиентом SMB/CIFS, не считая того, что **winbindd** сам является клиентом SMB/CIFS. Однако, как показывает практика, если на компьютере с Linux запущен **winbindd**, то, вероятно, на нем будет работать сервер Samba или он будет являться клиентом SMB/CIFS.

## **Управление Samba**

Управление системой Samba включает в себя две задачи: настройка параметров конфигурации в конфигурационном файле и их изменение "на лету". В первом случае задача решается путем редактирования конфигурационного файла Samba, а во втором случае – путем использования конфигурационной утилиты **smbcontrol**.

### **Настройка параметров в конфигурационном файле**

Главный конфигурационный файл Samba называется *smb.conf* и обычно расположен в директории */etc/samba*, хотя может располагаться и в другом месте (если вы компилируете Samba из исходного кода, то он может быть в директории */usr/local/samba/lib*).

Файл *smb.conf* разбит на разделы, каждый из которых начинается с имени, заключенного в квадратные скобки ([ ]), например, [**global**] или [**documents**]. В большинстве разделов настраиваются параметры общих ресурсов (файлов или принтеров), однако раздел [**global**] является особенным: в нем содержатся параметры, которые влияют на работу всего сервера или задают значения по умолчанию, используемые в последующих определениях общих ресурсов. Обычно раздел [**global**] находится в самом начале конфигурационного файла.

Строки в файле *smb.conf* могут являться комментариями, начинающимися с символов решетки (#) или точки с запятой (;), названиями разделов или строками, содержащими параметры Samba. В последнем случае строки имеют следующий вид:

*параметр* = *значение*

**Параметр** – это ключевое слово, например, **security** или **create mask**. Имена параметров не чувствительны к регистру. Для некоторых распространенных параметров существуют синонимы, а для некоторых – антонимы. Например, **writable** и **writeable** – это синонимы, а **read only** – антоним для них, т. е. конструкция **read only = Yes** эквивалентна конструкции **writable = No**.

**Значение**, присваиваемое параметру, может быть строкой произвольной формы (включая специализированные числовые значения, например, IP-адреса), логическим значением, переменной или списком. Логические значения могут принимать одно из следующих значений: **Yes**, **True**, **1** (синонимы) или **No**, **False**, **0**.

Переменные начинаются с символа процента (%) и предназначены для хранения информации, которая может быть не известна на момент создания конфигурационного файла. Например, переменная %D означает имя рабочей группы или домена сервера, %h – DNS-имя сервера, %H

– домашнюю директорию пользователя, %L – NetBIOS-имя сервера и %U – имя пользователя.

Некоторые параметры могут представлять собой списки из нескольких значений, например, списки имен пользователей. В этих случаях элементы списка разделяются запятыми (например, конструкция `george, mary` относится как к пользователю `george`, так и к пользователю `mary`). В большей части файла `smb.conf` символ пробела игнорируется, поэтому если значение содержит такой символ, оно должно быть заключено в кавычки.

Файл `smb.conf` содержит параметры для `smbd`, `nmbd`, `winbindd`, а также других сервисов и приложений Samba. Параметры различных программ не отделяются явно друг от друга, хотя иногда имя параметра позволяет ясно понять, для какого демона он предназначен.

### Управление Samba с помощью `smbcontrol`

Для управления работающей системой Samba используется программа `smbcontrol`.

Например, можно указать Samba, чтобы она закрыла общий доступ к определенному ресурсу, запустила принудительную процедуру выбора главного обозревателя, перезагрузила конфигурационный файл и т. д. Базовый синтаксис этой команды имеет следующий вид:

```
smbcontrol [-i] [-s файл_конфигурации]
smbcontrol [назначение] [сообщение] [параметр]
```

При использовании опции `-i` программа `smbcontrol` переходит в интерактивный режим, в котором можно выполнять по очереди несколько команд, не набирая каждый раз `smbcontrol` перед именем команды. В поле **назначение** можно указать имя сервера (`smbd`, `nmbd` или `winbindd`), ключевое слово `all` (если требуется послать сообщение всем демонам) или номер идентификатора процесса. Поле **сообщение** – это команда (список команд приведен в таблице 1). В поле **параметр** можно указать необязательный параметр, который может потребоваться какой-либо команде.

**Таблица 1. Список команд `smbcontrol`**

Команда (сообщение)	Описание
<code>close-share</code>	Закрывает общий доступ к ресурсу, указанному в параметре.
<code>debug</code>	Устанавливает уровень отладки, указанный в параметре.
<code>force-election</code>	Запускает принудительную процедуру выборов главного обозревателя.
<code>debuglevel</code>	Отображает текущий уровень отладки демона.
<code>printnotify</code>	Отправляет сообщение клиентам, подключенным к общему принтеру, внося принудительные изменения в статус очереди клиентов.
<code>samsync</code>	Синхронизирует базу данных пользователей с контроллером домена (в официальной документации сказано, что эта функция на текущий момент не работает, поэтому протестируйте ее, прежде чем использовать).
<code>shutdown</code>	Останавливает работу указанного демона.
<code>pool-usage</code>	Отображает информацию об использовании памяти для указанного демона.
<code>drvupgrade</code>	Уведомляет клиентов о новом доступном драйвере принтера. В качестве параметра этой команды указывается имя общего принтера.
<code>reload-config</code>	Указывает серверу принудительно перезагрузить файл <code>smb.conf</code> .

### Установка режима безопасности

В Samba имеется ряд параметров, связанных с аутентификацией пользователей. Наиболее

важным из них является параметр **security**, который может принимать пять различных значений:

- **Share.** Этот режим безопасности эмулирует метод аутентификации, используемый операционными системами Microsoft® Windows 9x/Windows Me. В этом режиме имена пользователей игнорируются, а пароли назначаются общим ресурсам. В этом режиме Samba пытается использовать предоставленный клиентом пароль, которым могут пользоваться разные пользователи.
- **User.** Этот режим безопасности установлен по умолчанию и использует для аутентификации имя пользователя и пароль, как это обычно делается в Linux. В большинстве случаев в современных операционных системах пароли хранятся в зашифрованной базе данных, которую использует только Samba.
- **Server.** Этот режим безопасности используется тогда, когда необходимо, чтобы Samba выполняла аутентификацию, обращаясь к другому серверу. Для клиентов этот режим выглядит так же, как аутентификация на уровне пользователя (режим User), но фактически для выполнения аутентификации Samba обращается к серверу, указанному в параметре **password server**.
- **Domain.** Используя этот режим безопасности, вы можете полностью присоединиться к домену Windows; для клиентов это выглядит так же, как аутентификация на уровне пользователя. В отличие от аутентификации на уровне сервера, доменная аутентификация использует более защищенный обмен паролями на уровне домена. Для полного присоединения к домену требуется выполнить дополнительные команды в системе Samba и, возможно, на контроллере домена.
- **ADS.** Этот режим безопасности похож на метод аутентификации в домене, но требует наличия контроллера домена Active Directory® Domain Services.

В качестве общей рекомендации мы советуем использовать аутентификацию на уровне пользователей, которая является наилучшим выбором в том случае, если ваш сервер Samba является членом рабочей группы Windows (простейшая форма сети SMB/CIFS). Рабочая группа отличается от домена преимущественно тем, что в последнем случае имеется контроллер домена – сервер, обеспечивающий аутентификацию в домене. Для использования контроллера домена необходимо установить режим безопасности (параметр **security**) Server, Domain или ADS. Режим Server более прост в настройке, но менее безопасен, тогда как режим ADS наоборот, является наиболее сложным в настройке и наиболее безопасным.

Для полного присоединения к домену (режим безопасности Domain или ADS) необходимо указать в разделе **[global]** файла smb.conf несколько дополнительных параметров:

```
password server = DOMCONT
domain logons = No
encrypt passwords = Yes
```

Вместо **DOMCONT** необходимо указать имя контроллера домена. Кроме того, на сервере Samba, который вы хотите присоединить к домену, необходимо выполнить следующую команду:

```
# net join member -U пользователь-администратор
```

Для присоединения сервера Samba к домену может также потребоваться настроить контроллер домена. Этот вопрос рассматривается в цели 312.4 (управление доменом).

Режим безопасности Share сильно устарел и существует только для обеспечения совместимости с очень старыми клиентами, не воспринимающими имена пользователей. Его

также можно использовать для общих ресурсов, для которых нужно обеспечить минимальную защиту, например, общие принтеры. Поскольку операционная система Linux требует, чтобы для всех типов доступов обязательно использовалась учетная запись, Samba пытается сравнивать полученный пароль с некоторым набором учетных записей до тех пор, пока не будет найдено совпадение или пароль не будет отвергнут. Вот список этих учетных записей:

- Гостевая учетная запись (задается параметром `guest account`), если `guest only = Yes`.
- Имя пользователя, указанное клиентом (не все клиенты указывают имя пользователя, но некоторые делают это).
- Имя пользователя, использовавшееся в последний раз при доступе с клиентского компьютера.
- Имя общего ресурса, к которому пытается получить доступ клиент.
- NetBIOS-имя клиента.
- Любые имена пользователей, указанные с помощью параметра `username`.

Поскольку большинство клиентов на сегодняшний день понимают имена пользователей, редко возникает необходимость в режиме безопасности на уровне ресурса. Использование этого режима лишь приводит к путанице и снижает уровень безопасности: пароли сравниваются с множеством учетных записей, и если один пароль случайно попадет другому пользователю, это создаст значительный риск для вашего сервера.

## Что дальше

Следующая статья этой серии содержит материалы цели 310.3 темы 310. В ней рассматривается формат файлов базы данных Trivial Database (TDB), которую Samba использует для хранения информации об учетных записях.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Samba roles](#) (EN).
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Чтобы получить ссылки на все статьи этой серии, которые помогут вам подготовиться к успешной сдаче экзамена LPI-302, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- В [разделе Linux сайта developerWorks](#) можно найти сотни [пошаговых инструкций и руководств](#), загрузить программные продукты, а также получить ссылки на форумы и многие другие ресурсы, ориентированные на разработчиков и администраторов Linux.

### Получить продукты и технологии

- Вы можете загрузить систему Samba и получить информацию о ней на [Web-сайте Samba](#) (EN).

# Изучаем Linux, 302 (смешанные среды): Файлы базы данных Trivial Database

*Работа с файлами базы данных Samba Trivial Database и устранение возможных проблем*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Samba использует файлы базы данных Trivial Database для хранения постоянных и временных данных, обеспечивая общий доступ к файлам и принтерам в смешанных средах Linux/Windows. Эта статья поможет вам подготовиться к сдаче экзамена LPI-302; вы узнаете о формате Trivial Database (TDB), который Samba использует для хранения информации, а также о том, как просматривать TDB-файлы и создавать для них резервные копии.

[Больше статей из этой серии](#)

**Дата:** 03.04.2012

**Уровень сложности:** сложный

## Краткий обзор

В этой статье рассматриваются следующие темы:

- Создание резервных копий файлов Samba Trivial Database (TDB).
- Восстановление TDB-файлов.
- Определение признаков повреждений TDB-файлов.
- Просмотр и редактирование содержимого TDB-файлов.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 310.3 темы 310. Цель имеет вес 1.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все описанные команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и в общих чертах понимает назначение Samba (о чем рассказывалось в предыдущей статье "[Изучаем Linux, 302: основные принципы](#)"). Для выполнения примеров этой статьи на вашем компьютере должно быть инсталлировано программное обеспечение Samba. Для выполнения некоторых действий потребуется сетевое окружение, в котором используется протокол Server Message Block (SMB)/Common Internet File System (CIFS).

## Понимание TDB-файлов

В процессе работы Samba использует большой объем информации, начиная от локальных паролей и заканчивая списком клиентов. Некоторые данные используются временно и могут быть удалены при перезагрузке Samba, а некоторые являются постоянными, и их нельзя терять. Иногда данные невозможно или не нужно хранить в оперативной памяти, поскольку они могут иметь слишком большой объем, к ним могут редко обращаться, либо они должны сохраняться при перезагрузках. Для этих случаев команда разработчиков Samba разработала базу данных под названием Trivial Database. Эта база данных является эффективным хранилищем записей типа *ключ-значение*; это означает, что данные хранятся, сортируются и извлекаются с помощью уникальных ключей, и что при этом не используется объединение таблиц, как это происходит в реляционных базах данных. Хранилища типа "ключ-значение" и, в частности, TDB, позволяют быстро сохранять информацию на жесткий диск и получать ее обратно.

## **Хранилища типа "ключ-значение"**

Несмотря на то, что у TDB имеется много альтернатив, например, GNU Database Manager (GDBM), проект Samba предъявляет особые требования к одновременной записи данных в БД несколькими процессами, а также к поддержке блокировок внутренних областей данных. Поэтому команда разработчиков Samba создали собственный менеджер базы данных и назвали его *Trivial Database Manager*. Дальнейшее усовершенствование TDB в рамках проекта Clustered TDB (CTDB) реализовало поддержку кластеризации, которая может использоваться в других проектах.

Если вы знакомы с реляционными СУБД, такими как IBM DB2, MySQL или PostgreSQL, то вы заметите, что по сравнению с ними хранилища типа "ключ-значение" достаточно примитивны. Тогда как реляционные базы данных могут содержать таблицы с несколькими столбцами, хранилища "ключ-значение" фактически содержат только два столбца: столбец с ключом и столбец со значением. В реляционных СУБД для извлечения данных из нескольких таблиц и столбцов используется структурированный язык запросов (Structured Query Language, SQL), а в хранилищах типа "ключ-значение" – только операции со столбцом ключей.

В силу своей простоты хранилища "ключ-значение" могут использоваться для ограниченного круга задач, однако при этом они обеспечивают очень высокое быстродействие. Хранилище "ключ-значение", по сути, представляет собой хэш-таблицу, хранящуюся на диске, поэтому местонахождение фрагмента данных можно предугадать, не обращаясь к индексу.

В частности, TDB была разработана для обработки множественных конкурирующих процессов записи и двоичных данных, таких, как структуры внутренних данных. Поэтому самым быстрым способом сохранения и извлечения этих данных является использование двоичного формата. Двоичное хранилище позволяет уменьшить размер файла базы данных и избавляет от необходимости преобразовывать данные в различные форматы в процессе записи информации в базу данных и ее извлечения оттуда.

## **TDB-файлы, используемые в Samba**

Samba хранит свои TDB-файлы в нескольких местах. Самый простой способ найти эти файлы – просмотреть вывод команды `smbd -b`, представленный в листинге 1.

### **Листинг 1. Вывод информации о сборке smbd**

```
# smbd -b
...
Paths:
  SBINDIR: /usr/sbin
  BINDIR: /usr/bin
  SWATDIR: /usr/share/swat
  CONFIGFILE: /etc/samba/smb.conf
  LOGFILEBASE: /var/log/samba
  LMHOSTSFILE: /etc/samba/lmhosts
  LIBDIR: /usr/lib
  MODULESDIR: /usr/lib/samba
  SHLIBE1T: so
  LOCKDIR: /var/lib/samba
  STATEDIR: /var/lib/samba
  CACHEDIR: /var/lib/samba
  PIDDIR: /var/run
  SMB_PASSWD_FILE: /var/lib/samba/private/smbpasswd
  PRIVATE_DIR: /var/lib/samba/private
```

На местоположения, в которых могут храниться TDB-файлы, указывают несколько путей в листинге 1. К счастью, многие из них одинаковы, поэтому количество вариантов сокращается. Например, переменные `LOCKDIR`, `STATEDIR` и `CACHEDIR` указывают на директорию `/var/lib/samba`, а переменная `PRIVATE_DIR` является поддиректорией с именем `private`. Вы можете самостоятельно зайти в эти директории и поискать в них TDB-файлы.

TDB-файлы можно разделить на две основные группы: постоянные и временные. В таблице 1 перечислены имена и описания постоянных файлов.

**Таблица 1. Постоянные TDB-файлы**

Имя файла	Назначение
<code>account_policy.tdb</code>	Хранит политики учетных записей, такие как пороговые значения блокировок и длина паролей.
<code>group_mapping.tdb</code>	Хранит информацию о сопоставлениях между идентификаторами пользователей Windows NT (SID) и UNIX®.
<code>ntdrivers.tdb</code>	Хранит информацию о драйверах принтеров.
<code>ntforms.tdb</code>	Хранит информацию о формах очереди печати, например, размеры бумаги.
<code>ntprinters.tdb</code>	Хранит информацию о параметрах инициализации принтеров.
<code>passdb.tdb</code>	Используется в некоторых режимах безопасности для хранения учетных данных аутентификации.
<code>registry.tdb</code>	Базовый реестр, к которому можно получить доступ с других Windows-компьютеров.
<code>secrets.tdb</code>	Хранит локальные секретные значения (например, ключи machine key компьютеров) и учетные данные (например, пароли LDAP).
<code>share_info.tdb</code>	Хранит списки контроля доступов (ACL) для каждого локального общего ресурса.
<code>winbindd_idmap.tdb</code>	Хранит информацию о сопоставлениях между идентификаторами пользователей Windows NT (SID) и динамически создаваемыми пользователями при использовании winbind.

Для всех TDB-файлов, перечисленных в таблице 1, следует создавать резервные копии. Эта процедура будет описана далее в этой статье.

Помимо постоянных файлов в Samba используется множество временных TDB-файлов, в которых хранится информация о состоянии и кэшированные данные. Эти файлы не нужно резервировать, поскольку они заново создаются при каждом запуске Samba.

## Использование TDB-файлов

Управление TDB-файлами в Samba осуществляется с помощью трех инструментов:

- **`tdbdump`**: позволяет выводить содержимое TDB-файлов.
- **`tdbbackup`**: позволяет создавать и проверять резервные копии TDB-файлов.
- **`tdbtool`**: позволяет создавать, просматривать и редактировать TDB-файлы.

## Просмотр содержимого TDB-файла

Самый быстрый способ заглянуть в TDB-файл – это сделать его дамп с помощью команды `tdbdump`. В листинге 2 показан дамп файла `ntprinters.tdb`.

## Листинг 2. Использование команды `tdbdump`

```
[root@bob ~]# tbdump /var/lib/samba/ntprinters.tdb
{
key(21) = "GLOBALS/c_setprinter\00"
data(4) = "\00\00\00\00"
}
```

```

{
key(13) = "SECDESC/test\00"
data(140) = "\80\00\00\00\00\00\02\00\80\00\00\00\01\00\04\80\14\00\00\00$\00\00
\00\00\00\00\00\00\00\00\01\02\00\00\00\00\00\05 \00\00\00 \02\00\00\01\02\00\00
...
\00 \02\00\00\00\02\18\00\0C\00\0F\10\01\02\00\00\00\00\00\05 \00\00\00 \
02\00\00"
}
{
key(17) = "SECDESC/cups-pdf\00"
data(140) = "\80\00\00\00\00\00\02\00\80\00\00\00\01\00\04\80\14\00\00\00$\00\00
\00\00\00\00\00\00\00\00\01\02\00\00\00\00\00\05 \00\00\00 \02\00\00\01\02\00\00
...
\00 \02\00\00\00\02\18\00\0C\00\0F\10\01\02\00\00\00\00\00\05 \00\00\00 \
02\00\00"
}

```

Из листинга 2 видно, что база данных содержит три ключа. Первый ключ длиной в 21 байт (длина указывается в скобках) называется `GLOBALS/c_setprinter` и содержит в конце имени NULL-символ (ноль в кодировке ASCII). Непечатные символы отображаются в шестнадцатеричном формате (символ обратной косой черты, после которой следуют две шестнадцатеричные цифры). Значение первого ключа длиной в 4 байта состоит только из NULL-символов.

Следующие два ключа называются `SECDESC/test` и `SECDESC/cups-pdf`, и оба заканчиваются NULL-символом. Значения ключей хранятся в нечитаемом двоичном формате, поэтому они отображаются в виде шестнадцатеричных непечатных символов.

### **Создание резервных копий и восстановление TDB-файлов**

В [таблице 1](#) были перечислены несколько постоянных TDB-файлов, которые должны оставаться в системе после перезагрузки и для которых нужно создавать резервные копии. Как и в большинстве баз данных, нельзя просто скопировать файл, поскольку его копия может оказаться поврежденной. Это может случиться в тех случаях, если в файл, который вы копируете, записываются какие-либо данные; это приводит к несогласованному состоянию резервной копии. Другой способ создания резервных копий заключается в остановке демона Samba и последующем копировании файлов.

Самый простой способ создания резервных копий TDB-файлов – это использование утилиты `tdbbackup` из пакета Samba. Эта утилита может безопасно создавать копию TDB-файла, даже если в него записываются данные. Утилита `tdbbackup` также может проверять целостность TDB-файлов и автоматически восстанавливать их из резервных копий в случае обнаружения ошибок. В листинге 3 приведен пример создания резервной копии TDB-файла.

### **Листинг 3. Создание резервной копии TDB-файла**

```

[root@bob samba]# ls -l account_policy.*
-rw----- 1 root root 8192 Apr  7  2008 account_policy.tdb
[root@bob samba]# tdbbackup account_policy.tdb
[root@bob samba]# ls -l account_policy.*
-rw----- 1 root root 8192 Apr  7  2008 account_policy.tdb
-rw----- 1 root root 36864 Dec  8 21:42 account_policy.tdb.bak
[root@bob samba]# tdbdump account_policy.tdb | md5sum
53ea608f0d93061480549c511756b778 -
[root@bob samba]# tdbdump account_policy.tdb.bak | md5sum
53ea608f0d93061480549c511756b778 -

```

Первая команда в листинге 3 просто выводит список всех файлов, начинающихся с `account_policy`, с целью показать, что они существуют в единственном экземпляре. Затем с помощью команды `tdbbackup account_policy.tdb` создается резервная копия базы политик учетных записей. Третья команда снова выводит список всех файлов, начинающихся с `account_policy`, с целью показать, что был создан новый файл с расширением `.bak`. Размер резервной копии отличается от размера оригинала, но если сделать дампы каждого файла и посчитать их MD5-хэши, то можно убедиться в том, что их контрольные суммы совпадают. Больший размер файла не является проблемой, поскольку содержимое каждой пары "ключ-значение" идентично.

Если по какой-либо причине исходный файл `account_policy.tdb` будет поврежден (например, при аварийной перезагрузке компьютера), можно восстановить его из резервной копии. Эта процедура показана в листинге 4.

#### Листинг 4. Проверка и восстановление TDB-файла

```
[root@bob samba]# ls -l account_policy.tdb
-rw----- 1 root root 1213 Dec  8 21:49 account_policy.tdb
[root@bob samba]# tdbbackup -v account_policy.tdb
tdb_oob len 1256 beyond eof at 1213
restoring account_policy.tdb
[root@bob samba]# ls -l account_policy.tdb*
-rw----- 1 root root 36864 Dec  8 21:49 account_policy.tdb
-rw----- 1 root root 36864 Dec  8 21:42 account_policy.tdb.bak
[root@bob samba]# tdbbackup -v account_policy.tdb
account_policy.tdb : 17 records
[root@bob samba]# tdbdump account_policy.tdb | md5sum
53ea608f0d93061480549c511756b778 -
```

Первая команда в листинге 4 показывает, что файл сильно уменьшился в размере. Затем снова запускается команда `tdbbackup` с флагом `-V` для проверки целостности TDB-файла. Если файл окажется поврежденным, то вы увидите список ошибок с последующим сообщением о том, что данный файл был восстановлен. Сравнив размеры файлов с помощью команды `ls`, можно увидеть, что текущая база данных была замещена своей резервной копией.

Команду `tdbbackup` можно безбоязненно запускать несколько раз. Если эта команда работает с неповрежденным файлом базы данных (например, предпоследняя команда в листинге 4), то она просто выводит количество строк в файле. Команда MD5 также показывает нам ту контрольную сумму, которую мы видели до того, как файл был поврежден.

**Примечание.** Команда `tdbbackup` поддерживает использование групповых символов, поэтому вы можете резервировать или восстанавливать несколько файлов одновременно.

#### Изменение TDB-файлов

Утилита `tdbtool` позволяет изменять данные внутри TDB-файлов, а также смотреть отдельные ключи и значения прямо в файле, избавляя от необходимости делать его дамп и сортировать вывод.

## Предупреждение

При работе с Samba редко возникает необходимость изменения данных внутри TDB-файла, поскольку либо файлы содержат структуры внутренних данных, либо для этого существуют более удобные инструменты. Тем не менее, рассмотрение этой темы входит в цель экзамена,

поэтому необходимо рассказать о ней.

Команда `tdbtool` может принимать команды из командной строки или работать в интерактивном режиме. Чтобы выполнить все необходимые действия в командной строке, запустите команду `tdbtool example.tdb` команда опции, где `example.tdb` – это имя вашего файла, а команда – команда, после которой идут все необходимые опции. Для использования оболочки `tdb` просто запустите команду `tdbtool`, по желанию указав в командной строке имя файла.

Для создания базы данных просто запустите `tdbtool`, а затем введите команду `create test.tdb`. В результате на жестком диске будет создана база данных `test.tdb`, открытая для записи в рамках текущего сеанса работы. Если у вас уже имеется TDB-файл, можно указать в командной строке его имя или использовать команду `open`. Интересно, что единственный способ создания TDB-файла непосредственно из командной строки требует указать имя дважды, например, `tdbtool test.tdb create test.tdb`, в результате чего выдается ошибка, но база данных успешно создается.

После создания новой базы данных или открытия существующей доступны следующие опции:

- **dump**: полностью показывает содержимое базы данных, как это делает `tdbdump`.
- **keys**: показывает только ключи в кодировке ASCII или (вместе с командой `hexkeys`) в шестнадцатеричном формате.
- **erase**: полностью удаляет содержимое базы данных без подтверждения.
- **info**: показывает общее количество ключей и байтов в базе данных.
- **check**: проверяет целостность базы данных.
- **speed**: оценивает быстродействие операций чтения и записи в базу данных.
- **show key**: выводит значения указанного ключа.
- **delete key**: удаляет ключ вместе с его значением.

Для добавления и управления данными используются команды `insert` и `store`. Каждая из этих команд принимает параметры для ключа и значения (параметры для значения являются необязательными). Если не указывать значение ключа, то оно будет содержать нулевой байт данных.

*Вставка записи* (операция `insert`) означает создание новой записи, тогда как при *сохранении записи* (операция `store`) ее старое значение замещается новым. Отличия этих команд продемонстрированы в листинге 5.

#### Листинг 5. Вставка и сохранение записей

```
tdb> insert mykey myvalue
tdb> insert mykey newvalue
insert failed
tdb> store mykey newvalue
Storing key:
key 5 bytes
mykey
data 8 bytes
[000] 6E 65 77 76 61 6C 75 65          newvalue
tdb> store newkey someothervalue
Storing key:
key 6 bytes
newkey
data 14 bytes
[000] 73 6F 6D 65 6F 74 68 65 72 76 61 6C 75 65      someothe rvalue
```

В листинге 5 выполняется следующая последовательность действий:

1. Вставка нового ключа с именем `mykey` и значением `myvalue`. Эта операция выполнена успешно.
2. Вставка ключа с тем же именем, но с другим значением. Эта операция завершилась с ошибкой, поскольку ключ с таким именем уже существует.
3. Вместо вставки ключа используется операция сохранения. Эта операция выполнена успешно с более подробным выводом.
4. Сохранение нового ключа с новым значением. Эта операция выполнена успешно, даже не смотря на то, что ключа с таким именем не существует.

В командной оболочке `tdb` можно использовать транзакции, позволяющие выполнять последовательность команд и применять либо отклонять их как единую группу. В листинге 6 приведен пример выполнения двух транзакций.

#### Листинг 6. Использование транзакций

```
tdb> transaction_start
tdb> insert somekey somevalue
tdb> show somekey

key 7 bytes
somekey
data 9 bytes
[000] 73 6F 6D 65 76 61 6C 75 65           somevalu e
tdb> transaction_cancel
tdb> show somekey
fetch failed
tdb> transaction_start
tdb> insert somekey somevalue
tdb> transaction_commit
tdb> show somekey

key 7 bytes
somekey
data 9 bytes
[000] 73 6F 6D 65 76 61 6C 75 65           somevalu e
```

Начало транзакции в листинге 6 определяется командой `transaction_start`. Затем выполняется вставка нового ключа. Если бы другой процесс в это время считывал данные из БД, то он не увидел бы этот ключ, поскольку транзакция еще не подтверждена (процесс с открытой транзакцией также не видит новый ключ). Затем транзакция отменяется командой `transaction_cancel`. Мы видим, что ключ отсутствует в базе данных.

После этого выполняется та же последовательность действий, но транзакция не отменяется, а подтверждается командой `transaction_commit`. Теперь ключ существует в базе данных, и его видят все пользователи.

Если база данных находится в режиме транзакции, доступ для других пользователей может блокироваться; это означает, что они будут ждать окончания транзакции. В связи с этим будьте аккуратны при использовании транзакций в рабочей базе данных! Несмотря на то, что транзакции являются мощным средством защиты данных, их чрезмерное использование потенциально может влиять на быстродействие.

## Что можно использовать помимо TDB

В зависимости от конфигурации учетные записи пользователей могут храниться в различных местах. Существуют две утилиты, которые позволяют настраивать взаимодействие с хранилищами данных с помощью интерфейса командной строки. Например, при помощи одних и тех же команд можно обращаться к данным, хранящимся как в TDB-файлах, так и в каталоге LDAP.

### Использование smbpasswd

Утилита `smbpasswd` позволяет добавлять и удалять учетные записи пользователей и компьютеров, а также изменять пароли. Наиболее часто она используется именно для смены пользователями своих паролей, либо для смены администраторами паролей других пользователей.

В следующих статьях вы более подробно узнаете о различных хранилищах паролей, но на высоком уровне пароли Samba могут храниться по-разному в зависимости от версии Samba и операционных систем, с которыми она интегрируется. И `smbpasswd`, и `pdredit` (описывается в следующем разделе) могут работать с любыми хранилищами, в том числе, отличными от TDB.

Клиенты Microsoft передают пароли по сети в виде собственных Microsoft-хэшей, а не в открытом виде или в виде UNIX-хэшей. Это означает, что невозможно взять Microsoft-хэш пароля и сопоставить его с паролем, хранящимся в базе данных паролей UNIX. Именно поэтому для хранения Microsoft-хэшей Samba должна использовать отдельную базу данных, которая называется *хранилищем паролей*.

### Использование pdredit

Утилита `pdredit` управляет базой данных пользователей и политиками учетных записей Samba. Она умеет выполнять все те же действия, что и `smbpasswd`, а также управлять политиками и переносить учетные данные между различными хранилищами.

Чтобы получить полный список пользователей, хранящихся в базе данных, запустите команду `pdredit -L`. Указав флаг `-v`, вы получите еще более подробную информацию о пользователях, как показано в листинге 7.

### Листинг 7. Подробная информация о пользователях

```
[root@bob tmp]# pdredit -L -v
-----
Unix username:      sean
NT username:
Account Flags:      [U          ]
User SID:           S-1-5-21-2287037134-1443008385-640796334-1001
Primary Group SID: S-1-5-21-2287037134-1443008385-640796334-513
Full Name:          Sean
Home Directory:    \\bob\sean
HomeDir Drive:
Logon Script:
Profile Path:       \\bob\sean\profile
Domain:             BOB
Account desc:
Workstations:
Munged dial:
Logon time:         0
Logoff time:        never
Kickoff time:       never
Password last set: Mon, 24 May 2010 21:28:49 CDT
Password can change: Mon, 24 May 2010 21:28:49 CDT
```

```
Password must change: never
Last bad password : 0
Bad password count : 0
Logon hours : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

Для работы с утилитой `pdbedit` необходимо обладать привилегиями пользователя `root`. С ее помощью можете изменять различные настройки пользователей (например, разрешенное время входа в систему), пароли и домашние директории, как если бы вы работали с утилитами Microsoft.

Для каждого параметра пользователя используются различные опции командной строки, поэтому для получения точной информации обратитесь к `man`-странице `pdbedit(8)`.

Также важно упомянуть о том, что Samba воспринимает учетные записи пользователей и компьютеров одинаково. В листинге 8 показана база данных паролей сервера Samba, настроенного в качестве контроллера домена.

### Листинг 8. База данных паролей, содержащая пароли компьютеров

```
[root@sergeant ~]# pdbedit -L
root:0:root
mythupstairs$:4294967295:MYTHUPSTAIRS
BOB$:1043:Machine
sean:1002:Sean,,,
sergeant$:4294967295:Machine
```

Имена, оканчивающиеся знаком доллара (\$) – это учетные записи компьютеров, используемые для аутентификации компьютеров в домене. Соответствующий секретный ключ может храниться в файле `secrets.tdb` на удаленном сервере и в файле `passdb.tdb` на контроллере домена.

### Что дальше

Эта статья завершает тему 310 "Архитектура, принципы и схема работы". Следующая статья открывает новую тему и содержит материалы цели 311.1 темы 311 ("Компиляция и установка Samba"). В ней рассказывается, как загрузить исходный код Samba и скомпилировать его.

### Ресурсы

#### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Trivial Database files \(EN\)](#).
- Прочтите о различных [хранилищах паролей \(EN\)](#) и узнайте больше о том, как работают утилиты `smbpasswd` и `pdbedit`.
- Изучите протокол [NT LAN Manager](#) и узнайте, какие данные передаются по сети при аутентификации пользователей.
- [Глава 41 \(EN\)](#) руководства Samba содержит подробную информацию о базах данных TDB и о том, как устранять связанные с ними проблемы.
- На Web-сайте [программы сертификации LPIC \(EN\)](#) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302 \(EN\)](#), а также [примеры](#)

[заданий и вопросов](#) (EN).

- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- В [разделе Linux сайта developerWorks](#) можно найти сотни [пошаговых инструкций и руководств](#), загрузить программные продукты, а также получить ссылки на форумы и многие другие ресурсы, ориентированные на разработчиков и администраторов Linux.
- Смотрите [демонстрационные материалы по запросу на сайте developerWorks](#) (EN), ориентированные как на новичков, так и на опытных разработчиков.

## **Получить продукты и технологии**

- Загрузите [Samba](#) (EN) и следите за последними новостями разработки проекта.
- Загрузите [исходный код TDB](#) (EN), если вы планируете использовать TDB в ваших проектах.
- Если вам необходим графический интерфейс или более глубокая интеграция с различными средами, взгляните на [инструменты для работы с учетными записями](#) (EN) для Samba.

# Изучаем Linux, 302 (смешанные среды): Конфигурирование и компиляция Samba из исходного кода

*Собираем Samba с чистого листа*

[Родерик Смит \(Roderick Smith\)](#), автор и консультант, IBM

**Описание:** Как и большинство других программ для Linux, Samba является Open Source-проектом, поэтому можно бесплатно загрузить файлы с исходным кодом и скомпилировать из них двоичный пакет для своей системы. В результате всегда можно получить более новые версии программного обеспечения, чем те, которые включены в дистрибутив, настраивать различные опции компиляции, настраивать компилятор на оптимальную производительность и даже изменять исходный код. О том, как это сделать, вы узнаете из этой статьи.

[Больше статей из этой серии](#)

**Дата:** 05.04.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Идентификация важных пакетов Samba и их содержимого.
- Поиск и установка приложений, от которых зависит Samba.
- Описание структуры программного обеспечения Samba.
- Определение важных параметров компиляции Samba.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 311.1 темы 311. Цель имеет вес 1.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и в общих чертах понимает структуру программного обеспечения (исходный и двоичный код) и процедуры компиляции. Для выполнения примеров этой статьи ваш компьютер должен быть подключен к Интернету и на нем должен быть проинсталлирован компилятор C (например, GNU Compiler Collection [GCC]).

## Получение исходного кода Samba

Исходный код Samba можно бесплатно загрузить с Web-сайта Samba (раздел [Ресурсы](#)). Если вы уже знаете, как скомпилировать и проинсталлировать программное обеспечение из исходного кода, то у вас не должно возникнуть проблем с Samba. Тем не менее, поскольку во многих случаях Samba очень важна для работы в сетевых сервисов, то необходимо убедиться в том, что вы загрузили именно то программное обеспечение, которое вам требуется. С этой целью разработчики Samba предоставляют ключи аутентификации, которые можно использовать для проверки загруженного дистрибутива.

### Загрузка исходного tarball-файла

Исходный код Samba можно бесплатно загрузить с Web-сайта Samba. Ссылка на последнюю стабильную версию (на момент написания статьи это версия 3.5.6) расположена на его главной странице. Щелкните по ссылке, чтобы получить эту версию дистрибутива Samba; tarball-файл будет помещен в директорию загрузок вашего браузера. Ссылка на загрузку Samba следующей, четвертой версии, также расположена на главной странице Web-сайта, однако учтите, что эта версия находится в стадии альфа-тестирования уже много лет, поэтому используйте ее, только если вам необходимы ее возможности или если вы хотите участвовать в разработке Samba.

На странице загрузки Samba также имеются ссылки на архивы старых версий Samba, ссылки на инструкции по загрузке ПО с помощью Git, Control Version System и других инструментов, ссылки на различные вспомогательные утилиты, например, менеджеры GUI, пакеты SMB/CIFS для других платформ и т. д. В частности, там есть одна особенно важная ссылка на ключ GnuPG (GPG) для пакета Samba. Если вы хотите проверить подлинность пакета Samba, как это будет описано в следующем разделе, необходимо скачать этот файл (`samba-pubkey.asc`).

### Проверка подлинности пакета Samba

Можно распаковать, скомпилировать, проинсталлировать и использовать Samba, не проверяя подлинность исходного пакета. Однако в силу критической важности сервера Samba, выполняющего множество различных команд, лучше проверять его целостность. Для этого выполните следующие действия:

1. Убедитесь в том, что файлы `samba-version.tar.asc` и `samba-pubkey.asc` загружены.
2. Выполните команду `gunzip samba-version.tar.gz` чтобы разархивировать полученный файл (но не извлечь содержимое).
3. Выполните команду `gpg --import samba-pubkey.asc`, чтобы импортировать публичный ключ Samba в кольцо для ключей GPG, если вы еще не сделали этого.
4. Выполните команду `gpg --verify samba-version.tar.asc` для проверки ключа. Вы должны получить примерно следующее сообщение:  

```
gpg: Signature made Thu 07 Oct 2010 02:23:24 PM EDT using DSA key ID 6568B7EA
gpg: Good signature from "Samba Distribution Verification Key
<samba-bugs@samba.org\>"
```

Скорее всего, сообщение также будет содержать информацию о том, что ключ не подписан доверенной подписью. Это говорит о следующем ограничении процедуры проверки: если предположить, что главный Web-сайт Samba был скомпрометирован и злоумышленник подменил установочный пакет Samba и оба ключа поддельными версиями, то вы будете доверять обоим этим ключам, думая, что они подлинные. Для дополнительной защиты на шаге 4 можно запустить программу `gpg` с опцией `--keyserver wwwkeys.pgp.net`. Эта опция говорит программе `gpg` о том, что в процессе проверки подлинности необходимо получить ключ с Web-ресурса `wwwkeys.gpg.net`. Тогда, если кто-то захочет изменить и

распространять поддельный дистрибутив Samba, то ему придется скомпрометировать не только Web-сервер Samba, но и Web-сервер ключей.

## Распаковка tarball-файла

После того, как подлинность загруженного пакета успешно установлена (или если вы просто пропустили этот шаг), можно распаковать tarball-файл с исходным кодом. Это можно сделать, находясь в домашней директории или, например, в директории `/usr/src/`, предназначеннной для хранения исходного кода локальных приложений. Если вы распаковываете исходный код в директорию `/usr/src/`, вам могут потребоваться привилегии пользователя `root`. Также они могут потребоваться для предоставления прав на запись в эту директорию обычным пользователям.

Какой бы вариант вы не выбрали, перейдите в корень директории, в которую вы будете распаковывать файл с исходным кодом, и выполните следующую команду:

```
$ tar xvf ~/samba-version.tar
```

В этой команде предполагается, что вы уже разархивировали tarball с помощью `gunzip` и поместили его в корень вашей домашней директории. Если же вы еще не сделали этого, то можно либо предварительно разархивировать его, либо добавить команду `Z` в список команд `tar` и указать дополнительное расширение в имени, как показано ниже:

```
$ tar xvzf ~/samba-version.tar.gz
```

Если файл расположен не в корне вашей домашней директории, а в другой директории, то укажите путь к этой директории. Ну и, конечно, необходимо указывать реальное имя файла загруженного пакета определенной версии.

Эта команда выводит на экран список извлекаемых файлов. Если вы увидите сообщения об ошибке, то это может означать, что у вас нет разрешений на запись в текущую директорию или на жестком диске не осталось свободного места. После успешного выполнения этой команды вы увидите новую директорию с именем `samba-версия`. Это и есть дерево исходного кода Samba.

## Компиляция Samba

Теперь, когда у нас имеется исходный код, можно приступить к его компиляции. Однако прежде чем начать, убедитесь, что ваш компьютер удовлетворяет всем предварительным требованиям, перечисленным в начале статьи. На нем должно быть инсталлировано и настроено все необходимое программное обеспечение, а также вы должны быть готовы к устранению проблем, которые могут возникнуть в процессе компиляции.

## Установка требуемого программного обеспечения

Для компиляции Samba должны быть установлены несколько других пакетов, самый важный из которых – это `GCC`. `GCC` – это набор компиляторов для языка `C`, на котором написана большая часть Samba. В большинстве дистрибутивов Linux `GCC` можно установить из соответствующего пакета с именем `gcc`. Другим критически важным инструментом является утилита `make`, которая обращается к `gcc` и другим инструментам разработки в соответствии с правилами, определенными разработчиками Samba.

Samba использует несколько *библиотек* – программных пакетов, обеспечивающих поддержку вспомогательных функций для других программ. Вполне возможно, что эти библиотеки уже установлены на вашем компьютере, но для компиляции программы требуются *файлы заголовков* библиотек, которые зачастую устанавливаются в виде отдельных пакетов с

именами, заканчивающимися на *-dev* или *-devel*. Как минимум, используйте инструменты вашего дистрибутива для работы с пакетами, чтобы убедиться в том, что на компьютере установлены библиотеки разработчика *libc* или *libc6*. В зависимости от настроек конфигурации некоторые библиотеки могут потребоваться, а некоторые нет. В случае отсутствия нужной библиотеки сценарий *configure* (а, возможно, сама процедура компиляции) выдаст сообщение об ошибке, позволяющее выяснить, какая библиотека необходима.

Многие дистрибутивы упрощают установку основных инструментов и библиотек разработчиков, позволяя выполнить ее за один шаг. В Ubuntu, например, для этого нужно просто установить пакет **build-essential**. Такие дистрибутивы, как, например, Fedora, позволяют выбирать пакеты для установки во время установки операционной системы, поэтому, если вы собираетесь использовать компьютер для разработки программного обеспечения, вы можете заранее установить сразу все нужные пакеты. Если вы не можете найти такие опции в вашей операционной системе, вам придется устанавливать пакеты постепенно по мере необходимости.

### Настройка конфигурации Samba

Для настройки конфигурации Samba необходимо перейти в поддиректорию *source3* основной директории, содержащей исходный код Samba. Эта поддиректория содержит исходный код основного пакета Samba.

**Примечание.** Остальные поддиректории дерева содержат исходный код дополнительных и вспомогательных программ, документации и т. д. Например, поддиректория *client* содержит файлы, обеспечивающие возможность монтирования операционной системой Linux общих ресурсов SMB/CIFS на клиентах, а поддиректория *swat* – исходный код Samba Web Administration Tool (SWAT). Какие-то из этих программ компилируются в процессе компиляции основного пакета Samba, а какие-то вам придется скомпилировать самостоятельно. Результатом компиляции основного пакета Samba в поддиректории *source3* являются главные программы сервера – демоны *smbd* и *nmbd*, а также вспомогательные библиотеки и многочисленные вспомогательные инструменты.

Процессом конфигурации управляет сценарий *configure*. В простейшем случае можно запустить команду *./configure*, которая настроит Samba с параметрами по умолчанию, однако для настройки Samba на этапе компиляции существует довольно много опций. Их полный список можно увидеть, выполнив команду *./configure --help* (если вы работаете с консолью, которая не пролистывается, можно перенаправить вывод в файл и затем просматривать его в текстовом редакторе или с помощью команды *less*).

Также вы можете использовать различные переменные окружения, список которых также выводится по команде *./configure --help*. В большинстве случаев значения по умолчанию обеспечивают хорошую работу, тем не менее, если вы хорошо знаете о важных параметрах системы, вы можете настроить эти переменные в соответствии с вашими задачами. Например, вы можете задать значение переменной окружения *CFLAGS*, чтобы установить флаги компилятора C. Если вы не знаете, для чего нужна та или иная переменная окружения, то лучше оставьте все как есть.

После того как вы внимательно изучили все опции и переменные окружения, можно приступить к конфигурированию Samba:

```
$ CFLAGS="-O3" ./configure --without-ldap
```

В этом примере мы используем флаг компилятора GCC **-O3** и компилируем Samba без поддержки протокола LDAP (Lightweight Directory Access Protocol). Разумеется, в вашем

случае вы можете изменить или вообще не использовать эти опции.

## Компиляция Samba

Фактически, чтобы скомпилировать программу, от вас потребуется выполнить лишь одно действие – ввести следующую команду:

```
$ make
```

В результате вы увидите на экране строки с информацией о действиях, выполняемых утилитой `make`. Эти строки имеют следующий вид:

```
Compiling lib/netapi/joindomain.c  
Compiling lib/netapi/serverinfo.c  
Compiling lib/netapi/getdc.c
```

Будем надеяться, что процесс компиляции завершится без ошибок. Обычно он продолжается несколько минут в зависимости от мощности компьютера. Если у вас компьютер с многоядерным процессором, можно использовать команду `make` с опцией `-j`, которая будет выполнять компиляцию в несколько потоков, ускоряя процесс. Например, команда `make -j 4` будет одновременно компилировать до четырех файлов с исходным кодом. После того, как компиляция Samba будет завершена, выполните следующую команду для ее инсталляции:

```
# make install
```

Хотя все предыдущие действия может выполнить обычный пользователь (если, конечно, у него есть доступ на запись в корневую директорию, содержащую исходный код), для выполнения команды `make install` необходимо обладать привилегиями пользователя `root`, поскольку эта она копирует двоичные файлы и документацию Samba в системные директории (обычно это директория `/usr/local/`, если вы не изменили ее с помощью сценария `configure`).

## Устранение ошибок

К сожалению, иногда процессы конфигурации или компиляции завершаются с ошибкой. Наиболее распространенная причина заключается в отсутствии необходимой библиотеки. Если ошибка возникает на этапе конфигурирования, то, вероятно, в конце вывода должно содержаться сообщение об ошибке, указывающее на отсутствующий компонент; например, в сообщении может говориться о том, что не были найдены библиотеки PAM (Pluggable Authentication Module – подключаемый модуль аутентификации). Для поиска и инсталляции необходимого программного обеспечения следует использовать инструменты для работы с пакетами, например, Synaptic или Yumex. Помните о том, что библиотеки разработчика могут устанавливаться отдельно от основного пакета с библиотеками.

Точно так же можно устранять ошибки на этапе работы команды `make`; однако при возникновении ошибок на этапе компиляции часто выводится большое число сообщений. Если вы столкнулись с такой проблемой, то не обращайте внимания на последние сообщения, а сразу пролистайте вывод назад к первому сообщению об ошибке. Зачастую одна ошибка приводит к другой, и так далее по цепочке. Исправление первой ошибки может устраниТЬ всю последующую цепочку, в результате чего процесс компиляции успешно завершится.

Устранение ошибок команды `make` может потребовать больше усилий, чем устранение ошибок на этапе конфигурирования, поскольку ошибки компиляции, скорее всего, связаны с

отсутствующими файлами или даже неверным синтаксисом в определенном файле. Если проблема заключается в отсутствующем файле, то его имя может послужить подсказкой – попробуйте поискать пакет с таким именем (без расширения) среди пакетов вашего дистрибутива и, возможно, вы определите недостающую библиотеку разработчика, которую забыл отметить сценарий `configure`. Если это не помогло, попробуйте поискать имя файла в Интернете – возможно, это поможет определить имя библиотеки, которая должна быть проинсталлирована. Если проблема связана с неправильным синтаксисом, то это уже более серьезное дело. Вы должны суметь отключить проблемную функцию с помощью соответствующей опции сценария `configure`; возможно, вам потребуется обновить версию компилятора C (или наоборот, откатиться на более старую) или версию Samba, которую вы пытаетесь компилировать. Если вы работаете с предварительным релизом (prerelease) программного обеспечения, то вы можете столкнуться с программной ошибкой, которая потребует от вас самостоятельного внесения исправлений. Эти действия выходят за рамки нашей статьи, поэтому в таких случаях вы можете обратиться за помощью к разработчикам Samba или к опытным программистам.

## Что дальше

Следующая статья этой серии содержит материалы цели 311.2 темы 311. В ней рассматривается процесс установки Samba из исходного кода и двоичных пакетов. В задачи этой цели входит запуск серверных программ Samba – демонов `smbd` и `nmbd` (остальные компоненты Samba выполняют вспомогательные задачи, например, компонент SWAT предназначен для настройки Samba с помощью Web-интерфейса).

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Configure and build Samba from source](#) (EN).
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Чтобы получить ссылки на все статьи этой серии, которые помогут вам подготовиться к успешной сдаче экзамена LPI-302, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#).
- Следите на последними новостями на портале [Web-трансляций и технических мероприятий developerWorks](#) (EN), относящимися к продуктам IBM и событиям ИТ-индустрии.
- Посещайте [бесплатные брифинги developerWorks Live!](#) (EN), чтобы быть в курсе последних событий, относящихся к продуктам IBM и направлениям в ИТ-индустрии.
- Смотрите [демонстрационные материалы по запросу на сайте developerWorks](#) (EN), ориентированные как на новичков, так и на опытных разработчиков.

### Получить продукты и технологии

- Вы можете загрузить систему Samba и получить информацию о ней на [Web-сайте Samba](#) (EN).
- Дополнительные опции загрузки доступны на [странице загрузки](#) (EN) Web-сайта Samba.

- На [Web-сайте GCC](#) (EN) вы найдете информацию и ссылки на загрузку GNU Compiler Collection и связанных инструментов и библиотек.
- На [Web-странице GNU make](#) (EN) вы найдете информацию и ссылки на загрузку утилиты GNU make.

# Изучаем Linux, 302 (смешанные среды): Инсталляция и обновление Samba

*Запускаем Samba*

[Родерик Смит \(Roderick Smith\)](#), автор и консультант, IBM

**Описание:** Для использования Samba надо сначала инсталлировать это программное обеспечение. Это можно сделать несколькими способами, но все они сводятся к двум основным методам: компиляция и инсталляция из исходного кода либо установка из предварительно собранных пакетов двоичного кода. Первый метод более сложный, но зато обеспечивает большую гибкость, тогда как второй метод намного проще для большинства дистрибутивов, но при этом авторы пакетов сами решают, какие версии будут поддерживаться, какие обновления будут установлены, какие параметры будут использованы на этапе компиляции и т. д.

[Больше статей из этой серии](#)

**Дата:** 10.04.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Инсталляция пакетов Samba.
- Инсталляция самостоятельно скомпилированных двоичных файлов Samba.
- Обновление уже инсталлированной Samba.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 311.2 темы 311. Цель имеет вес 1.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой в Linux и использовать инструменты для работы с пакетами Linux, и в общих чертах понимает структуру программного обеспечения (исходный и двоичный код). Для выполнения примеров этой статьи ваш компьютер должен быть подключен к Интернету или вы должны иметь инсталляционный диск Linux, содержащий

пакеты Samba.

## Выбор метода инсталляции

Выбор метода инсталляции Samba зависит от дистрибутива Linux и доступных инструментов, а также от требований, предъявляемых к определенным версиям и функциональным возможностям Samba. Некоторые методы инсталляции могут оказаться недоступными для отдельных дистрибутивов Linux. Хотя инсталляция из RPM- и Debian-пакетов с помощью графических средств является самым простым и наилучшим способом, только установка из исходного кода доступна во всех без исключения Linux-системах, причем этот метод может потребовать инсталляции дополнительного программного обеспечения.

Большинство дистрибутивов Linux содержат инструменты для работы либо с RPM-, либо с Debian-пакетами. В Red Hat, Fedora, OpenSUSE, Mandriva, PCLinuxOS и некоторых других операционных системах используются RPM-пакеты, тогда как в Debian, Ubuntu и некоторых других – Debian-пакеты. При использовании одного из вышеперечисленных дистрибутивов самый простой способ инсталляции Samba – это инсталляция на основе пакета двоичного кода, поставляемого создателем дистрибутива. Такой пакет можно установить, выполнив одну (возможно, несколько) простую команду, после чего будет запущен процесс инсталляции, который обычно занимает несколько секунд. Некоторые дистрибутивы, например, Slackware, используют для установки программ пакеты других типов; для работы с ними используются команды, отличные от команд для работы с RPM- и Debian-пакетами, которые мы будем рассматривать в этой статье.

Установка из исходного кода позволяет настраивать параметры Samba и оптимизировать процесс компиляции с учетом возможностей и сетевых подключений вашего компьютера. В этом случае можно также установить версию Samba, которая может быть еще не доступна для вашего дистрибутива. При установке из исходного кода требуется выполнить ряд дополнительных действий, а сам процесс установки может оказаться более продолжительным, чем процесс установки из двоичного пакета. В операционной системе Gentoo большинство программ устанавливается из исходного кода, но при этом используется усовершенствованная процедура, поэтому этот способ похож на использование RPM- или Debian-пакетов (за подробной информацией обратитесь к документации Gentoo).

В большинстве случаев следует инсталлировать Samba из RPM- или Debian-пакета, либо из другого двоичного файла, предназначенного для того или иного дистрибутива Linux. Инсталлировать Samba из исходного кода имеет смысл тогда, когда невозможно сделать это другим способом или если перед вами стоят нестандартные задачи, требующие дополнительных настроек на этапе компиляции.

## Инсталляция из исходного кода

В [предыдущей статье этой серии](#) рассказывалось о компиляции Samba из исходного кода; сначала необходимо выполнить это действие, если вы намерены инсталлировать Samba на основе исходного кода. В этой статье предполагается, что вы уже скомпилировали исходный код, и вам осталось лишь инсталлировать Samba.

### Первоначальная установка

Если исходный код системы Samba уже скомпилирован, ее можно установить, выполнив в директории с исходным кодом (обычно это поддиректория source3 дерева исходного кода Samba) следующую команду:

```
# make install
```

Эту команду необходимо выполнять из-под учетной записи пользователя root.

В результате выполнения этой команды Samba, как правило, инсталлируется в директорию `/usr/local`, которая обычно содержит все локально скомпилированные двоичные файлы.

Обратите внимание на то, что при инсталляции Samba на базе исходного кода *не* инсталлируется сценарий запуска подсистемы System V (SysV) или Upstart, поэтому Samba не будет запущена автоматически после перезагрузки компьютера. Мы кратко рассмотрим этот вопрос в разделе [Запуск Samba](#).

### Обновление Samba до новой версии из исходного кода

Если Samba была инсталлирована на основе исходного кода, то в результате выполнения следующей процедуры все старые программные файлы будут переименованы в файлы с расширением `.old`. Если окажется, что новая версия не работает, то команда `make revert` позволяет вернуться к старой версии Samba.

Если вы хотите полностью удалить старую версию Samba, инсталлированную на базе исходного кода, то следует сделать *именно ее* текущей версией и выполнить команду `make uninstall`. Эта команда полностью удаляет инсталлированное приложение. После этого можно будет инсталлировать новую версию (как из исходного кода, так и из двоичного пакета), не опасаясь конфликтов версий.

Если до этого вы инсталлировали Samba на базе двоичного пакета, то теоретически эта версия и версия, которая будет скомпилирована локально из исходного кода, смогут сосуществовать на компьютере одновременно; тем не менее, это может привести к путанице, поскольку, по всей вероятности, работать будет только одна из них. Таким образом, лучше всего будет удалить старую версию, инсталлированную на базе двоичного пакета, прежде чем устанавливать новую версию. RPM- и Debian-пакеты Samba можно удалить с помощью команд `rpm -e samba` и `dpkg -r samba`, соответственно (может потребоваться указать пакет с другим именем или удалить несколько пакетов в зависимости от того, как ваш дистрибутив создал пакеты Samba). Перед деинсталляцией двоичного пакета Samba вы можете сохранить сценарий запуска подсистем System V (SysV) или Upstart; возможно, вам удастся изменить его таким образом, чтобы он запускал локально скомпилированную версию Samba.

### Установка из RPM-пакета

RPM – это популярная и довольно мощная система управления пакетами. Можно устанавливать приложения, загружая RPM-файлы и используя команду `rpm` для их установки, либо воспользовавшись системой управления мета-пакетами (например, YUM – Yellowdog Updater, Modified), которая будет выполнять за вас такие скучные действия, как инсталляция или обновление зависимых пакетов.

### Инсталляция пакетов с помощью YUM

YUM – это стандартный инструмент Red Hat, Fedora и некоторых других дистрибутивов Linux на основе RPM. В состав некоторых из них включены другие инструменты с похожей функциональностью

Для инсталляции пакета с помощью YUM нужно запустить из-под учетной записи пользователя `root` команду `yum`, указав подкоманду и имя устанавливаемого пакета:

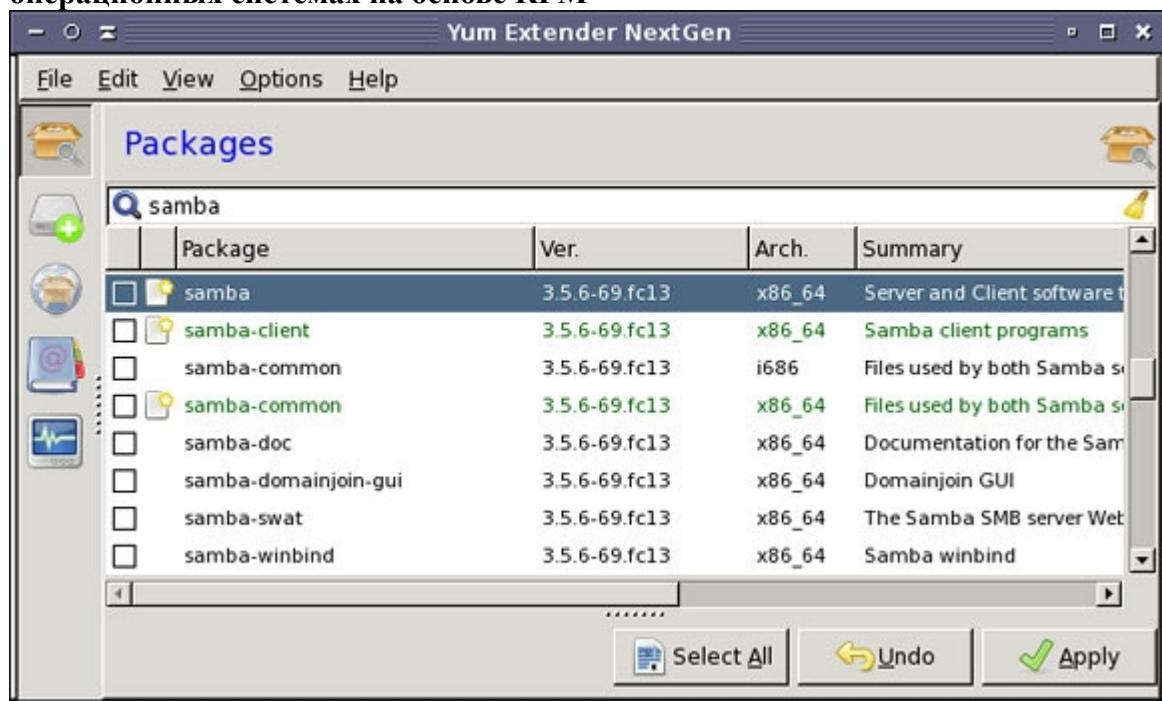
```
# yum install samba
```

**Примечание.** Пакеты Samba в разных дистрибутивах могут называться по-разному. Возможно, вам придется инсталлировать не пакет `samba`, а пакет с другим именем, например, `samba-server`. В операционной системе Fedora этот пакет так и называется – `samba`.

После того, как вы набрали эту команду, YUM проверяет репозитории, загружает последние версии одного или нескольких пакетов и инсталлирует их. В некоторых случаях в результате выполнения этой команды инсталлируются несколько пакетов Samba или различные зависимые пакеты, не связанные с Samba. Например, в операционной системе Fedora при инсталляции пакета **samba** также устанавливаются пакеты **samba-common** и **samba-client**.

Для поиска и установки Samba или связанных с ней пакетов можно использовать и графические утилиты YUM, например, Yumex (Yum Extender, команда `yumex`), как показано на рисунке 1. Yumex и другие графические утилиты могут быть особенно полезны при поиске пакетов, относящихся к Samba, например, пакета Samba Web Administration Tool (SWAT; `samba-swat`) показанного на рисунке 1.

**Рисунок 1. Графический интерфейс Yumex для управления пакетами в некоторых операционных системах на основе RPM**



### Инсталляция пакетов с помощью RPM

Некоторые дистрибутивы не поддерживают использование YUM, однако вам может потребоваться инсталлировать RPM-пакет, загруженный вручную с какого-либо Web-сайта. Например, это может быть пакет с более новой версией приложения, которая еще не была добавлена в репозиторий вашего дистрибутива. В таких случаях для инсталляции приложения можно использовать утилиту `rpm`.

По возможности перед инсталляцией пакета следует проверять его подлинность с помощью команды `gpg`, как рассказывалось в [предыдущей статье](#) этой серии. После проверки подлинности пакета (вы можете не выполнять ее) запустите команду `rpm` с опцией `--install` (или `-i`). Можно также добавить опции `--verbose` (`-v`) и `--hash` (`-h`), чтобы отслеживать процесс инсталляции пакета. В итоге ваша команда должна выглядеть примерно так:

```
# rpm -ivh samba-3.5.6-69.fc13.x86_64.rpm
```

Конечно же, вы должны указать в этой команде имя загруженного вами файла. Если в

процессе инсталляции вы получили сообщение об ошибке, то вам придется решать проблему самостоятельно. В большинстве случаев необходимо установить необходимые зависимые пакеты. Это можно сделать с помощью YUM или вручную (в этом случае необходимо выяснить, какие зависимые пакеты необходимо установить, загрузить и инсталлировать их прежде чем устанавливать Samba; можно сделать это одновременно с инсталляцией Samba, указав несколько имен файлов в командной строке `rpm`).

## Обновление до новой версии с помощью RPM

Обновлять приложения с помощью RPM проще простого. Если вы используете YUM, процесс обновления в точности повторяет процесс установки; однако вместо подкоманды `install` можно по желанию указать подкоманду `update`. При непосредственном использовании утилиты `rpm` вместо опции `--install (-i)` следует использовать опцию `--upgrade (-U)`. Фактически, опцию `--upgrade / -U` можно использовать для инсталляции новых пакетов вместо опции `--install / -i` – именно так и поступают некоторые администраторы.

При обновлении приложения с помощью RPM его старая версия удаляется и инсталлируется новая; при этом автоматически удаляются все устаревшие файлы. Возможно, вы захотите проверить конфигурационные файлы, например `/etc/samba/smb.conf`. Обычно существующие конфигурационные файлы остаются неизменными, а обновленная конфигурация помещается в файл с похожим названием, например `/etc/samba/smb.rpmnew`; это позволяет сравнить старую и новую конфигурации и при необходимости внести изменения. В качестве меры предосторожности перед обновлением приложения вы можете создать резервную копию текущей конфигурации.

## Установка из Debian-пакета

Концептуально Debian-пакеты похожи на RPM-пакеты, однако для управления ими используются другие утилиты. Два основных, хотя и не единственных дистрибутива, использующих Debian-пакеты – это Debian и Ubuntu.

## Инсталляция с помощью APT

Так же, как и система YUM, используемая во многих дистрибутивах на основе RPM, система Advanced Package Tools (APT) предоставляет средства сетевого управления пакетами, включая определение зависимых пакетов. Система управления пакетами APT также доступна во многих дистрибутивах на основе RPM, например в PCLinuxOS, который использует APT по умолчанию.

Перед инсталляцией Samba рекомендуем вам загрузить самый последний список пакетов APT. Это можно сделать с помощью команды `apt-get` и ее подкоманды `update`:

```
# apt-get update
```

После выполнения этой команды APT проверит свои настроенные репозитории и загрузит самый последний список доступных пакетов, таким образом, вы сможете инсталлировать самую последнюю версию Samba, доступную для вашей операционной системы. Для установки пакета с помощью командной строки APT используйте команду `apt-get` и ее подкоманду `install`:

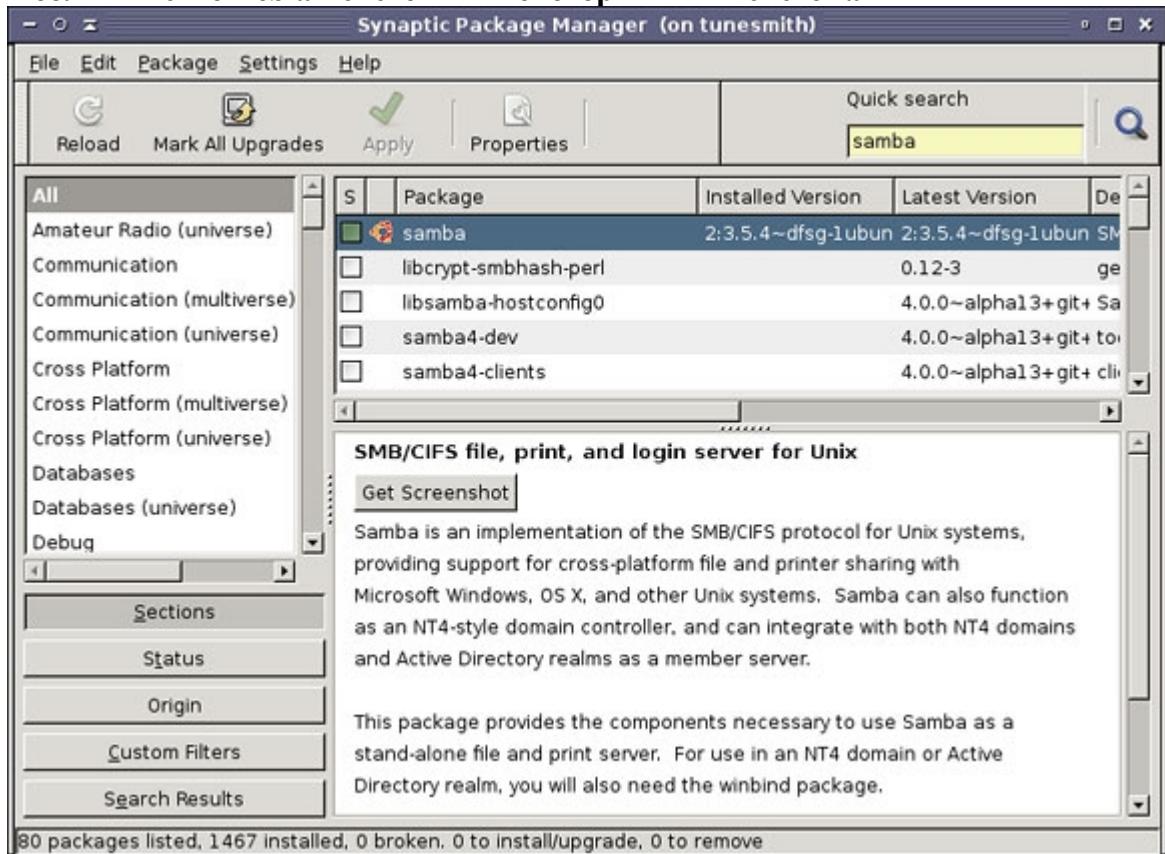
```
# apt-get install samba
```

В результате вы увидите список пакетов, которые будут инсталлированы, удалены и обновлены; также вам будет предложен список опционных пакетов. Если вы согласны с

предложенными действиями, нажмите Y в строке приглашения. После этого все необходимые пакеты будут загружены и инсталлированы с помощью низкоуровневых инструментов Debian. .

Если вы предпочтете использовать графические инструменты, то можете воспользоваться утилитой Synaptic (вызывается из командной строки по команде `synaptic`), показанной на рисунке 2. Так же, как и Yumex, Synaptic особенно полезна в тех случаях, когда вы не знаете точное имя инсталлированного пакета или когда вам необходимо найти все вспомогательные пакеты.

**Рисунок 2. Графический интерфейс системы управления пакетами Synaptic, доступной в большинстве Debian-систем и в некоторых RPM-системах**



## Установка с помощью dpkg

Если вы не можете или не хотите инсталлировать Samba с помощью APT, то это можно сделать с помощью низкоуровневой утилиты `dpkg`, которая управляет файлами Debian-пакетов (файлы с расширением `.deb`), которые можно загрузить из Интернета или передать на компьютер любым другим способом. Мы рекомендуем всегда по возможности проверять подлинность пакетов с помощью `gpg`, как это было описано в предыдущей статье этой серии. Для инсталляции нового пакета можно использовать опцию `--install (-i)`:

```
# dpkg -i samba_2:3.5.4~dfsg-1ubuntu8.1_i386.deb
```

Если все зависимые пакеты уже инсталлированы, то эта команда инсталлирует основной пакет Samba. Если зависимые пакеты не инсталлированы или инсталлированы не полностью, то `dpkg` сообщит об этом. В этом случае нужно инсталлировать все необходимые пакеты либо с помощью APT, либо вручную с помощью `dpkg` (команда `dpkg` позволяет одновременно инсталлировать сразу несколько пакетов).

## **Обновление до новой версии с использованием Debian-пакетов**

Можно обновить Samba с помощью `apt-get` или `dpkg`, выполняя те же действия, что и во время первоначальной инсталляции. В отличие от RPM-пакетов для обновления приложений на основе Debian не предусмотрена отдельная опция. Так же, как и при работе с RPM, по окончании процедуры обновления следует проверить старые конфигурационные файлы и убедиться, что они не были изменены, а также просмотреть файл с новой конфигурацией, если вы намерены использовать новые параметры, которые она может содержать.

Если вы используете APT, то не забудьте обновить базу данных доступных приложений, выполнив команду `apt-get update` перед использованием подкоманды `install`. Также можно обновить *все* программное обеспечение вашего компьютера, выполнив команду `apt-get upgrade` или `apt-get dist-upgrade` (последняя команда выполняет более глубокий анализ зависимых пакетов, вследствие чего некоторые устаревшие пакеты могут быть удалены).

## **Запуск Samba**

Если вы устанавливаете Samba из двоичного пакета, разработанного специально для вашего дистрибутива, то в него включен сценарий запуска SysV или Upstart, которые запускают Samba при загрузке компьютера. Однако при первой установке пакета Samba этот сценарий может и не быть активирован. Чтобы определить, на каких уровнях выполнения будет запускаться Samba, используйте инструменты для управления локальной загрузкой компьютера, например, `chkconfig` (используется в Fedora и дистрибутивах на ее основе) или `rc-update` (используется в операционных системах на основе Debian), или вручную проверьте ссылки на сценарии запуска SysV или конфигурационные файлы Upstart.

**Примечание.** Хотя возможно запускать Samba посредством супер-сервера (например, `inetd` или `xinetd`), такие конфигурации встречаются редко и создают проблемы, связанные с производительностью.

Если вы установили Samba из исходного кода, то придется создать собственный сценарий запуска SysV или Upstart, или добавить строку для запуска сервера в локальный сценарий, например, в файл `/etc/rc.d/rc.local` или `/etc/init.d/rc.local`. Обычно запускают два сервера – `smbd` и `nmbd` с параметром `-D`, который указывает, что они должны выполняться в режиме демонов. Минимальная конфигурация запуска выглядит следующим образом:

```
/usr/local/sbin/nmbd -D  
/usr/local/sbin/smbd -D
```

Конечно же, вы должны указать путь к двоичным файлам вашей системы. Таким же образом можно запускать различные вспомогательные службы, например, SWAT.

## **Что дальше**

Эта статья завершает тему 311 "Компиляция и установка Samba". Следующая статья открывает новую тему и содержит материалы цели 312.1 темы 312 ("Настройка и использование Samba"). В ней рассматривается базовая настройка Samba, в том числе структура конфигурационного файла Samba, настройка базовых параметров Samba и устранение наиболее распространенных неисправностей.

## **Ресурсы**

### **Научиться**

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Install and upgrade Samba \(EN\)](#).

- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Чтобы получить ссылки на все статьи этой серии, которые помогут вам подготовиться к успешной сдаче экзамена LPI-302, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#).
- В [разделе Linux сайта developerWorks](#) можно найти сотни [пошаговых инструкций и руководств](#), загрузить программные продукты, а также получить ссылки на форумы и многие другие ресурсы, ориентированные на разработчиков и администраторов Linux.
- Следите на последними новостями на портале [Web-трансляций и технических мероприятий developerWorks](#) (EN), относящимися к продуктам IBM и событиям ИТ-индустрии.
- Посещайте [бесплатные брифинги developerWorks Live!](#) (EN), чтобы быть в курсе последних событий, относящихся к продуктам IBM и направлениям в ИТ-индустрии.
- Смотрите [демонстрационные материалы по запросу на сайте developerWorks](#) (EN), ориентированные как на новичков, так и на опытных разработчиков.
- В статье [Learn Linux, 302 \(Mixed environments\): Configure and build Samba from source](#) (EN) (developerWorks, апрель 2011 г.) рассказывается, как скомпилировать Samba из исходного кода. Материал этой статьи необходимо знать для установки приложений из исходного кода; для установки Samba из двоичных пакетов этого знать не требуется.

## Получить продукты и технологии

- Вы можете загрузить систему Samba и получить информацию о ней на [Web-сайте Samba](#) (EN).

# Изучаем Linux, 302 (смешанные среды): Конфигурация Samba

*Настройка Samba для выполнения различных задач*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Конфигурационные параметры Samba хранятся в обычном текстовом файле, поэтому самым сложным инструментом настройки Samba является текстовый редактор. Из этой статьи вы узнаете о структуре конфигурационного файла Samba и взаимодействии Samba с сетевым окружением, о настройке журналирования, а также о том, как использовать систему отладки для устранения неисправностей в работе Samba.

[Больше статей из этой серии](#)

**Дата:** 24.05.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели экзаменов сертификации LPIC-3*.

В этой статье рассматриваются следующие темы:

- Структура конфигурационного файла сервера Samba.
- Использование конфигурационных параметров и переменных Samba.
- Идентификация ключевых портов TCP/UDP, используемых в Server Message Block (SMB)/Common Internet File System (CIFS).
- Настройка журналирования Samba.
- Отладка и устранение неисправностей в работе Samba.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 312.1 темы 312. Цель имеет вес 6.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. Кроме этого, у вас должен быть доступ к среде Windows, которую можно использовать для проверки доступа к файлам и принтерам.

## Конфигурационный файл Samba

Как и большинство других демонов UNIX, Samba настраивается с помощью простых текстовых файлов, которые можно просматривать в обычном текстовом редакторе (в отличие от двоичных файлов, работа с которыми требует использования графических средств). Самый важный конфигурационный файл называется *smb.conf* и содержит все параметры, необходимые Samba для работы в конкретном рабочем окружении.

**Примечание.** Не смотря на то, что с файлом smb.conf можно работать в текстовом редакторе, команда разработчиков Samba создала Web-инструмент под названием *Samba Web Administration Tool*. Также существуют и другие альтернативные инструменты, например, webmin. Важно помнить о том, что файл smb.conf можно редактировать как до, так и после использования этих инструментов, поскольку вся работа происходит с обычным текстовым файлом.

Конфигурационный файл Samba имеет достаточно простой формат и может содержать три различные конструкции:

- **Разделы** – группируют параметры по различным независимым областям. Например, все параметры общих файловых ресурсов содержатся в отдельном разделе.
- **Параметры** – представляют собой пары вида "ключ-значение". Ключи – это хорошо знакомые всем атрибуты, например, "read only" ("только для чтения").
- **Комментарии** – позволяют оставлять заметки в конфигурационном файле, не влияющие на конфигурацию, например, описания файловых ресурсов.

## Разделы

Разделы группируют параметры по различным независимым областям. Каждый раздел начинается с названия, заключенного в квадратные скобки ([ ]). Далее раздел продолжается до тех пор, пока не начнется новый раздел, или до конца файла.

Существуют три специальных раздела со следующими названиями:

- **global**. Все параметры, содержащиеся в этом разделе, влияют на работу всего сервера. При необходимости параметры раздела **global** можно переопределить на уровне общего ресурса.
- **homes**. Этот раздел является шаблоном для домашних директорий пользователей; Samba самостоятельно сопоставляет имена пользователей в соответствии с настройками этого раздела, избавляя от необходимости настраивать отдельный общий ресурс каждый раз, когда необходимо предоставить пользователю доступ к его домашней директории.
- **printers**. Этот раздел аналогичен разделу **homes**, но используется для принтеров.

Если раздел называется по-другому, то считается, что он относится к общему файловому ресурсу или принтеру.

Когда Samba получает запрос на подключение к общему ресурсу, то она ищет раздел с именем этого ресурса, определяющего его свойства. Если такой раздел не найден, Samba просматривает список локальных пользователей компьютера, чтобы выяснить, не указывает ли запрос на пользователя. Если пользователь с таким именем не найден, то Samba ищет принтер с таким именем в списке локальных принтеров. Если Samba находит имя пользователя, то используется раздел конфигурации **homes**. Если же Samba находит принтер с таким именем, то используется раздел **printers**. В любом случае параметры раздела переопределяют глобальные параметры раздела **global**.

Если все перечисленные проверки не увенчались успехом, то выполняется последняя проверка. Если настроена служба по умолчанию, то используется она. В противном случае клиенту возвращается сообщение об ошибке. По умолчанию служба по умолчанию не настроена, поэтому неверное указание имени общего ресурса приведет к ошибке.

## Параметры

Параметры имеют форму **ключ = значение**, т. е. каждому *ключу* присваивается *значение*. Все ключи описаны на man-странице smb.conf. Искусство конфигурирования Samba заключается, по большей части, в том, чтобы понять, какие ключи необходимо использовать для получения требуемого результата и какие значения им присваивать.

В основном значениями параметров являются строки. Samba поддерживает использование макросов, позволяющих динамически изменять значение параметра в соответствии с именем общего ресурса или введенными пользователем данными. Например, по умолчанию все параметры раздела `homes` указывают на домашние директории пользователей UNIX, однако с помощью макросов можно задать любое местоположение, и в момент подключения вместо имени пользователя указывать другой путь к домашней директории. Макросы начинаются с символа `%` и будут рассмотрены позже в этой статье.

Если значение параметра не помещается в одной строке, то все содержащие его строки за исключением последней должны заканчиваться обратным слешем (`\`), так же, как и в командной оболочке UNIX.

## Комментарии

Комментарии начинаются с символа точки с запятой (`;`) или символа решетки (`#`).

Комментарии можно использовать для пояснения того, почему был использован тот или иной параметр, отслеживания изменений в конфигурации или обозначения границ разделов.

## Пример конфигурации

В листинге 1 показаны различные фрагменты конфигурационного файла `smb.conf`.

### Листинг 1. Пример конфигурационного файла Samba

```
# Это комментарий
; И это тоже
# Не забудьте внести все общие ресурсы в Wiki! -Opsteam
[global]
    workgroup = BIGCO
    # %v замещается версией Samba
    server string = Samba Server Version %v
    # По умолчанию все файлы, начинающиеся
    с точки будут иметь атрибут "скрытый"
    hide dot files = yes

# Домашние директории берутся из файла паролей UNIX
# этот раздел будет использоваться для всех пользователей
[homes]
    comment = Home directories
    # файлы, начинающиеся с точки, будут скрыты,
    т. к. это определено в разделе глобальных ключей

[printers]
    comment = System printers
    printable = yes

# Общедоступный ресурс
[projecta]
    path = /var/spool/projects/projecta
    # Переопределим глобальное значение
    для файлов, начинающихся с точки
    hide dot files = no
```

Эта конфигурация имеет следующие особенности:

- В этой конфигурации используются два различных способа вставки комментариев. В первом случае комментарии начинаются с символа решетки, во втором случае – с символа точки с запятой.

- В этой конфигурации определен один общий ресурс с именем *projecta*. Все остальные общие ресурсы будут созданы автоматически на основе пользователей и принтеров, определенных в системе
- Частью параметра **server string** является макрос %V. При запуске вместо конструкции %V будет отображаться номер версии Samba.
- В разделе глобальных переменных параметр **hide dot files** установлен в yes, но в разделе общего ресурса *projecta* он установлен в no. Для конфигурации домашних директорий используется раздел **homes**, при этом все файлы в этих директориях, начинающиеся с точки (например, .profile) будут скрыты. Файлы проекта *projecta*, начинающиеся с точки, будут отображаться.

## Работа Samba в сетевом окружении

Samba – это сетевая служба, работающая по протоколу IP, который позволяет ей взаимодействовать с другими узлами сети, также использующими этот протокол. Чтобы администратор Samba мог решать проблемы, связанные с сетевым взаимодействием, он должен понимать, как различные службы Samba работают в сетевом окружении.

Не вдаваясь в подробности, можно сказать, что Samba предоставляет сетевые службы трех различных типов:

- **Службы общего доступа к файлам и принтерам** – обеспечивают сетевой доступ к файлам и принтерам для устройств, на которых запущены эти службы.
- **Службы имен** – службы разрешения имен, необходимые для работы в сетях Microsoft.
- **Доменные службы** – позволяют Samba взять на себя различные роли сервера Microsoft, например, legacy domain controller, и обеспечивают интеграцию с новыми серверами Active Directory Domain Services (AD DS).

## Общий доступ к файлам и принтерам

Общий доступ к файлам и принтерам реализован посредством **smbd** – одного из демонов Samba. Когда Microsoft впервые реализовала поддержку IP-сетей, для предоставления общего доступа к файлам она использовала механизм NetBIOS over TCP. Этот метод заключался в инкапсуляции трафика NetBIOS в сеанс TCP-протокола с использованием TCP-порта 139.

Протокол NetBIOS выполняет несколько функций. TCP-порт 139 используется только для служб сеансов, которые выполняют передачу файлов и сообщений. Службы имен не работают на этом порту.

Хотя механизм NetBIOS over TCP работает, существует частичное наложение между службами сеансов и надежностью, предоставляемыми с одной стороны NetBIOS, а с другой стороны – протоколом TCP. Путем некоторых изменений стало возможно обеспечить работу SMB/CIFS непосредственно поверх TCP. Этот метод известен как *прямая передача* и используется для упрощения протокола. Прямая передача использует для работы TCP-порт 445.

Когда протокол NetBIOS был удален из стека протоколов, перед Microsoft встало задание найти другой способ разрешения имен. Естественным выбором стала система доменных имен DNS (Domain Name System), и именно поэтому DNS лежит в основе доменных служб AD DS.

По умолчанию Samba прослушивает порты 139 и 445. Можно использовать другие порты, изменив глобальный параметр **smb ports**. Например, **smb ports = 445** указывает Samba прослушивать только порт 445. Вы можете настроить Samba на прослушивание любого порта, но в этом случае клиентам, которые хотят подключиться к серверу, необходимо сообщить о том, что они должны использовать нестандартный порт.

Если вы не знаете точно, какой порт прослушивает Samba, то это можно выяснить с помощью команды **netstat**. В листинге 2 приведен практический пример ее

использования.

## Листинг 2. Использование команды netstat для поиска портов, прослушиваемого SBM

```
# netstat -antp | grep smbd
# netstat -antp | grep smb
tcp    0  0  :::445           :::*               LISTEN      2830/smbd
tcp    0  0  ::ffff:192.168.1.143:445  ::ffff:192.168.1.147:4724  ESTABLISHED 2877/smbd
```

В листинге 2 продемонстрирован запуск команды **netstat**, вывод которой был отфильтрован командой **grep** для поиска строки **smb**. Использовались следующие параметры **netstat**: показать все (-a) соединения TCP (-t) в числовом (-n) формате вместе с именами процессов (-p). В результате мы получили две строки. Первая строка содержит слово **LISTEN**, которое означает, что демон прослушивает порт в ожидании входящих подключений. В нашем случае прослушивается порт 445. Вторая строка содержит слово **ESTABLISHED**, которое означает, что с адреса 192.168.1.147 было установлено соединение с портом 445 локального хоста (192.168.1.143). Таким образом, листинг 2 позволяет сделать вывод о том, что **Smbd** прослушивает только порт 445 и что в данный момент подключен один клиент.

## Службы имен

В NetBIOS включены службы имен, отвечающие за просмотр сетевого окружения и разрешение имен. Например, путем отправки запроса службе имен, работающей на UDP-порту 137, узлу SERVER1 будет сопоставлен его IP-адрес. Для просмотра и выбора вспомогательных ролей, таких как главный обозреватель, используется UDP-порт 138, известный также как *порт службы датаграмм*. Службы имен реализованы посредством демона **nmbd**.

Важно отметить, что вместо TCP-порта службы имен используют UDP-порт, поскольку UDP-пакеты не устанавливают соединений и могут передаваться в широковещательных запросах сразу всем узлам сети, а не каждомуциальному узлу в одноадресной передаче. Поддержка широковещательных запросов протоколом UDP позволяет упростить работу служб имен NetBIOS.

Третья версия Samba не содержит каких-либо параметров для управления номерами портов, прослушиваемых демоном **nmbd**, однако в четвертой версии Samba появились глобальные параметры **nbt port** и **dgram port**, которые управляют портами службы имен и службы датаграмм, соответственно.

Чтобы просмотреть список портов, открытых демоном **nmbd**, можно использовать команду, подобную команде из листинга 2:

## Листинг 3. Просмотр списка портов, прослушиваемых nmbd

```
# netstat -anup | grep nmbd
udp    0      0 192.168.1.255:137      0.0.0.0:*
udp    0      0 192.168.1.143:137      0.0.0.0:*
udp    0      0 0.0.0.0:137          0.0.0.0:*
udp    0      0 192.168.1.255:138      0.0.0.0:*
udp    0      0 192.168.1.143:138      0.0.0.0:*
udp    0      0 0.0.0.0:138          0.0.0.0:*
```

Помимо того, что вместо демона `smbd` был указан демон `nmbd`, команда из листинга 3 ищет не TCP-порты, а UDP-порты, для чего используется опция `-U` команды `netstat`.

Результатом выполнения команды является список, показывающий, что `nmbd` прослушивает порты 137 и 138 на различных сетевых интерфейсах, а также на широковещательном адресе 192.168.1.255. Оба порта службы имен используют для работы как прямые взаимодействия между узлами сети, так и широковещательные взаимодействия.

## Доменные службы

Команда разработчиков Samba постоянно усовершенствует свой программный продукт, все более тесно интегрируя его с сетями Microsoft и реализуя функции, которые позволили бы Samba полностью заменить инфраструктуру Microsoft. Для этого Samba должна уметь имитировать соответствующие службы сетевой инфраструктуры.

В большинстве этих служб так или иначе задействованы протокол безопасности Kerberos и протокол LDAP (Lightweight Directory Access Protocol). Эти протоколы будут более подробно рассмотрены в следующих статьях, а сейчас достаточно просто знать о том, что Samba может не просто предоставлять общий доступ к файлам, а делать нечто большее.

Итоговый список портов, используемых Samba

В таблице 1 перечислены все порты, которые использует Samba для предоставления общего доступа к файлам.

**Таблица 1. Итоговый список портов, используемых Samba**

Порт	Протокол	Служба	Демон	Описание
137	UDP	netbios-ns	nmbd	Службы имен NetBIOS
138	UDP	netbios-dgm	nmbd	Службы датаграмм NetBIOS
139	TCP	netbios-ssn	smbd	NetBIOS over TCP (службы сеансов)
445	TCP	microsoft-ds	smbd	NetBIOS over TCP (службы сеансов)

Названия в столбце **Служба** – это общеизвестные имена служб, перечисленные в файле `/etc/services`. Файл `services` помогает людям и приложениям сопоставлять имена служб и номера используемых ими портов. Несмотря на то, что многие службы резервируют как TCP-, так и UDP-порты, приложениям не обязательно использовать оба этих типа портов. Такое резервирование устраниет возможные конфликты в тех ситуациях, когда две различные службы пытаются использовать один и тот же номер порта для различных протоколов.

Нелишне также отметить, что вовсе не обязательно использовать именно те номера портов и имена служб, которые присутствуют в файле `/etc/services`. Если вы можете сообщить об этом клиентам, то можно запускать демонов на различных портах, отличных от портов по умолчанию. Например, можно запускать Samba на портах 5137-5139 и 5445, если использующие ее клиенты знают об этом и не будут использовать стандартные порты.

## Устранение неисправностей в работе Samba

У Samba есть свои проблемы. Иногда эти проблемы возникают в результате вмешательства системного администратора, а иногда – в результате действий пользователей. Задача системного администратора – выявить первопричину проблемы и найти ее решение.

## Проверка конфигурационного файла

Если Samba не запускается или вы просто хотите проверить правильность конфигурационного файла, то вам поможет утилита `testparm`. Эта утилита проверяет корректность файла `smb.conf`. В листинге 4 показан результат проверки файла, содержащего ошибку, с помощью `testparm`.

#### Листинг 4. Проверка файла smb.conf, содержащего ошибку, с помощью testparm

```
# testparm
Load smb config files from /etc/samba/smb.conf
Unknown parameter encountered: "hide dto files"
Ignoring unknown parameter "hide dto files"
Processing section "[homes]"
Processing section "[printers]"
Processing section "[public]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
    workgroup = MYGROUP
    server string = Samba Server Version %v
    passdb backend = tdbsam
    log file = /var/log/samba/log.%m
    max log size = 50
    cups options = raw

<< rest of the output omitted >>
```

Вывод **testparm** начинается с указания местоположения файлов. Если вы хотите указать другой файл, то передайте его имя в качестве параметра командной строки, как показано в следующем примере:

```
testparm /home/me/smb.conf
```

Далее **testparm** сообщает о неверном параметре с именем **hide dto files**. На самом деле этот параметр должен называться **hide dot files**.

После обработки конфигурационного файла выводится информация о роли сервера и краткая версия конфигурационного файла. В этой версии отсутствуют комментарии, и она приведена к единому формату; иногда это позволяет обнаружить ошибку, которую вы пропустили, просматривая файл smb.conf в текстовом редакторе.

Следует запускать **testparm** после каждого изменения конфигурационного файла. Samba игнорирует большинство опечаток в конфигурационных файлах и при запуске не всегда выводит сообщения об ошибках на консоль, поэтому есть риск не заметить ошибку до тех пор, пока что-то не станет работать неправильно. Утилита **testparm** предупреждает о любых опечатках в файле smb.conf.

По умолчанию **testparm** выводит только те параметры конфигурации, которые определены в файле smb.conf. Если вы подозреваете, что в каком-то месте используются значения по умолчанию, то опция **-V** указывает **testparm** выводить также значения по умолчанию.

С помощью **testparm** можно также выводить только определенный раздел или значение определенного параметра. В листинге 5 показано, как с помощью **testparm** посмотреть значение параметра **security mask**.

#### Листинг 5. Вывод значения отдельного параметра с помощью testparm

```
# testparm -s --parameter-name "security mask"
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
```

```
Processing section "[printers]"
Processing section "[public]"
Loaded services file OK.
0777
```

В листинге 5 параметр `-S` указывает `testparm` не ожидать пользовательского ввода в промежутке между анализом файла `smb.conf` и выводом его содержимого на экран. Конструкция `--parameter name "security mask"` запрашивает значение параметра `security mask`. Результатом является значение `0777`, которое является значением по умолчанию. В этом режиме необязательно указывать опцию `-V` для вывода значений по умолчанию.

### Подключение в качестве клиента

Вместо того, чтобы запускать графическую оболочку пользователя и самостоятельно пробовать выполнять все действия, можно проверить работу Samba с помощью командной строки вашего компьютера. Первая и самая простая проверка заключается в том, чтобы суметь подключиться к порту Samba. Проще всего сделать это с помощью утилиты `telnet`, как показано в листинге 6.

### Листинг 6. Проверка подключения с помощью утилиты telnet

```
# telnet bob 139
Trying 192.168.1.134...
telnet: connect to address 192.168.1.134: Connection refused
```

В листинге 6 пользователь `root` подключается к порту 139 сервера `bob`. Для проверки прямого подключения SMB можно использовать порт 445. Результатом является сообщение `Connection refused`, которое говорит о том, что либо демон не прослушивает указанный порт по этому адресу, либо подключение блокируется брандмауэром. То же самое могут означать и другие сообщения, например, `No route to host` или `Connection timed out`.

Обычно клиенты подключаются к серверу по его имени, а не по IP-адресу. Если с помощью `telnet` вы подключаетесь к серверу по имени, а не по IP-адресу, то обратите особое внимание на возвращаемый IP-адрес. В вышеупомянутом примере серверу `bob` был сопоставлен IP-адрес 192.168.1.134. Иногда в DNS могут содержаться ошибочные записи, в результате чего клиенты подключаются по неверному адресу.

Если вы не используете DNS для разрешения Windows-имен, то можно использовать команду `nmblookup` для поиска имен NetBIOS. В листинге 7 показан запрос для сервера `bob`.

### Листинг 7. Выполнение запроса NetBIOS-имени для сервера bob

```
# nmblookup bob
querying bob on 192.168.1.255
192.168.1.138 bob<00>
```

Согласно листингу 7, сервер `bob` имеет IP-адрес 192.168.1.138, а не 192.168.1.134, как было определено в листинге 6. Этот результат указывает на проблему с DNS, особенно если порты 139 и 445 отзываются по адресу 192.168.1.138.

Другая проверка заключается в том, чтобы выяснить, не запрещен ли доступ к определенному хосту в конфигурационном файле. Для этого мы снова используем утилиту `testparm`, как показано в листинге 8.

### Листинг 8. Проверка доступа с помощью `testparm`

```
# testparm /etc/samba/smb.conf seanspc 192.168.1.147
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[public]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Deny connection from seanspc (192.168.1.147) to homes
Deny connection from seanspc (192.168.1.147) to printers
Deny connection from seanspc (192.168.1.147) to public
```

### Брандмауэр или приложение?

Существует множество способов заблокировать подключение к компьютеру, однако все их можно разделить на две категории: сеть и приложение. Заблокировав доступ на сетевом уровне с помощью корпоративного или персонального (например, `iptables`) брандмауэра, вы увидите, что подключение `telnet` из [листинга 6](#) отклоняется или разрывается по таймауту. Это происходит потому, что пакеты никогда не доходят до приложения Samba.

Если Samba настроена таким образом, чтобы не разрешать подключаться определенным компьютерам, вы увидите, что подключение `telnet` успешно устанавливается, но любые попытки получить доступ с клиентского компьютера завершаются ошибкой. В этом случае пакеты доходят до приложения, но содержат подозрительный IP-адрес или имя хоста, поэтому Samba посыпает в ответ сообщение об ошибке на уровне приложения. Не получив пакет на уровне приложения, Samba не сможет определить, пришел ли этот пакет с IP-адреса, который включен в число разрешенных адресов.

В листинге 8 программе `testparm` передаются три аргумента:

- Путь к конфигурационному файлу Samba.
- NetBIOS-имя проверяемого компьютера.
- IP-адрес проверяемого компьютера.

Из листинга 8 видно, что для указанного компьютера доступ ко всем общим ресурсам запрещен. В этом режиме `testparm` не выполняет фактического подключения к компьютеру, а вместо этого обращается к конфигурационному файлу и проверяет, разрешен ли этому компьютеру доступ или нет.

Если к этому моменту все проверки были выполнены успешно, можно попытаться установить клиентское подключение с помощью утилиты `smbclient`. Прежде всего, нужно попытаться просмотреть список общих ресурсов, как показано в листинге 9.

### Листинг 9. Просмотр общих ресурсов компьютера

```
[sean@bob source3]$ smbclient -L '\\bob'
Enter sean's password:
Anonymous login successful
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.6-69.fc13]
```

```

Sharename      Type      Comment
-----        ----
extdrive      Disk
Sean Walberg's iMac Disk
timemachine   Disk
IPC$          IPC       IPC Service (Samba Server Version 3.5.6-69.fc13)
test          Printer   test
Downstairs_Laser Printer  HP 6L
Cups-PDF      Printer   Cups-PDF
Anonymous login successful
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.6-69.fc13]

```

Server	Comment
BOB	Samba Server Version 3.5.6-69.fc13
Workgroup	Master
-----	-----
MYGROUP	BOB
WORK	SWALBERG-XPLT
WORKGROUP	IMAC-1FC525

В листинге 9 с помощью параметра **-L** пользователь запрашивает список общих ресурсов сервера с именем *bob*. Поскольку имя сервера – это UNC-путь, то перед ним ставятся две обратных косых черты (\\\). Будьте внимательны и не перепутайте одиночные кавычки с двойными. Две обратные косые черты, заключенные в одиночные кавычки, интерпретируются как символы escape-последовательности.

Если на сервере настроен более высокий уровень безопасности, то может потребоваться передать имя пользователя или домена с помощью параметров **-W** и **-U** соответственно.

Наконец, можно попытаться подключиться к общему ресурсу, не используя параметр **-L**, а указав полный UNC-путь к общему ресурсу. В листинге 10 показано клиентское подключение к серверу с использованием другой рабочей группы и имени пользователя.

#### **Листинг 10. Подключение к общему ресурсу с использованием другого имени пользователя и домена**

```

[sean@bob source3]$ smbclient '\\swalberg-xplt\photos' -U swalberg -W WORK
Enter swalberg's password:
Domain=[WORK] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \> dir
.
D          0 Thu Jan  6 11:39:50 2011
..
D          0 Thu Jan  6 11:39:50 2011
<< files omitted >>
            38156 blocks of size 4194304. 2938 blocks available
smb: \>>

```

Если все эти проверки успешно выполнены, то можно быть точно уверенными в том, что проблема заключается не в конфигурации Samba, а возникает где-то между клиентом и сервером или же на самом клиенте. В следующем разделе мы проанализируем файлы журналов, которые могут подсказать, где именно следует искать проблему.

## Журнилирование и отладка

Журнилирование и устранение неисправностей идут рука об руку. Журналы позволяют видеть, не возникали ли какие-либо ошибки в работе Samba, и получать дополнительную информацию о них. Если вы пытаетесь выяснить, почему что-то не работает, то можно настроить более подробный уровень журналирования, позволяющий определить источник проблемы. В файле `smb.conf` имеется ряд полезных параметров для управления журналированием, которые можно использовать для создания журналов определенных типов.

Работа Samba не отличается от работы обычного демона UNIX в том плане, что она может записывать события для определенной категории `syslog`, а также создавать собственные файлы журналов. Более того, можно извлекать журналы с сервера Samba с помощью инструмента Microsoft Event Viewer (MMC-оснастка "Просмотр событий"). В последнем случае существует небольшая проблема, которая заключается в том, что Samba не может записывать события в формате журнала регистрации событий Windows, поэтому журналы Samba нуждаются в предварительной обработке.

### Уровни журналирования

Каждое генерируемое Samba сообщение имеет *уровень* от 0 до 10. Наиболее важные сообщения (например, сообщения о новых подключениях и важные сообщения об ошибках) имеют более низкие уровни. Отладочные сообщения имеют более высокие уровни. Т. е. чем выше уровень журналирования, тем подробнее будет журнал. Можно управлять количеством журналируемых сообщений, задавая их максимальный уровень. На уровне журналирования 1 в журнал попадают только сообщения с приоритетами 0 и 1. Если вам требуется более подробное журналирование, указывайте более высокие уровни.

Все сообщения уровня 4 и выше предназначены для разработчиков и не особо полезны для системных администраторов. Уровень журналирования 0 отключает запись всех сообщений за исключением некоторых сообщений запуска и сообщений о критических ошибках.

Для настройки уровня журналирования используйте параметр `log level` в разделе `global`; например, параметр `log level = 2` устанавливает уровень журналирования 2, на котором будут записываться все события с приоритетом равным или меньшим 2.

### Примечание к экзамену

В рамках экзамена LPIC упоминается, что очень важно знать о параметре `debuglevel`. `debuglevel` – это синоним *уровня журналирования*, и оба этих параметра взаимозаменямы.

Можно изменить уровень журналирования во время выполнения, послав процессу Samba сигнал `SIGUSR1` для повышения уровня или сигнал `SIGUSR2` для его понижения.

Можно еще больше конкретизировать уровни журналирования, увеличивая подробность сообщений, касающихся только определенных функций. Для этого нужно указать, события какого класса необходимо записывать в журнал. Имеются следующие классы:

- **all.** Необязательный параметр, используется, если не указано никакое другое ключевое слово.
- **tdb.** Запись событий, относящихся к базам данных TDB – хранилищам типа "ключ-значение", которые использует Samba.
- **printdrivers.** Функции управления драйверами принтеров.
- **lanman.** Отладка протокола NTLM (NT LAN Manager).
- **smb.** Отладка протокола SMB.
- **rpc\_parse.** Анализ вызовов удаленных процедур (RPCs).

- **rpc\_srv.** Вызовы удаленных процедур на стороне сервера.
- **rpc\_cli.** Вызовы удаленных процедур на стороне клиента.
- **passdb.** Старый способ хранения паролей на сервере Samba.
- **sam.** Локальная база данных паролей Samba.
- **auth.** Различные внутренние модули авторизации пользователей Samba.
- **winbind.** Компонент, предназначенный для прозрачного подключения пользователей Microsoft к UNIX-системам.
- **vfs.** Сообщения отладки модулей виртуальной файловой системы Virtual File System, позволяющих расширять функциональность Samba с помощью подключаемых модулей.
- **idmap.** Сопоставление сущностей, имеющих идентификаторы пользователей UNIX и идентификаторы безопасности Microsoft.
- **quota.** Сообщения, относящиеся к обработке квот как политиками Microsoft Windows NT, так и политиками файловой системы UNIX.
- **acls.** Обработка списков контроля доступа.
- **locking.** Статус ошибки блокировки файлов.
- **msdfs.** Сообщения, относящиеся к поддержке Samba распределенной файловой системы.
- **dmapi.** Функциональность API-интерфейса управления данными. Для использования этой функциональности Samba должна быть скомпилирована с поддержкой сторонней реализации DMAPI.
- **registry.** Эмуляция системного реестра Windows.

Для использования этого дополнительного журналирования добавьте ключевое слово и значение к параметру уровня журналирования, разделенные двоеточием (:). Например, `log level = 1 winbind:3` устанавливает системный уровень журналирования по умолчанию 1 и увеличивает уровень журналирования событий `winbind` до 3. Это позволяет отслеживать проблемы, не путаясь в большом количестве ненужных log-файлов.

#### Расположение log-файлов

Чтобы изменить имя log-файла, используйте параметр `log file`. Значение этого параметра может содержать макросы. Часто используется подход, в котором для каждого клиента создается отдельный log-файл. Это можно сделать следующим способом:

```
log file = /var/log/samba/log.%m
```

Эта команда создает отдельный log-файл для каждого клиента, а остальные сообщения продолжают сохраняться в файле `log.smbd`.

Если необходимо отправлять события в `syslog`, то можно задать параметр `syslog = 1` для отправки всех сообщений уровня 1 или 0 на локальный syslog-сервер. Samba использует категорию (facility) `LOG_DAEMON` и сопоставляет уровни журналирования Samba с приоритетами `syslog` следующим образом:

- **LOG\_ERR.** Уровень журналирования 0.
- **LOG\_WARNING.** Уровень журналирования 1.
- **LOG\_NOTICE.** Уровень журналирования 2.
- **LOG\_INFO.** Уровень журналирования 3.
- **LOG\_DEBUG.** Уровни журналирования от 4 и выше.

Если вы используете более продвинутый демон `syslog`, который может отфильтровывать входящие сообщения и посыпать уведомления системному администратору, то это

прекрасный способ отслеживать работоспособность сервера Samba.

## Метаданные журналов

Можно добавлять или удалять определенную информацию, отображаемую во всех log-файлах, с помощью дополнительных глобальных параметров:

- **debug timestamp.** Добавляет отметку времени к сообщению в журнале; этот параметр включен по умолчанию.
- **debug uid.** Записывает идентификаторы пользователя и группы, с которыми был запущен процесс Samba, сгенерировавший событие.
- **debug prefix timestamp.** Оставляет отметки времени, но удаляет информацию о месте в исходном коде Samba, которое сгенерировало событие.
- **debug pid.** Записывает идентификатор процесса Samba, сгенерировавшего событие.
- **debug hires timestamp.** Записывает метки времени с точностью до миллисекунд, а не секунд.
- **debug class.** Записывает класс сообщения, который помогает в тех случаях, когда необходимо повысить подробность событий определенного класса (эта опция позволяет вам определить, какой класс вам нужен).

Журналирование может помочь обнаружить проблему, а может высыпать на вас тонны ненужного мусора. У Samba имеется множество различных параметров журналирования – используйте их с умом.

## Трассировка системных вызовов

Если ничего так и не помогло решить проблему, то можно использовать системные инструменты UNIX, чтобы взглянуть, что происходит внутри процесса. Linux-приложение **strace** позволяет выполнять трассировку всех системных вызовов, которые генерирует приложение. Программы используют системные вызовы для открытия и чтения файлов, создания и удаления процессов, а также для взаимодействия с остальными компонентами операционной системы.

В листинге 11 показано, как пользователь root выполняет трассировку процесса Samba, посылающего ошибку клиенту.

### Листинг 11. Трассировка процесса с помощью strace

```
# ps -ef | grep smb
sean    13375 28812  0 21:54 ?          00:00:00 smbd -D
root    14294 13593  0 21:55 pts/2      00:00:00 grep smb
root    16132 28812  0 Feb27 ?          00:00:36 smbd -D
root    28812    1  0 Feb14 ?          00:00:28 smbd -D
root    28814 28812  0 Feb14 ?          00:00:00 smbd -D
[root@bob /]# strace -e trace=file -p 13375
Process 13375 attached - interrupt to quit
<< Output omitted >>
chdir("/home/sean")                      = 0
stat64("somedir", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
stat64("somedir/*", 0xbfc5f60)            = -1 EACCES (Permission denied)
getcwd("/home/sean", 4096)                 = 11
lstat64("/home/sean/somedir", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
lstat64("/home/sean/somedir/*", 0xbfc5fffc) = -1 EACCES (Permission denied)
```

Первая команда выводит список всех процессов Samba. Поскольку Samba присваивает идентификатор подключенного пользователя, легко понять, что процесс 13375 принадлежит

пользователю. Далее запускается команда `strace` с двумя параметрами. Первый параметр, `-e trace=file`, ограничивает вывод только системными вызовами, выполняющими действия с файлами. Для решения определенных типов проблем, с которыми вы можете столкнуться, неплохо начать именно с этого. Второй параметр, `-p 13375`, указывает команде `strace` подключиться к запущенному процессу с указанным идентификатором.

Если посмотреть на вывод этой команды, то видно, что `smb` постоянно проверяет директории предмет изменения. Если пользователь пытается выполнить действие, приводящее к ошибке, то можно увидеть нечто похожее на содержимое [листиングа 11](#). Последние несколько команд пытаются получить информацию о файле, находящемся в директории, с помощью вызова `stat64`. В результате выдается сообщение `permission denied`, которое означает, что доступ пользователю был запрещен на системном уровне, а не сервером Samba. Эта команда может предоставить дополнительную информацию для решения проблемы, например, информацию об изменении атрибутов директории или о том, что пользователю запрещен доступ в директорию.

### Что дальше

Эта статья завершает тему конфигурирования Samba. Следующая статья этой серии содержит материалы цели 312.2 темы 312. В ней будет рассказано, как создавать и настраивать общие ресурсы, и получать доступ к ним с других компьютеров.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Configure Samba](#) (EN).
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- Чтобы получить ссылки на все статьи этой серии, которые помогут вам подготовиться к успешной сдаче экзамена LPI-302, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#).
- В статье [Quantify performance changes using application tracing](#) (EN) (developerWorks, июль 2006 г.) объясняется, как использовать утилиты `truss` и `strace` для отслеживания системных вызовов.
- [Журнализование действий пользователей](#) (EN) можно выполнять на уровне виртуальной файловой системы Samba.
- Онлайновая [man-страница smb.conf](#) (EN) более удобна в использовании, чем текстовая версия.
- В [разделе Linux](#) сайта developerWorks можно найти сотни [пошаговых инструкций и руководств](#), загрузить программные продукты, а также получить ссылки на форумы и многие другие ресурсы, ориентированные на разработчиков и администраторов Linux.

### Получить продукты и технологии

- [Репозиторий Samba Git repository](#) (EN) поможет вам выяснить текущее состояние

Samba.

- Загрузите [последнюю версию Samba](#) (EN), чтобы использовать все ее новые возможности.

# Изучаем Linux, 302 (смешанные среды): Файловые службы

*Создание и настройка общих файловых ресурсов в смешанной среде*

[Шон Уолберг](#), старший сетевой инженер, P.Eng

**Описание:** Эта статья поможет вам подготовиться к сдаче экзамена 302 сертификационной программы института Linux Professional Institute (LPI). Из этой статьи вы узнаете о создании общих файловых ресурсов и настройке файловых служб Samba в смешанной среде.

[Больше статей из этой серии](#)

**Дата:** 10.05.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Создание и настройка общего доступа к файлам.
- Планирование миграции файловых служб.
- Скрытие административного общего ресурса IPC\$.
- Создание сценариев для поддержки работы пользователей и групп с общими файловыми ресурсами.
- Использование инструментов командной строки для работы с общими файловыми ресурсами.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 312.2 темы 312. Цель имеет вес 4.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. Кроме этого, у вас должен быть доступ к среде Windows, которую можно использовать для проверки доступа к файлам и принтерам.

## Создание общих файловых ресурсов

В [предыдущей статье](#) говорилось о том, что все имена разделов в файле smb.conf, за исключением `homes`, `printers` и `global`, воспринимаются Samba как имена общих

ресурсов. Помимо имени общего ресурса важным параметром является его сопоставление с директорией на диске. Ниже приведен пример простейшего общего ресурса, доступного для использования:

```
[tmpdir]
path = /var/tmp
```

В этом примере был создан общий ресурс с именем *tmpdir*, сопоставленный с директорией */var/tmp* на сервере Samba. Если бы, например, этот сервер назывался *phoenix*, то пользователи смогли бы получить доступ к файловому ресурсу, используя UNC-путь *\phoenix\tmpdir*. Однако такая редко встречающаяся конфигурация не очень удобна: по умолчанию доступ к общим ресурсам предоставляется в режиме "только для чтения", поэтому запись данных в созданный нами общий ресурс невозможна.

### Использование параметров безопасности

В Samba имеется множество параметров безопасности, которые определяют, какие ресурсы должны быть доступны и откуда. Многие из этих параметров влияют на взаимодействие Samba и разрешений файловой системы UNIX, и не рассматриваются в этой статье. Тем не менее, мы рассмотрим несколько основных параметров.

Серверы SMB (Server Message Block), включая собственные реализации Microsoft, имеют общий ресурс *IPC\$*, являющийся общим ресурсом межпроцессного взаимодействия и использующийся для выполнения программных функций через сеть. Знак доллара (\$) в конце имени ресурса означает, что этот ресурс является скрытым и не виден клиентам Microsoft, даже если сервер будет сообщать о его наличии.

Samba создает общий ресурс *IPC\$* даже если он не создан в файле *smb.conf*. Если вы создадите этот ресурс, то сможете управлять доступом к нему. Ниже приведен пример общего ресурса *IPC\$* с ограниченным доступом:

```
[IPC$]
hosts allow = 192.168.1.0/255.255.255.0
browsable = no
```

Этот фрагмент кода создает общий ресурс *IPC\$* и разрешает обращаться к нему только из подсети 192.168.1.0. Далее устанавливается параметр *browsable*, который определяет, должен ли этот ресурс быть видимым для клиентов, когда они запрашивают его, или нет.

Некоторое время назад было принято скрывать важные общие ресурсы, такие как *IPC\$*, в надежде на то, что злоумышленники могли не заметить их. Этот подход не стоит ограничивать только ресурсом *IPC\$*. Например, к ресурсу *IPC\$* необходимо подключаться для получения списка общих ресурсов сервера. Если вы посмотрите на пакеты отладки и трассировки Samba, то увидите, что параметр *browsable* ни на что не влияет. Здесь я рассказал о нем лишь потому, что данный вопрос упоминается в программе экзамена. Гораздо лучше ограничивать доступ списком хостов, которым разрешено подключаться к ресурсу *IPC\$*, а не пытаться спрятать его.

Пользователь может подключиться к общему ресурсу, не указывая свое имя, становясь, таким образом, *гостем*. По умолчанию гостевой доступ к общим ресурсам не разрешен, но его можно разрешить, установив параметр *guest ok = yes* на уровне общего ресурса. По умолчанию гостевым пользователям присваивается имя *nobody*, которое можно изменить с помощью глобального параметра *guest account*.

Для связи двух областей файловой системы в UNIX широко используются символические

ссылки. Например, в домашней директории можно создать символьическую ссылку на директорию системных временных файлов и использовать ее как часть дерева домашней директории. Samba разрешает переходить по таким символьическим ссылкам, позволяя получать доступ к областям файловой системы, находящимся за пределами общего файлового ресурса. Если вы хотите запретить эту возможность, используйте параметр `follow symlinks = no` на уровне общего ресурса.

Если вы хотите ограничить доступ пользователей к общему ресурсу, используйте параметр `valid users`. Например, параметр `valid users = sean, jon, isaac` разрешает получать доступ к общему ресурсу, для которого он был установлен, только трем указанным пользователям. Этот параметр можно использовать в дополнение к файловым разрешениям для дополнительной защиты важных ресурсов.

## Домашние директории

Обычно принято предоставлять пользователям домашние директории для хранения их личных файлов. Каждому пользователю, записи о котором присутствует в файле паролей UNIX, назначена домашняя директория. Используя раздел `[homes]`, в Samba можно экспорттировать любое количество домашних директорий без необходимости создавать отдельный раздел конфигурации для каждой домашней директории. Если кто-либо обращается к общему ресурсу с именем `joe`, то Samba начинает искать общий ресурс с таким именем. Если такой общий ресурс не найден, Samba выполняет поиск пользователя с тем же именем. Если этот пользователь обнаруживается, то Samba использует раздел `[homes]` конфигурационного файла в качестве шаблона для этого общего ресурса.

В листинге 1 показан типовой раздел `[homes]`.

### Листинг 1. Шаблон для домашних директорий пользователей

```
[homes]
comment = Home Directories
writable = yes
browsable = yes
valid users = %S
```

Каждая строка конфигурации в листинге 1 означает следующее:

- Начало конфигурирования раздела `homes`.
- Создание комментария, который будут видеть пользователи, просматривающие подробные сведения о доступных файловых ресурсах этого сервера.
- Установка разрешения на запись, которое позволяет пользователям изменять содержимое их домашних директорий.
- Установка флага, означающего, что этот ресурс будет виден в списке общих ресурсов, запрашиваемом пользователями; пользователи, указавшие свое имя, будут видеть как общий ресурс, так и домашнюю директорию.
- Разрешает доступ к общему ресурсу только тому пользователю, которому этот ресурс принадлежит.

Обратите внимание на макрос `%S` в листинге 1. Этот макрос преобразуется в имя общего ресурса. Поскольку имя пользователя и имя общего ресурса совпадают, то параметр `valid users` разрешает использовать ресурс только его владельцу.

Теперь каждый раз, когда пользователи просматривают список общих ресурсов, они видят свои домашние директории, к которым они могут подключиться. Общий ресурс будет сопоставлен с их домашней директорией UNIX.

Другая интересная особенность, связанная с домашними директориями, заключается в том, что подключаясь непосредственно к ресурсу `\server\homes`, вы попадаете в свою собственную домашнюю директорию, как если бы вы подключались к ресурсу `\server\имя_пользователя`. Эта дополнительная возможность была реализована командой разработчиков Samba для того, чтобы помочь пользователям, использующим компьютер совместно, и избавить от путаницы.

### **Сценарии для добавления пользователей и групп**

Если вы посмотрите на список пользователей домена Microsoft, то может оказаться, что пользователь, подключающийся к вашему серверу, не имеет локальной учетной записи UNIX. Один из способов решения этой проблемы заключается в том, чтобы с помощью параметра `add user script` заставить Samba автоматически создавать для пользователя учетную запись при его подключении. Макрос `%U` преобразуется в имя подключающегося пользователя. Можно использовать системную утилиту `useradd` или написать собственные сценарии.

Для групп существует похожий параметр `add group script`, используемый при работе с инструментами Microsoft, предназначенными для управления экземпляром Samba. В этом случае также могут оказаться сценарии `add user to group` и `delete`. Полный список автоматизируемых задач вы можете найти на man-странице `smb.conf`.

Автоматическое создание пользователей по запросу не всегда является наилучшей идеей. Лучшим решением может оказаться использование механизмов аутентификации `winbind` или LDAP, позволяющих Samba и Linux использовать общую базу данных пользователей.

### **Символы в верхнем и нижнем регистре**

В среде Microsoft регистр символов в именах файлов и директорий не имеет значения. Таким образом, `FILE`, `file` и `FiLe` указывают на один и тот же файл. Однако в Linux регистр символов имеет значение, и в данном случае `FILE`, `file` и `FiLe` – это три разных файла. Для разрешения конфликтов Samba должна знать, как в таких ситуациях следует сопоставлять имена различных сред. Процесс сопоставления регистров является частью более общего процесса под названием *преобразование имен*.

Преобразованием регистров имен файлов управляют несколько параметров. Самым важным из них является параметр `case sensitive`, который может принимать значения `yes`, `no` или `auto`. Если параметр `case sensitive` включен, то Samba использует символы в том регистре, в котором они указаны клиентами. Если этот параметр отключен, то Samba выполняет поиск в директории без учета регистра.

Одна из проблем, связанных с регистрозависимостью имен, заключается в том, что если параметр установлен неверно, то вы можете не получить доступ к некоторым файлам. Рассмотрим для примера директорию, содержащую два файла с именами `test` и `TEST`. Если Samba не использует регистрозависимый доступ, то невозможно определить разницу между двумя этими файлами.

По умолчанию этот параметр имеет значение `auto`; в этом случае выполняется поиск клиентского поля расширения, сообщающего о том, поддерживает ли клиент регистрозависимый доступ или нет. Клиенты Windows не поддерживают эту функциональность, поэтому они не будут учитывать регистр символов.

Параметры `default case` и `preserve case` работают в связке друг с другом. Если параметр `preserve case` установлен в `yes`, то используется регистр, указанный клиентом. Если параметр `preserve case` установлен в `no`, то для определения регистра вновь создаваемого файла используется значение, установленное параметром `default`.

## Включение функции Usershare

С помощью функции *Usershare* пользователи могут создавать свои собственные общие ресурсы без необходимости редактирования файла `smb.conf`. После того, как администратор включит эту функцию, обычные пользователи могут использовать средства командной строки для экспорта выбранных ими директорий, а также для удаления общих ресурсов, если они станут не нужны.

Для включения функции *Usershare* сначала необходимо разрешить ее использование на глобальном уровне. В листинге 2 показан фрагмент файла `smb.conf`, включающий функцию *Usershare*.

### Листинг 2. Включение функции Usershare

```
[global]
usershare path = /var/lib/samba/usershares
usershare max shares = 5
usershare prefix allow list = /home
usershare prefix deny list = /var, /usr
```

В листинге 2 настраиваются параметры раздела `[global]`. Сначала с помощью параметра `usershare path` определяется директория, которую Samba будет использовать для конфигурации пользовательских общих ресурсов. Для этой директории существует несколько ограничений, о которых будет рассказано позже. Далее устанавливается ограничение на количество пользовательских общих ресурсов. Два последних параметра показывают, как задать ограничения для директорий, которые будут находиться в общем доступе. Параметр `usershare prefix allow list` ограничивает все общие ресурсы указанной директорией — в нашем случае общие ресурсы должны находиться в директории `/home`. Параметр `usershare prefix deny list` имеет противоположное значение и разрешает все за исключением указанных директорий.

Samba налагает два ограничения на пользовательские общие ресурсы. Во-первых, директория, указанная в параметре `usershare path`, должна быть доступна для записи пользователю, создающему общий ресурс, и иметь установленный бит закрепления (`1000` или `+t`). Во-вторых, если параметр `usershare owner only` не установлен в `false`, то пользователь должен быть владельцем директории, к которой он предоставляет общий доступ.

Первое ограничение связано с разрешениями файловой системы и означает, что при создании директории `usershare path` необходимо быть внимательными. Если вы хотите ограничить доступ к общему ресурсу, разрешив подключаться к нему только пользователям из группы `usershares`, то необходимо выполнить следующие команды:

```
# mkdir -p /var/lib/samba/usershares
# chown root:usershares /var/lib/samba/usershares
# chmod 1770 /var/lib/samba/usershares
```

Первая команда создает конечную директорию и ее родительские директории. Следующая команда назначает владельцами этой директории пользователя `root` и группу `usershares`. Последняя команда устанавливает для директории права на чтение, запись и выполнение для владельца и группы, запрещает доступ для остальных пользователей, а также устанавливает бит закрепления. Таким образом, эту директорию могут использовать только пользователь `root` и члены группы `usershares`, а благодаря биту закрепления, файлы могут быть удалены

только их владельцами.

Настройка общего ресурса, возможно, является самой сложной задачей. Пользователь может выполнить следующую команду:

```
net usershare add docs /home/me/Documents/ "My docs" Everyone:F
```

Эта команда экспортирует директорию /home/me/Documents в общих ресурс с именем *docs* и предоставляет всем полный доступ к ней. Другие команды, которые можно использовать:

- **net usershare list** – выводит список общих ресурсов, созданных пользователем.
- **net usershare info docs** – показывает конфигурацию общего ресурса *docs*.
- **net usershare delete docs** – удаляет общий ресурс *docs*.

## Инструменты командной строки

Samba содержит несколько инструментов командной строки. С помощью утилиты **libsmbclient** из состава Samba были созданы другие широко используемые утилиты SMB/ CIFS.

Одним из наиболее заметных отличий UNIX от Windows является то, что UNIX содержит одну большую файловую систему, тогда как в Windows имеется несколько обозначенных буквами дисков. Утилита **smbclient** позволяет просматривать удаленные общие ресурсы Windows, используя интерфейс, похожий на интерфейс протокола FTP, однако чтобы быть прозрачным для приложений, удаленный общий ресурс Windows должен монтироваться как любая другая файловая система.

В составе Samba имеется утилита **smbmount**, которая иногда переупаковывается в файл *mount.cifs*. Эту команду можно запустить непосредственно в командной строке либо вызвать через команду **mount**. В листинге 3 показан процесс монтирования в Linux удаленного общего ресурса CIFS в качестве обычной файловой системы.

### Листинг 3. Монтирование удаленного общего ресурса CIFS

```
# mount -t cifs '\\\\192.168.1.134\\docs' /mnt -o user=myuser  
Password:  
# mount  
...  
\\\\\\192.168.1.134\\docs on /mnt type cifs (rw)
```

Первая команда монтирует файловую систему CIFS по указанному UNC-пути в точку монтирования /mnt. За исключением использования UNC-пути эта команда выглядит как обычная команда **mount**. Параметры передаются с помощью опции **-O**. Единственной опцией, которая здесь нужна, является имя пользователя. Man-страница *mount.cifs* содержит полный список всех опций, среди которых могут содержаться пароль и имя домена. Если вы не указываете пароль, вам будет предложено ввести его. Наконец, команда **mount** показывает смонтированную файловую систему.

Другая команда – это **smbsh**. Вместо монтирования файловой системы стандартными средствами UNIX **smbsh** перехватывает библиотечные вызовы обращений к файлам и при необходимости перенаправляет запросы к общему ресурсу CIFS. В настоящее время в большинстве систем эта команда отсутствует, поскольку она не так надежна, как

монтирование файловой системы.

## Перемещение общих файловых ресурсов

Если вы перемещаете файловые службы с одного сервера на другой, то пользователи могут не располагать информацией о новом сервере. Samba позволяет настроить сервер таким образом, чтобы он отзывался на другое имя. Например, если вы переносите общие файловые ресурсы с сервера с именем *phoenix* на сервер с именем *fs2*, то можете заставить сервер *fs2* отвечать на запросы, отправляемые серверу *phoenix*. Разумеется, вы должны убедиться в том, что сервер *phoenix* не будет отвечать на эти запросы, выключив или переименовав его.

Для добавления псевдонима сервера используется глобальный параметр ***netbios aliases***. Если вы хотите изменить имя сервера Samba на какое-то другое имя, отличное от UNIX-имени сервера, используйте параметр ***netbios name***.

Команды, относящиеся к монтированию файловых систем, о которых вы уже знаете к этому моменту, также могут пригодиться для копирования файлов с одного сервера на другой в рамках подготовки к миграции. Также могут оказаться полезными утилиты UNIX наподобие ***rsync***.

## Что дальше

Следующая статья этой серии содержит материалы цели 312.3 темы 312. В ней рассматривается создание, настройка и использование служб печати в смешанной среде.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): File services](#) (EN).
- Онлайновая [man-страница \*smb.conf\*](#) (EN) более удобна в использовании, чем текстовая версия.
- Узнайте подробнее о том, как получить доступ к удаленным файловым системам с Linux-компьютера, из документа [SMB HOWTO](#) (EN).
- Если вы используете SELinux, то вам может пригодиться документ [managing Samba and SELinux](#) (EN), особенно при настройке домашних директорий.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.

## Получить продукты и технологии

- Загрузите [последнюю версию Samba](#) (EN), чтобы использовать все ее новые возможности.

# Изучаем Linux, 302 (смешанные среды): Службы печати

*Создание и управление общими ресурсами печати Samba в смешанной среде*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Эта статья поможет вам подготовиться к сдаче экзамена 302 сертификационной программы института Linux Professional Institute (LPI). Из этой статьи вы узнаете о настройке общего доступа к принтерам для клиентов Linux и Microsoft.

[Больше статей из этой серии](#)

**Дата:** 10.05.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

## Краткий обзор

В этой статье рассматриваются следующие темы:

- Создание и настройка общих ресурсов печати.
- Интеграция Samba с системой печати CUPS (Common UNIX® Print System).
- Управление драйверами принтеров Windows® и их загрузкой.
- Настройка общего ресурса [print\$].
- Вопросы безопасности при работе с общими принтерами.
- Настройка и управление системой учета печати.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 312.3 темы 312. Цель имеет вес 2.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. Кроме этого, у вас должен быть доступ к среде Windows, которую можно использовать для проверки доступа к файлам и принтерам.

## Настройка общего ресурса печати

Настройка общего ресурса печати похожа на настройку общего файлового ресурса. Для этого необходимо создать раздел, установить значения нескольких параметров и подключить клиентов. Тем не менее, поскольку в данном случае мы имеем дело с печатью, то существует несколько тонких отличий. В листинге 1 приведена типовая конфигурация, содержащаяся в файле smb.conf.

## Листинг 1. Типовая конфигурация для общих ресурсов печати

```
[global]
  load printers = yes
  printing = sysv
[printers]
  comment = Printers
  path = /var/spool/samba
  writable = no
  printable = yes
```

В листинге 1 не настраиваются никакие принтеры, а просто содержатся инструкции для загрузки списка системных принтеров и определяется раздел с именем **[printers]**, выступающий в роли шаблона. В [предыдущей статье](#) говорилось о том, что раздел **[homes]** используется в качестве шаблона для домашних директорий. Раздел **[printers]** выполняет те же функции для принтеров.

На глобальном уровне параметр **load printers** выдает команду Samba выполнять поиск системных принтеров и настраивать общий доступ к ним. Параметр **printing = sysv** указывает на использование более старой системы печати SysV. Сейчас нам достаточно этого, но позже вы узнаете, как настраивать более современную систему печати CUPS.

Раздел **[printers]** определяет, как будет выглядеть каждый общий ресурс печати. Если у вас нет необходимости использовать отдельные параметры для каждого принтера, например, ограничивать доступ пользователей, то можно работать с этим шаблоном. Параметр **path** используется системой Samba. Когда вы посыпаете задание принтеру Samba, то прежде, чем послать файл на устройство печати, **smbd** помещает его в директорию, указанную в параметре **path**. После этого вы должны убедиться в том, что общий ресурс не находится в режиме **writable**, а находится в режиме **printable** для того, чтобы Samba могла правильно использовать его в качестве ресурса печати.

Теперь вы можете увидеть на вашем сервере список принтеров. Если на клиентском компьютере Windows установлен драйвер этого принтера, то можно продолжить его установку и работать с ним в сети.

## Интеграция Samba с системой печати CUPS

В большинстве дистрибутивов Linux система печати CUPS заменила систему SysV. CUPS является более гибкой и удобной для пользователей по сравнению с SysV, и обеспечивает прозрачную совместимость со старой системой печати. Если вы используете команды **lp** или **lpq**, то, вероятно, вы работаете с CUPS.

Одним из механизмов, обеспечивающих преимущество CUPS в сравнении со старой системой печати, является предварительная обработка заданий печати. Например, если вы посыпаете на печать PDF-файл, то CUPS распознает формат файла, пересыпает вывод через интерпретатор (например, GhostScript) и преобразует файл в язык, который распознается принтером. Раньше для этого требовалось затратить много усилий, но сегодня все это умеет делать CUPS.

Простейшая конфигурация, необходимая для включения системы печати CUPS, показана в листинге 2.

## Листинг 2. Включение поддержки CUPS

```
[global]
  printing = cups
```

```
printcap = cups
```

Параметры листинга 2 устанавливаются в глобальном режиме. Первый параметр обеспечивает работу функций печати через библиотеки CUPS, а не через `lpr`. Второй параметр указывает Samba получить список принтеров от демона CUPS, а не из системного файла `printcap`. Даже если вы используете CUPS, файл `printcap` все равно может присутствовать в системе, поскольку CUPS поддерживает его использование для совместимости с приложениями, не поддерживающими CUPS.

Даже если CUPS интегрирована с Samba, вы можете продолжать использовать инструменты CUPS для управления очередью печати. Если вам не приходилось работать с CUPS, то вам поможет следующий список наиболее важных команд:

- **`lp`.** Отправляет вывод на принтер (обычно вывод передается этой команде по конвейеру от другой команды, например, `cat /etc/motd | lp`).
- **`cupsenable` и `cupsdisable`.** Запуск и остановка принтера, соответственно (эти команды также применяются для сброса принтера).
- **`cupsreject`.** Отклоняет или принимает задание для принтера (эта команда не изменяет статус принтера, а вместо этого сообщает CUPS о необходимости отклонить входящее задание).
- **`lpadmin`.** Позволяет настраивать параметры для принтера, например, устанавливать квоты.
- **`lpq`.** Показывает элементы очереди для указанного принтера.
- **`lprm`.** Позволяет отменять задания принтера.

Печать необработанных (raw) данных и интеллектуальная настройка

Обычно принтеры понимают такие языки описания страниц (Page Description Language, PDL), как Adobe® PostScript® или Printer Command Language (PCL). Некоторые принтеры могут использовать языки описания страниц, разработанные их производителями. Задача по преобразованию посылаемых приложениями команд печати в команды языка описания страниц, с которым работает принтер, возлагается на драйвер принтера Windows. Без драйвера принтера посылаемые приложениями команды распечатываются непосредственно в виде текста, в результате чего на печать выводится один мусор.

Процесс перевода клиентским приложением документа в подходящий для принтера формат называется *печатью необработанных данных* (raw printing). В этом режиме Samba просто копирует полученные от клиента байты информации на принтер. Такой подход имеет два недостатка:

- Для того чтобы клиент смог распечатать данные, на нем должен быть проинсталлирован драйвер нужного принтера.
- Системе Samba трудно понять, какие именно действия выполняются; вследствие этого, например, она не может определить, сколько страниц было распечатано.

Несмотря на эти недостатки, печать необработанных данных легко настраивается. Приложив дополнительные усилия, можно настроить распространение драйверов принтеров с сервера Samba на клиентские компьютеры.

## Установка драйверов на клиентские компьютеры Windows

Заставить Samba инсталлировать драйверы принтеров – непростое дело. Самым лучшим вариантом является установка универсального драйвера PostScript и его использование для всех принтеров. О том, как преобразовать клиентские PostScript-данные в формат, подходящий для использования принтером, позаботится CUPS, используя свою

интеллектуальную систему печати. Реализовав этот подход вместо использования "родных" драйверов, вы также сможете подробно узнать о том, какое количество страниц было распечатано.

Вам потребуется загрузить пакет с драйверами CUPS (текущая версия – cups-windows-6.0-1.i386.rpm) ссылку на который вы найдете в разделе [Ресурсы](#). Этот пакет устанавливает в директорию /usr/share/cups/drivers следующие драйверы:

- cups6.inf
- cups6.ini
- cupsps6.dll
- cupsui6.dll

Драйверы CUPS поддерживают только клиентов Microsoft Windows 2000 и выше; обычно все работает хорошо, однако более поздняя утилита Samba жестко запрограммирована на поиск устаревших драйверов Microsoft и не работает в случае их отсутствия. В вашей системе должны присутствовать следующие файлы, которые также должны быть скопированы в директорию /usr/share/cups/drivers:

- ps5ui.dll
- pscript5.dll
- pscript.hlp
- pscript.ntf

Обычно эти файлы находятся в директории C:\WINDOWS\ServicePackFiles\i386.

Когда все эти восемь файлов будут скопированы в директорию драйверов CUPS, мы создадим общий ресурс print\$. Этот общий ресурс жестко задан для всех клиентов Windows, которые ищут в нем драйверы принтера после того, как он установлен. Конфигурация этого ресурса не содержит ничего необычного и показана в листинге 3.

### Листинг 3. Общий ресурс print\$

```
[print$]
comment = Printer Driver Export
path = /etc/samba/drivers
browseable = yes
guest ok = no
read only = yes
write list = root
```

Конфигурация в листинге 3 настраивает простой общий ресурс, доступный в режиме "только для чтения" всем пользователям, за исключением пользователя root. Не забудьте перезапустить Samba для вступления изменений в силу.

Наконец, установим драйверы принтера с помощью команды **cupsaddsmb**. Если ваш общий ресурс печати называется Downstairs\_Laser, то просто запустите команду **cupsaddsmb -v Downstairs\_Laser**. Вам будет предложено ввести пароль пользователя root, после чего вы увидите на экране поток выполняющихся действий.

Теперь клиенты могут подключаться к серверу Samba и дважды щелкнуть значок принтера. Они смогут использовать принтер без выполнения дополнительных действий (например, идентификация принтера или установка драйвера).

### Драйверы для принтеров определенных производителей

Для того чтобы система CUPS понимала, что именно распечатывается, следует использовать

универсальный драйвер CUPS PostScript. Если вы решите установить на клиентские компьютеры драйверы Windows, то процедура будет похожей. Во-первых, в этом случае не используется команда `cupsaddsmb`. Вы вручную копируете драйверы в директорию `/etc/samba/drivers`, а также в директорию, соответствующую архитектуре (в предыдущем случае за вас это делала утилита `cupsaddsmb`). Например, для 32-разрядных драйверов используется директория `W32X86`.

Для того чтобы сообщить Samba о драйвере, используется команда `rpcclient`. Вы должны указать имена файлов драйвера принтера и официальное имя принтера. Практический пример есть в разделе [Ресурсы](#).

## Ведение учета и безопасность

Применительно к принтерам *ведение учета* означает отслеживание и установку квот на допустимое количество используемой бумаги для каждого пользователя. *Безопасность* означает возможность знать, кто использует принтеры, и при необходимости ограничивать доступ к ним. Под доступом к принтеру можно понимать возможность распечатывать документы на этом принтере или возможность отменять задания печати других пользователей.

## Безопасность принтеров

Задачей Samba в процессе вывода задания на печать является получение от клиента файла, предназначенного для печати, например, PostScript, и передача его подсистеме CUPS. Все действия, которые выполняет Samba, например, отображение клиенту статуса очереди печати, выполняются через подсистему CUPS, которая запрашивает необходимые файлы и управляет принтерами. Samba является лишь связующим звеном между CUPS и клиентами.

Можно заставить клиентов печатать напрямую через CUPS, используя протокол печати через Интернет (IPP, Internet Printing Protocol). Тем не менее, нельзя недооценивать простоту установки принтеров из сетевого окружения Microsoft Network Neighborhood. Просто помните о том, что любое управление на уровне Samba можно потенциально переопределить на уровне подсистемы CUPS, если клиенты подключаются напрямую через нее.

Предположим, что общий ресурс печати защищен с помощью параметра `valid users = alice, bob`, который позволяет отправлять задания на печать только двум пользователям. Если пользователь `mallory` попытается напечатать документ через Samba, ему будет отказано в доступе. Если подсистема CUPS не была настроена с такими же разрешениями, то пользователь `mallory` может отправить документ на печать через очередь CUPS.

Другая проблема безопасности связана с гостевыми пользователями. Если вы можете перечислить всех пользователей, которым необходим доступ к принтеру, то убедитесь, что в разделе конфигурации `[printers]` установлен параметр `guest ok = no`. После этого только прошедшие аутентификацию пользователи смогут отправлять задания на печать. В противном случае вашими принтерами смогут воспользоваться даже посторонние пользователи.

## Квоты печати

CUPS, а не Samba, управляет не только заданиями печати, но и ведением учета. CUPS позволяет использовать *квоты печати*, которые ограничивают количество страниц, которое может распечатать отдельный пользователь за определенный интервал времени. В листинге 4 показано, как установить квоту для принтера.

## Листинг 4. Установка квоты для принтера

```
# lpadmin -p Downstairs_Laser -o job-quota-period=604800 -o job-page-limit=100 \
```

```
job-k-limit=50000
```

В листинге 4 для принтера задаются три параметра. Первый – это временной интервал квоты, равный 604 800 секундам (т. е. 1 неделе). Второй параметр – это ограничение на количество страниц, распечатываемых за указанный временной интервал (в нашем случае 100 страниц). Третий параметр ограничивает объем распечатываемых данных 50 мегабайтами. Поскольку трудно оценить данный объем, то последний параметр является необязательным, хотя он позволяет ограничивать объем распечатываемых графических изображений.

Для проверки установленной для принтера квоты загляните в файл /etc/cups/printers.conf, в котором вы увидите параметры `QuotaPeriod`, `PageLimit` и `KLimit` с только что заданными значениями. Чтобы убрать все квоты, установите эти параметры в 0.

### Ведение учета печати

Каждое задание печати записывает все события в файл `page.log`, который обычно расположен в директории /var/log/cups. В листинге 5 показана типовая строка этого файла.

### Листинг 5. Типовая запись журнала учета печати

```
Downstairs_Laser 755 sean [26/Apr/2011:15:02:27 -0500] 1 1 - localhost smbprn.0019.FWqosE
```

В левой части строки содержится наиболее важная информация. Перечислим все поля по порядку:

- Имя принтера.
- Номер, назначенный заданию печати.
- Пользователь, распечатавший документ.
- Дата в формате GMT и часовой пояс.
- Количество напечатанных страниц.
- Количество отправленных на печать копий.

Таким образом, количество использованных страниц – это количество напечатанных страниц помноженное на количество отправленных копий. Подробное описание остальных полей журнала вы найдете в разделе [Ресурсы](#).

**Примечание.** Самым серьезным ограничением квот системы CUPS является то, что они применяются ко всем пользователям.

### Что дальше

Следующая статья этой серии содержит материалы цели 312.4 темы 312. В ней рассматривается использование Samba в качестве основного и резервного контроллеров домена Microsoft.

### Ресурсы

#### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Print services \(EN\)](#).
- Онлайновая [man-страница smb.conf](#) (EN) более удобна в использовании, чем текстовая версия.
- Прочитайте материал [темы 107](#) (EN) учебного руководства developerWorks, чтобы освежить в памяти управление печатью из командной строки.

- В [главе 21](#) (EN) руководства Samba-HOWTO описана "классическая" поддержка печати, которая использовалась до появления CUPS. На этом Web-сайте также представлен практический пример добавления принтера драйвера Windows.
- В [главе 22](#) (EN) руководства Samba-HOWTO содержится дополнительная информация об интеграции CUPS и Samba.
- В разделе [page log](#) (EN) документации CUPS перечислены все поля журнала печати, на тот случай, если вам понадобится собрать какие-то дополнительные сведения.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.

#### **Получить продукты и технологии**

- Загрузите [драйвер CUPS для Windows](#) (EN).
- Загрузите [последнюю версию Samba](#) (EN), чтобы использовать все ее новые возможности.

## **Изучаем Linux, 302 (смешанные среды): Управление доменом**

*Использование Samba для сетевой аутентификации*

[Родерик Смит \(Roderick Smith\)](#), автор и консультант, IBM

**Описание:** С точки зрения SMB/CIFS домен и рабочая группа ничем не отличаются – в обоих случаях это единый набор компьютеров, обычно находящихся в одной локальной сети. Однако в домене имеется специальный компьютер – *контроллер домена*, который управляет учетными записями и подключениями клиентов ко всем серверам домена, а также выполняет ряд дополнительных ролей. Samba может работать в качестве контроллера домена, но для этого необходимо настроить ряд параметров.

[Больше статей из этой серии](#)

**Дата:** 07.06.2012

**Уровень сложности:** средний

# Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

## Краткий обзор

В этой статье рассматриваются следующие темы:

- Членство в домене.
- Настройка первичного контроллера домена.
- Настройка резервного контроллера домена.
- Добавление компьютеров в домен.
- Управление сценариями входа в систему.
- Управление перемещаемыми профилями.
- Управление системными политиками.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 312.4 темы 312. Цель имеет вес 4.

## Предварительные требования

Чтобы извлечь наибольшую пользу из этой статьи, необходимо обладать практическими навыками работы с инструментами командной строки Linux и знать основы конфигурирования Samba. Вы должны знать общую структуру конфигурационного файла smb.conf и уметь редактировать его в каком-либо текстовом редакторе. Вы должны уметь настроить сервер Samba для работы с файлами.

## Настройка основных доменных функций

Основная функция контроллера домена заключается в управлении проверкой подлинности других компьютеров. Для этого сервер Samba должен уметь принимать определенные типы данных аутентификации от клиентов и соответствующим образом отвечать им. Для включения этой возможности необходимо настроить несколько параметров в файле smb.conf. На практике контроллеры домена выполняют дополнительные роли в сетевом окружении, поэтому может потребоваться настроить дополнительные параметры Samba.

Прежде чем двигаться дальше, необходимо рассказать о взаимодействии компьютеров в домене Microsoft® Windows NT®. Контроллер домена является ядром такой сети. Клиентами домена могут быть файловые серверы или серверы печати под управлением Samba, операционной системы Windows® или другого программного обеспечения. Эти компьютеры называются *серверами, являющимися членами домена*, и являются как серверами (по отношению к рабочим станциям пользователей), так и клиентами (по отношению к контроллеру домена). Архитектура сетевого окружения может повлиять на эти отношения. Например, в одноранговой сети один и тот же компьютер может являться как сервером-членом домена, так и клиентом файлового сервера. Контроллер домена может работать в качестве файлового сервера и даже в качестве клиента.

## Настройка обязательных доменных функций

Минимальная конфигурация для работы Samba в качестве контроллера домена предполагает настройку следующих параметров smb.conf.

```
workgroup = EXAMPLE
security = User
encrypt passwords = Yes
passdb backend = tdb:sam:/etc/samba/private/passdb.tdb
domain logons = Yes
admin users = ntadmin
```

Некоторые из этих параметров при необходимости можно изменить, а некоторые – нет. Рассмотрим их подробно:

- **workgroup** – задает имя домена Windows NT. *Домен* – это просто рабочая группа с дополнительными возможностями.
- **security** – этот параметр должен иметь значение **User**.
- **encrypt passwords** – этот параметр должен иметь значение **Yes**.
- **passdb backend** – этот параметр может иметь любое допустимое значение, однако, если предполагается использовать и основной, и резервный контроллеры домена, то, возможно, потребуется задать для него определенное значение в соответствии с разделом [Настройка резервного контроллера домена](#).
- **domain logons** – этот параметр должен иметь значение **Yes**.
- **admin users** – задает имена одного или нескольких пользователей (в нашем примере **ntadmin**), которые будут являться администраторами. Эти пользователи будут иметь привилегии пользователя root для любых общих ресурсов, к которым они подключаются. Чтобы иметь возможность присоединять новые компьютеры к домену, необходимо указать администраторов либо таким способом, либо добавить пользователя root в базу учетных записей Samba.

Эти параметры накладывают определенные условия, а именно: сервер Samba должен содержать локальные учетные записи Linux всех пользователей, которых он должен аутентифицировать, и эти учетные записи также должны присутствовать в базе паролей Samba. Как только Samba будет настроена в соответствии с этими требованиями, она начнет принимать доменные учетные записи компьютеров под управлением Microsoft Windows 9x/Me или компьютеров Samba, настроенных с параметром **security = Server**. Для обеспечения работы с учетными записями компьютеров под управлением более новых ОС Windows или компьютеров Samba, настроенной с параметром **security = Domain**, необходимо создать доверительные учетные записи компьютеров.

### **Создание доверительных учетных записей компьютеров**

Любой компьютер можно свободно добавить в сетевое окружение рабочей группы, однако для получения всех преимуществ доменного окружения компьютер должен являться полноправным членом домена. С точки зрения пользователя главное преимущество доменного окружения заключается в использовании *механизма единого входа* (single sign-on), которая означает, что пользователю достаточно один раз ввести имя и пароль, чтобы получить доступ к любому серверу, также являющемуся членом домена. С точки зрения администрирования Samba членство в домене позволяет использовать на рядовых серверах домена параметр конфигурации **security = Domain**. Этот параметр обеспечивает более надежную защиту по сравнению с параметром **security = Server**, хотя для полного присоединения компьютера к домену требуется выполнить ряд дополнительных действий.

Для полного присоединения к домену компьютер (как рядовой сервер домена, так и клиентская рабочая станция) должен иметь собственную учетную запись на контроллере домена. К самим контроллерам домена это требование не предъявляется. Учетная запись компьютера никак не связана с учетными записями работающих на нем пользователей и называется *доверительной учетной записью компьютера*.

Лучше всего использовать отдельную группу Linux, которая будет содержать доверительные учетные записи всех компьютеров. Следующая команда создает группу с именем **trust**, которая и будет использоваться для этих целей:

```
# groupadd -r trust
```

После того, как группа доверительных учетных записей компьютеров создана, можно переходить к созданию самих учетных записей – по одной для каждого члена домена (клиента или сервера) в вашей сети. В качестве имен учетных записей вы должны использовать NetBIOS-имена компьютеров, преобразованные в нижний регистр и оканчивающиеся знаком доллара (\$). Например, если NetBIOS-имя вашего компьютера WEMBLETH, то его доверительная учетная запись будет называться wembleth\$. Для создания учетных записей можно использовать команды **useradd** и **smbpasswd**, как показано ниже:

```
# useradd -d /dev/null -M -g trust -s /bin/false wembleth$  
# smbpasswd -a -m wembleth$
```

В этом примере мы создали доверительную учетную запись компьютера с именем *wembleth\$*, задали для нее домашнюю директорию */dev/null* (-d /dev/null), указали, что эту директорию не нужно создавать (-M), добавили учетную запись в группу **trust** (-g **trust**) и установили в качестве командной оболочки значение */bin/false* (-s /bin/false). Во второй команде для этой учетной записи была создана запись в базе паролей Samba и указан ее тип с помощью параметра -m (доверительная учетная запись компьютера).

Если вы создали нового пользователя для параметра **admin user**, то необходимо задать для него пароль Samba с помощью команды **smbpasswd** (аналогично команде **smbpasswd ntadmin** для задания его пользовательского пароля). Эта учетная запись будет использоваться при добавлении компьютеров в домен и обладает высокими привилегиями Samba. Если вы не намерены использовать эту учетную запись для выполнения других задач, то мы рекомендуем закомментировать строку **admin users** в файле *smb.conf* после того, как вы присоедините все клиентские компьютеры к домену.

### Настройка дополнительных доменных функций

Конфигурация, рассмотренная в предыдущем разделе, является достаточной для выполнения наиболее важных функций контроллера домена, тем не менее контроллеры домена, как правило, выполняют дополнительные роли в сетях NetBIOS. В частности, контроллер домена может выступать в роли главного обозревателя домена, сервера Windows Internet Name Service (WINS, другой вариант названия – NetBIOS Name Server) или сервера времени. Эти дополнительные службы можно настроить с помощью следующих параметров:

```
domain master = Yes  
preferred master = Yes  
os level = 65  
wins support = Yes  
time server = Yes
```

После того, как эти изменения будут сохранены в файле *smb.conf*, Samba по прошествии определенного времени обнаружит их и изменит свой режим работы. Если вы хотите ускорить этот процесс, то для немедленной перезагрузки конфигурации можете послать серверу сигнал **SIGHUP** или использовать сценарии инициализации SysV вашего

дистрибутива.

## Добавление компьютеров в домен

К этому моменту наша конфигурация обеспечивает полноценную работу Samba в качестве контроллера домена, и мы можем приступать к добавлению компьютеров в домен. То, каким образом это делается, зависит от операционной системы, установленной на сервере-члене домена или клиентской рабочей станции. В случае использования операционной системы Windows 7 необходимо выполнить следующие действия:

1. Добавьте в системный реестр следующие записи:

```
HKEY_LOCAL_MACHINE\System\CCS\Services\LanmanWorkstation\Parameters  
DWORD DomainCompatibilityMode = 1  
DWORD DNSNameResolutionRequired = 0
```

Вместо этого вы можете загрузить файл Win7\_Samba3DomainMember.reg и дважды щелкнуть на нем для его запуска. Это действие не нужно выполнять в Windows Vista® или более ранних версиях Windows.

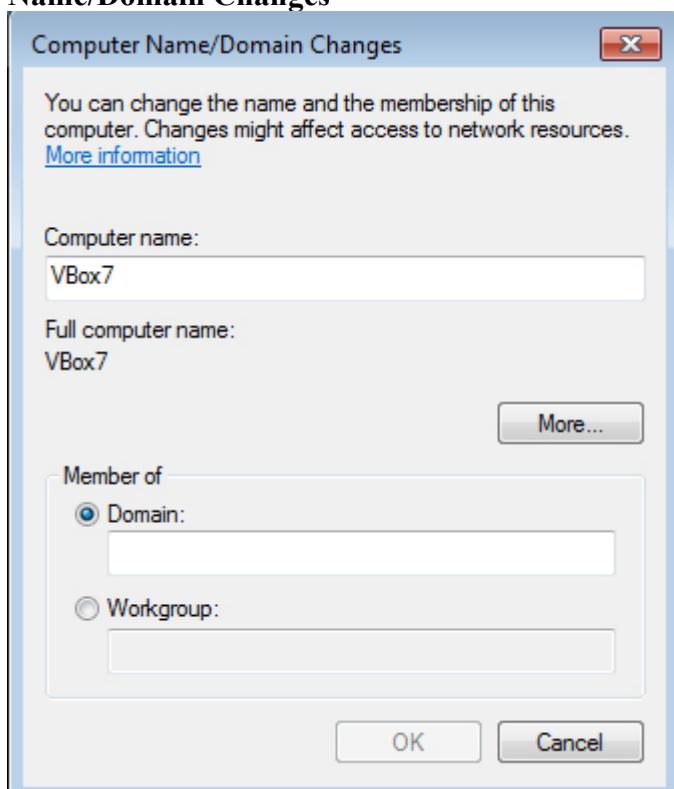
2. В панели Control Panel System запустите оснастку Security.
3. Нажмите **System**.
4. Нажмите **Change Settings** в разделе **Computer Name, Domain, and Workgroup Settings**.

Откроется окно свойств системы.

5. Нажмите **Change**.

Откроется окно **Computer Name/Domain Changes**, изображенное на рисунке 1.

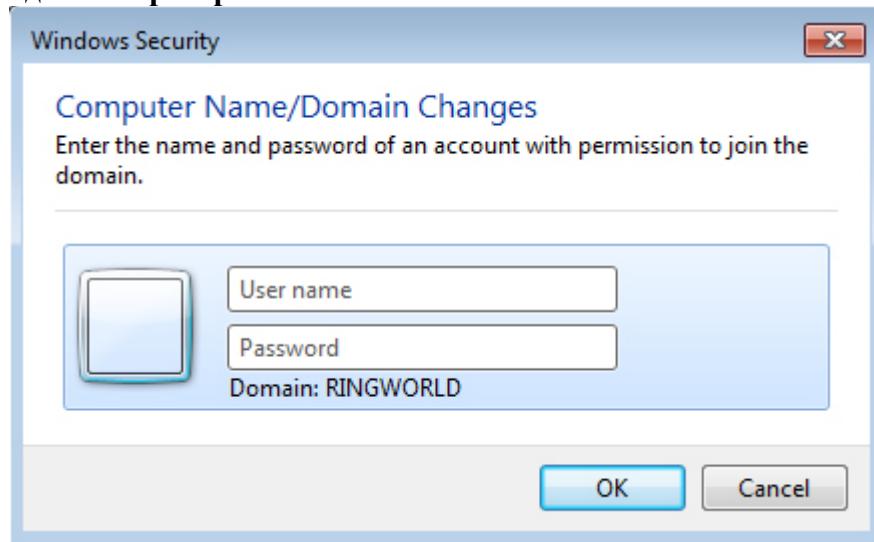
**Рисунок 1. Присоединение компьютера к домену с помощью окна Computer Name/Domain Changes**



6. Выберите параметр **Domain** и введите в текстовом поле имя домена.
7. Нажмите **OK**.

Откроется окно **Windows Security**, изображенное на рисунке 2.

**Рисунок 2. Окно Windows Security с запросом на ввод учетных данных администратора**



8. Введите имя и пароль учетной записи администратора Samba, указанной в параметре **admin users**, и нажмите **OK**.

Вам будет предложено перезагрузить компьютер, после чего он будет присоединен к домену.

**Примечание.** Не все версии Windows могут быть присоединены к домену. Например, Домашние (Home) версии Windows можно использовать только в рабочих группах.

После того, как компьютер будет присоединен к домену и перезагружен, вы увидите окно входа с приглашением нажать Ctrl-Alt-Del для входа в систему. Чтобы войти в систему, используйте имя и пароль учетной записи, зарегистрированной на сервере Samba. После этого вы сможете получить доступ к любому компьютеру, который является членом этого же домена; для авторизации будет использоваться ваша учетная запись, которую вы указали при входе в систему, и, таким образом, вам не придется вводить пароль повторно. Если вы настроили на компьютере общие файловые ресурсы или принтеры, то для аутентификации компьютер будет использовать сервер Samba.

Если вы хотите, чтобы сервер Samba функционировал в качестве обычного рядового сервера, то его также необходимо присоединить к домену. Для этого первым делом необходимо настроить в его конфигурационном файле smb.conf следующие параметры:

```
password server = SERVERNAME
domain logons = No
encrypt passwords = Yes
security = Domain
domain master = No
preferred master = No
os level = 1
wins support = No
```

Многие из этих параметров имеют значения, противоположные значениям параметров контроллера домена; при настройке этих параметров необходимо убедиться, что главным

браузером или сервером NetBIOS-имен будет являться только один компьютер. Параметр `password server` должен указывать на контроллер используемого домена. Для параметра `security` можно выбрать одно из двух значений: `Domain` (для полного присоединения к домену, как в нашем примере) или `Server` (в этом случае Samba не будет полностью присоединена к домену, но будет отправлять запросы на проверку паролей контроллеру домена с использованием более простых, не доменных протоколов). Если вы установили параметр `security = Domain`, то вы должны полностью присоединить компьютер к домену, выполнив следующую команду:

```
# net join member -U ntadmin
```

Вместо `ntadmin` укажите имя пользователя административной учетной записи. Когда вам будет предложено ввести пароль, введите пароль пользователя `ntadmin`. Если все пройдет успешно, то вы увидите сообщение о том, что компьютер был присоединен к домену, указанному в параметре `workgroup` файла `smb.conf`.

После того, как сервер Samba будет присоединен к домену, компьютер будет выполнять проверку подлинности (но только для Samba), обращаясь к контроллеру домена. В теме 313.3 (Winbind) рассматривается настройка Linux-компьютеров, позволяющая использовать контроллер домена для проверки подлинности не только связанных с Samba приложений.

### **Создание доменных общих ресурсов и пользовательских настроек**

После того, как мы проверили базовую функциональность подключения к домену, можно пойти дальше и создать специальный доменный общий ресурс под названием `NETLOGON`, а также сконфигурировать Samba таким образом, чтобы она сохраняла пользовательские настройки рабочего стола Windows на контроллере домена.

### **Создание общих ресурсов NETLOGON**

На общем ресурсе `NETLOGON` хранятся сценарии входа в систему. Эти сценарии выполняются каждый раз, когда пользователь подключается к домену. Таким образом, следует создавать и проверять работу этих сценариев в Windows, поскольку это *не* сценарии Linux.

Общий ресурс `NETLOGON` – это обычновенный общий файловый ресурс с соответствующим именем (создание общих файловых ресурсов рассматривается в теме 312.2). Обычно доступ на запись к ресурсу `NETLOGON` ограничивается и разрешается только определенным пользователям. Типовая конфигурация может выглядеть следующим образом:

```
[netlogon]
comment = Network Logon Service
path = /var/samba/netlogon
guest ok = No
read only = Yes
write list = abe
```

В этом примере был создан общий ресурс с доступом только на чтение, расположенный в директории `/var/samba/netlogon`. Параметр `write list` предоставляет пользователю `abe` право на запись в этот общий ресурс. Заметим, что пользователь `abe` должен иметь разрешения файловой системы Linux на запись в директорию `/var/samba/netlogon`; если разрешения Linux не позволяют пользователю `abe` записывать данные в эту директорию, то указывать его в параметре `write list` бесполезно.

Клиенты Windows должны знать о том, какие сценарии, хранящиеся на общем ресурсе NETLOGON, должны запускаться. Это решается с помощью глобального параметра **logon script**. В следующем простом примере просто указывается имя файла:

```
logon script = LOGON.BAT
```

В этом примере всем клиентам сообщается о том, что будет запускаться файл LOGON.BAT. Тем не менее, иногда для различных клиентов необходимо указывать различные сценарии входа. Например, можно использовать отдельный сценарий для каждой операционной системы. В этом случае конфигурация может выглядеть следующим образом:

```
logon script = LOGON-%a.BAT
```

В этом примере переменная **%a** принимает различные значения для различных клиентских операционных систем, как показано в таблице 1. Можно также использовать другие переменные (см. материалы цели 312.1). Если вы решите использовать предыдущий пример, то для клиентов под управлением Windows 7 будет запускаться сценарий входа *LOGON-Vista.BAT*, а для клиентов под управлением Windows XP – *LOGON-WinXP.BAT*. Заметьте, что не все операционные системы используют сценарии входа. Например, их не использует Linux, поэтому нет необходимости создавать сценарий *LOGON-CIFSFS.BAT*.

**Таблица 1. Значения переменной %a**

Значение	Клиентская операционная система
Samba	Samba
CIFSFS	Файловая система Linux Common Internet File System (CIFS)
OS2	IBM® Operating System/2® (OS/2)
WfWg	Windows for Workgroups
Win95	Windows 95, 98 или Me
WinNT	Windows NT
Win2K	Microsoft Windows 2000
WinXP	Windows XP
WinXP64	Windows XP 64 bit
Win2K3	Windows Server® 2003
Vista	Windows Vista или Windows 7
UNKNOWN	Все остальные клиенты

В дополнение к сетевым сценариям входа общий ресурс NETLOGON может содержать файл *системных политик*. Этот файл может автоматически вносить изменения в системный реестр Windows, обеспечивая его лучшую работу в домене. В операционных системах Windows 9x/Me файл политик называется *Config.POL* и создается на компьютере Windows с помощью инструмента *Policy Editor* (*Poledit.exe*). В операционных системах Windows NT, Windows 2000, Windows Server 2003, Windows XP, Windows Vista и Windows 7 этот файл называется *NTConfig.POL* и создается из группы **Start > Programs > Administrative Tools**. Рассмотрение процесса создания файлов политик выходит за рамки этой статьи, однако вам необходимо знать о возможной необходимости размещения этого файла на общем ресурсе NETLOGON.

### **Создание перемещаемых профилей**

Обычно операционная система Windows хранит информацию о настройках пользователей на локальном компьютере. Это удобно, если каждый пользователей всегда работает за одним компьютером, однако это не всегда так. Например, в компьютерном классе университета

каждый студент может работать на различных компьютерах. В этой ситуации лучше хранить настройки пользователей на удаленном сервере. Для этого как раз и предназначены *перемещаемые профили*.

Чтобы начать использовать перемещаемые профили для Windows NT, Windows 2000, Windows XP, Windows Vista и Windows 7, необходимо создать общий ресурс PROFILES. Этот общий ресурс обычно делается невидимым для пользователей, чтобы не путать их. Кроме того, доступ к файлам и директориям этого ресурса ограничивается внутри Samba для того, чтобы пользователи не могли получить доступ к чужим профилям. Несмотря на это, пользователям должен быть предоставлен доступ на чтение и запись в директорию Linux, используемую в качестве общего ресурса (в зависимости от групповых политик может потребоваться установить для этой директории флаги доступа 0777). Общий ресурс Samba для хранения перемещаемых профилей может выглядеть следующим образом:

```
[profiles]
comment = NT Profiles Share
directory = /var/samba/profiles
read only = No
create mode = 0600
directory mode = 0700
browseable = No
```

Традиционно этот общий ресурс создается на контроллере домена под управлением Samba, хотя можно размещать его на любом другом файловом сервере. Кроме того, при создании ресурса PROFILES необходимо сообщить о его расположении операционной системе Windows. Для этого можно использовать глобальный параметр **logon path**:

```
logon path = \\%L\PROFILES\%U
```

В этом примере в названии пути к директории с перемещаемыми профилями мы использовали переменные %L (NetBIOS-имя текущего сервера) и %U (имя пользователя, запускающего сеанс). Убедитесь, что перемещаемые профили действительно сохраняются на созданном общем ресурсе PROFILES.

### **Настройка резервного контроллера домена**

Домены Windows NT могут иметь несколько контроллеров домена для обеспечения избыточности в случае выхода из строя одного из них. В терминах домена Windows NT один из них является *первичным контроллером домена* (Primary Domain Controller, PDC), а остальные – *резервными контроллерами домена* (Backup Domain Controller, BDC). Такая конфигурация несколько усложняет общую картину, но является незаменимой в тех случаях, если работа сетевого окружения зависит от исправности контроллера домена.

Включение резервного контроллера домена в Samba лучше всего осуществлять с помощью протокола LDAP, в котором хранится информация об учетных записях. LDAP был разработан для обмена информацией между базами данных, необходимого для конфигураций такого типа, поэтому имеет смысл связать первичный и резервный контроллеры домена с главным и подчиненным серверами LDAP, соответственно. Серверы LDAP и серверы Samba могут работать как на одном компьютере, так и на разных. Можно также использовать один сервер LDAP для первичного и резервного контроллеров домена, хотя это не лучший способ, поскольку при этом частично теряется отказоустойчивость, в особенности, если сервер LDAP работает на том же компьютере, что и один из контроллеров домена.

Для простейшей настройки первичного контроллера домена с использованием сервера LDAP необходимо настроить несколько дополнительных параметров Samba, идентифицирующих

сервер LDAP. Минимальная конфигурация глобальных параметров выглядит следующим образом:

```
passdb backend = ldapsam://localhost:389
ldap suffix = dc=example,dc=org
ldap user suffix = ou=Users
ldap group suffix = ou=Groups
ldap machine suffix = ou=Computers
ldap idmap suffix = ou=Idmap
ldap admin dn = cn=ntadmin,dc=example,dc=org
```

В этом примере мы настраиваем Samba на использование сервера LDAP, запущенного на том же самом компьютере (`passdb backend = ldapsam://localhost:389`), а также определяем наиболее важные свойства LDAP, используемые для идентификации и администрирования учетных данных.

**Примечание.** Конфигурация LDAP сама по себе является непростой задачей. Предполагается, что ваши главный и подчиненный серверы LDAP к этому моменту уже настроены. Для получения дополнительной информации о LDAP обратитесь к разделу [Ресурсы](#).

Я рекомендую вам сначала настроить ваш первичный контроллер домена на использование главного сервера LDAP и провести всестороннюю проверку этой конфигурации. После этого можно приступать к настройке резервного контроллера домена, начав с выполнения нескольких предварительных действий:

1. Наберите на резервном контроллере домена команду `net rpc getsid`.

С помощью этой команды вы получите важный идентификатор, который должен совпадать на компьютерах первичного и резервного контроллеров домена.

2. Наберите на резервном контроллере домена команду `smbpasswd -w mypass`, где `mypass` – это пароль администратора LDAP.
3. Синхронизируйте локальные базы данных учетных записей на первичном и резервном контроллерах домена.

Это можно сделать по-разному, но простейший способ заключается в копировании файлов `/etc/passwd`, `/etc/group` и `/etc/shadow` с одного компьютера на другой. Если вы используете LDAP для управления учетными записями Linux, то это действие может оказаться необязательным.

4. Скопируйте общий ресурс NETLOGON с первичного контроллера домена на резервный.

Помимо параметров конфигурации общего ресурса NETLOGON из файла `smb.conf` не забудьте скопировать саму директорию. Следует регулярно синхронизировать содержимое этой директории, а также последние изменения.

После выполнения этих действий можно создавать конфигурацию резервного контроллера домена в файле `smb.conf`.

```
passdb backend= ldapsam:ldap://slave-ldap.example.org
domain master = No
domain logons = Yes
os level = 64
ldap suffix = dc=example,dc=org
ldap user suffix = ou=Users
ldap group suffix = ou=Groups
ldap machine suffix = ou=Computers
```

```
ldap idmap suffix = ou=Idmap
ldap admin dn = cn=ntadmin,dc=example,dc=org
idmap backend = ldap:ldap://master-ldap.example.org
idmap uid = 10000-20000
idmap gid = 10000-20000
```

Эти параметры заставляют резервный контроллер домена отказаться от участия в выборах главного обозревателя (и позволить выиграть выборы первичному контроллеру), использовать подчиненный сервер LDAP (`slave-ldap.example.org`) в качестве базы данных паролей и использовать главный сервер LDAP (`master-ldap.example.org`) для хранения сопоставлений между учетными записями Linux и Windows.

## Настройка доверительных отношений между доменами

### Основы доменов Active Directory

Домены, о которых рассказывается в этой статье – это домены Windows NT, которые поддерживает Samba 3.x. Однако начиная с Windows 2000, Microsoft перешла к новой архитектуре доменов, известной как *домены Active Directory®*. Samba 3.x может быть присоединена к домену Active Directory в качестве члена-сервера, но ее возможности работы в качестве полноценного контроллера домена Active Directory в лучшем случае ограничены. В Samba 4.x поддержка Active Directory значительно улучшена, но в текущий момент (по состоянию на март 2011 г.) эта версия Samba находится в стадии альфа-тестирования.

Active Directory, по существу, объединяет три различных стека сетевых протоколов: Server Message Block (SMB)/CIFS, LDAP и Kerberos. SMB/CIFS отвечает за возможность совместного использования файлов и принтеров, LDAP позволяет хранить информацию об учетных записях, а Kerberos обеспечивает шифрование. Все эти технологии доступны и в Linux, но их интеграция может оказаться сложной задачей. Как уже упоминалось, можно настроить совместную работу Samba и LDAP для управления учетными записями – этот подход настоятельно рекомендуется использовать, если вы планируете использовать резервный контроллер домена. Samba 3.x также можно увязать с Kerberos, хотя мы не рассматриваем здесь эти вопросы. Samba 4 частично реализует собственные функции LDAP и Kerberos, совмещая их в главном пакете дистрибутива.

Более подробно интеграция Samba с Active Directory рассматривается в теме 314.3.

В крупных организациях различные подразделения могут иметь свои собственные домены Windows NT. Эта конфигурация позволяет каждому подразделению управлять собственными учетными записями, но может ограничить возможности по использованию ресурсов других подразделений. Например, в двух соседних подразделениях может возникнуть необходимость в использовании общих ресурсов печати. Такой тип междоменного представления ресурсов может быть реализован посредством создания *доверительных отношений между доменами*, или для краткости просто *доверительных отношений*.

Доверительные отношения обладают следующими двумя особенностями, о которых необходимо помнить. Во-первых, эти отношения *не являются транзитивными*, т. е. действуют *только* для тех доменов, для которых они были явно настроены. Например, если домен PHYSICS доверяет домену GEOLOGY, а домен GEOLOGY – домену BIOLOGY, то домен PHYSICS не будет автоматически доверять домену BIOLOGY. Во-вторых, доверительные отношения являются односторонними. Например, если домен PHYSICS доверяет домену GEOLOGY, то это не означает, что GEOLOGY доверяет домену PHYSICS. Для создания двусторонних доверительных отношений необходимо настроить доверительные отношения с соседом в каждом домене.

Для настройки доверительных отношений между доменами каждый домен должен быть отдельно настроен соответствующим образом. Первый домен является *доверяющим доменом*, и его ресурсы будут доступны пользователям другого домена, который в этом случае будет являться *доверяемым доменом*. Предположим, например, что в домене PHYSICS есть широкоформатный принтер, который хотят использовать пользователи домена GEOLOGY. В этом случае домен PHYSICS будет являться доверяющим доменом, а домен GEOLOGY – доверяемым. Чтобы приступить к настройке доверительных отношений, сначала необходимо создать на доверяющем домене (PHYSICS) учетные записи Linux и Samba для доверяемого домена (GEOLOGY):

```
# useradd -d /dev/null -M -g trust -s /bin/false geology$  
# smbpasswd -a -i geology
```

Обратите внимание на то, что вы должны использовать символ доллара (\$) при создании учетной записи Linux с помощью `useradd`, но не при создании учетной записи Samba с помощью `smbpasswd`. Команда `smbpasswd` попросит вас ввести пароль для учетной записи доверительных отношений. Запомните этот пароль, поскольку вскоре он вам потребуется. Контроллер доверяемого домена GEOLOGY будет использовать эту учетную запись так, как если бы он являлся обычным членом доверяющего домена, позволяя пользователям домена GEOLOGY получать доступ к ресурсам домена PHYSICS.

После того, как учетная запись доверительных отношений создана, можно настраивать доверяемый контроллер домена на ее использование:

```
# net rpc trustdom establish physics
```

Выполните эту команду на компьютере Linux, который является членом доверяемого домена (в нашем случае GEOLOGY). Когда надо будет ввести пароль, то введите тот пароль, который был указан в команде `smbpasswd` для доверяющего домена. Если все прошло успешно, то пользователи домена GEOLOGY смогут теперь получить доступ к серверам домена PHYSICS. Если вам необходимо настроить двусторонние доверительные отношения, необходимо повторить эти действия во втором домене, поменяв роли доменов местами.

Для отмены доверительных отношений выполните в доверяемом домене следующую команду:

```
# net rpc trustdom revoke physics
```

Вместо предыдущей команды можно использовать следующую команду в доверяющем домене:

```
# net rpc trustdom del geology
```

## Что дальше

Следующая статья этой серии содержит материалы цели 312.5 темы 312. В ней рассматривается инструмент Samba Web Administration Tool, который позволяет конфигурировать Samba с помощью Web-интерфейса.

## Ресурсы

## Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Domain control \(EN\)](#).
- В статье [Аутентификация Samba на основе LDAP \(EN\)](#) (Кейт Робертсон, developerWorks, январь 2006 г.) более подробно рассказывается о том, как настроить Samba и LDAP.
- В главе [Backup Domain Control chapter \(EN\)](#) официального руководства Samba представлена более подробная информация о резервном контроллере домена.
- Глава [Samba net command documentation \(EN\)](#) официального руководства Samba содержит информацию по управлению доверительными отношениями между доменами.
- На Web-сайте [программы сертификации LPIC \(EN\)](#) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302 \(EN\)](#), а также [примеры заданий и вопросов \(EN\)](#).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI \(EN\)](#) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.

## Получить продукты и технологии

- Загрузите [файл Win7\\_Samba3DomainMember.reg \(EN\)](#), который требуется для присоединения компьютера под управлением Windows 7 к домену Samba.

# Изучаем Linux, 302 (смешанные среды): Конфигурирование SWAT

Управление Samba с помощью Web-интерфейса

[Родерик Смит \(Roderick Smith\)](#), автор и консультант, IBM

**Описание:** Samba Web Administration Tool (SWAT) – это графический Web-инструмент для администрирования Samba, который можно запустить на любом компьютере, на котором установлен Web-браузер. Поскольку SWAT является самостоятельным сервером, то, как и все серверы, требует выполнения определенных начальных настроек конфигурации. Из этой статьи вы узнаете о том, как установить, настроить и использовать SWAT для управления Samba.

**Дата:** 14.06.2012

**Уровень сложности:** средний

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к

нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

## Краткий обзор

В этой статье рассматриваются следующие темы:

- Обзор возможностей Samba Web Administration Tool (SWAT).
- Инсталляция и настройка SWAT.
- Конфигурирование Samba с помощью SWAT.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 312.5 темы 312. Цель имеет вес 4.

## Предварительные требования

Чтобы извлечь наибольшую пользу из этой статьи, необходимо обладать практическими навыками работы с инструментами командной строки Linux и знать основы конфигурирования Samba. Вы должны знать общую структуру конфигурационного файла smb.conf и иметь представление о работе серверов, включая использование сценариев запуска System V (SysV) и супер-серверов.

## Установка и запуск SWAT

В Samba версии 3.x SWAT является отдельным самостоятельным сервером (в дополнение к двум основным серверам Samba – `smbd` и `nmbsd`), поэтому его необходимо инсталлировать и запускать отдельно от Samba. Также может потребоваться отдельная инсталляция SWAT в зависимости от того, каким способом был установлен сервер Samba. Для обеспечения дополнительной защиты можно настроить SWAT на использование SSL-шифрования.

**Примечание.** Версия Samba 4, до сих пор находящаяся в разработке, содержит существенные изменения в архитектуре, затрагивающие, в том числе, и SWAT. В этой статье рассматривается SWAT для версии Samba 3.x (на сегодняшний день рекомендуется использовать в рабочих окружениях именно эту версию).

## Инсталляция SWAT

Если сервер Samba был создан из исходного кода, как описано в цели 311.1, то SWAT уже должен быть скомпилирован и установлен вместе с компонентами Samba. Запретить компиляцию SWAT позволяет опция `--disable-swat` команды `configure`. Если же вы хотите явно разрешить компиляцию SWAT, то можно не полагаться на значения по умолчанию, а указать команде `configure` опцию `--enable-swat`.

Если сервер Samba был инсталлирован из пакета двоичного кода, разработанного для вашего дистрибутива, то просмотрите список пакетов, чтобы узнать, как можно установить SWAT. Во многих дистрибутивах сервер SWAT содержится в пакете с именем `swat` или `samba-swat`. Этот пакет необходимо инсталлировать отдельно от основных пакетов Samba, хотя, он может инсталлироваться и автоматически. Используйте инструменты для работы с пакетами, чтобы узнать, какие пакеты установлены на вашем компьютере.

Если вы не можете найти отдельный пакет SWAT, то, возможно, он был установлен как часть другого пакета Samba. Двоичный файл SWAT называется, как ни странно, `swat`, поэтому вы можете поискать его в вашей файловой системе и, таким образом, определить, установлен ли SWAT на вашем компьютере. При использовании дистрибутива Gentoo Linux для извлечения пакета `samba` для сборки SWAT значение `swat` должно быть задано в качестве флага `USE`.

После того, как SWAT инсталлирован, его можно запускать. Работа SWAT осуществляется через супер-сервер, например, `xinetd` или `inetd`. Поскольку конфигурации этих двух супер-серверов существенно отличаются, то мы рассмотрим их отдельно.

## Запуск SWAT через xinetd

На сегодняшний день в большинстве дистрибутивов Linux в качестве супер-сервера используется **xinetd**. Если вы не знаете точно, какой супер-сервер работает на вашем компьютере, то попробуйте поискать строку **inetd** с помощью команды **ps**.

```
$ ps ax | grep inetd
17996 ? Ss 0:00 /usr/sbin/xinetd -pidfile /var/run/xinetd.pid -stayalive -reuse
```

В этом примере видно, что на компьютере работает **xinetd**. Если в выводе команды **ps** есть процесс **inetd**, то следуйте инструкциям из следующего раздела [Запуск SWAT через inetd](#). Если ни один из этих процессов не запущен, то, возможно, потребуется установить **xinetd** или **inetd** самостоятельно.

Сервер **xinetd** использует конфигурационный файл с именем **/etc/xinetd.conf**. В большинстве дистрибутивов этот файл содержит глобальные параметры, а файлы с настройками, относящимися к различным серверам, расположены в директории **/etc/xinetd.d**. Конфигурационный файл SWAT, скорее всего, будет называться **/etc/xinetd.d/swat**, а его содержимое должно быть похоже на содержимое листинга 1.

### Листинг 1. Пример конфигурационного файла SWAT

```
service swat
{
    port          = 901
    socket_type   = stream
    protocol      = tcp
    wait          = no
    only_from     = localhost
    user          = root
    server        = /usr/sbin/swat
    log_on_failure += USERID
    disable       = yes
}
```

Ваша конфигурация может иметь некоторые отличия, но основные параметры должны быть такими же. Если этот файл отсутствует в вашей системе, то можно либо создать его вручную, либо добавить в файл **/etc/xinetd.conf** запись, описывающую службу SWAT. Значения для большинства параметров лучше не изменять (либо оставить их, как есть, в конфигурационном файле вашего дистрибутива, либо задать в соответствии с предыдущим примером), однако может потребоваться настроить некоторые из них:

- **only\_from** – ограничивает доступ. В нашем примере **xinetd** разрешает доступ только с компьютера, на котором запущен SWAT. Эта мера защиты является достаточно эффективной, однако если необходимо управлять сервером Samba с другого компьютера (а, возможно, с нескольких), то можно указать в этой строке различные адреса. Можно указывать адреса различных типов; чтобы узнать подробности, обратитесь к [ман-странице xinetd.conf](#).
- **server** – содержит полный путь к двоичному файлу SWAT. Убедитесь, что файл **swat** присутствует в указанной директории.
- **disable = yes** – указывает **xinetd** на то, что SWAT *не* должен запускаться. Для использования SWAT необходимо изменить значение этого параметра на **disable = no**.

Последняя строка очень важна: в целях безопасности во многих дистрибутивах запуск SWAT запрещен по умолчанию в файле /etc/xinetd.d/swat. Кроме того, серверы могут быть запрещены в файле /etc/xinetd.conf:

```
disabled = swat
```

Если вы обнаружите такую строку, содержащую слово **swat**, то необходимо удалить ее, чтобы SWAT мог запускаться.

После настройки всех необходимых параметров в файле /etc/xinetd.d/swat необходимо, чтобы **xinetd** перезагрузил свой конфигурационный файл. В большинстве дистрибутивов это можно сделать, передав опцию **reload** в сценарий запуска SysV сервера **xinetd**:

```
# /etc/init.d/xinetd reload
```

После выполнения этой команды можно начинать использовать SWAT. Если вы хотите включить безопасность SSL, то обратитесь к разделу [Включение SSL-шифрования](#).

### Запуск SWAT через inetd

Если вы используете **inetd**, то процесс настройки будет похож на процесс настройки **xinetd**, но с некоторыми существенными отличиями. Как вы могли догадаться, конфигурационный файл **inetd** называется */etc/inetd.conf*. Последние версии **inetd** поддерживают разделение конфигурации на несколько файлов, каждый из которых относится к определенному серверу (как и в случае с **xinetd**), поэтому в директории */etc/inetd.d* вы можете найти несколько отдельных конфигурационных файлов.

При использовании как файла */etc/inetd.conf*, так и файла */etc/inetd.d/samba* конфигурация Samba состоит из одной строки:

```
swat stream tcp nowait root /usr/sbin/tcpd /usr/sbin/swat
```

Эта строка содержит почти ту же самую информацию, что и строка конфигурации **xinetd**. Номер порта можно определить, найдя имя (**swat**) в начале строки в файле */etc/services*. Фрагмент */usr/sbin/tcpd* в этом примере не имеет аналога для конфигурации **xinetd**, поскольку он указывает на программу, которую запускает **inetd** при попытках обращений клиента к серверу. В этом примере **inetd** запускает TCP Wrappers (используя для этого имя программы */usr/sbin/tcpd*). TCP Wrappers, в свою очередь, выполняют собственные проверки безопасности и запускают SWAT, т. е. программу, указанную в последнем фрагменте строки (*/usr/sbin/swat*).

В файле */etc/inetd.conf* строка для SWAT может быть закомментирована с помощью символа решетки (#). В этом случае необходимо раскомментировать эту строку для включения SWAT. После раскомментирования этой строки и соответствующей настройки конфигурации необходимо, чтобы **inetd** перезагрузил свой конфигурационный файл. Это делается точно так же, как и в случае с **xinetd**, который был рассмотрен в предыдущем разделе.

### Включение SSL-шифрования

Как только что было сказано, стандартная конфигурация SWAT не поддерживает шифрование. В этом нет ничего страшного, если вы разрешили доступ только с локального компьютера (как было показано в конфигурации для **xinetd**) или наложили подобные ограничения с помощью TCP Wrappers. Однако если вы планируете работать с SWAT удаленно, то отсутствие шифрования означает, что пароль пользователя root будет

передаваться по сети в незашифрованном виде. Это создает большой риск безопасности, особенно если необходимо подключаться к SWAT с удаленного компьютера через Интернет, а не через локальную сеть, в которой работает сервер Samba.

Чтобы повысить уровень безопасности, можно настроить в конфигурации SWAT поддержку SSL. Для этого необходимо установить дополнительное программное обеспечение OpenSSL и **stunnel** и внести изменения в конфигурацию. Во многих дистрибутивах доступны оба этих пакета, и их инсталляция не должна вызывать трудностей.

**Примечание.** Лучше всего сначала настроить и протестировать SWAT *без* использования SSL-шифрования. Таким образом, вы будете точно знать, что сервер работает в соответствии с минимальными настройками. Если при этом возникают проблемы, то они не связаны с SSL. После устранения таких проблем можно приступать к настройке шифрования.

После того, как вы инсталлировали программы OpenSSL и **stunnel**, выполните следующие действия:

1. При необходимости создайте пользователя и группу для **stunnel**.

Возможно, эти действия будут выполнены автоматически при установке пакета **stunnel**, и этот шаг не потребуется.

2. Сгенерируйте SSL-сертификат и закрытый ключ, выполнив следующую команду (в одной строке) от имени пользователя root:

```
# openssl req -new -x509 -days 365 -nodes -out  
    /etc/stunnel/stunnel.pem -keyout /etc/stunnel/stunnel.pem
```

В процессе выполнения этой команды вы должны будете предоставить различную информацию (например, имя компьютера и адрес электронной почты). Помните, что сертификат, создаваемый этой командой, действует в течение 365 дней, поэтому вам придется повторять эту процедуру каждый год.

3. Измените разрешения для файла `/etc/stunnel/stunnel.pem`, сделав его владельцами пользователя и группу **stunnel**, созданные на шаге 1.
4. Создайте или измените файл `/etc/stunnel/stunnel.conf`.

В листинге 2 приведен пример конфигурации. Если в вашей системе уже есть этот файл, то оставьте его параметры без изменений, но убедитесь, что в конце присутствуют три строки, показанные в листинге 2 (три последние строки, начиная со строки `[swat]`). Эти строки указывают **stunnel**, как нужно обрабатывать подключения к SWAT: в частности, **stunnel** прослушивает порт 901 и передает дешифруемый трафик на порт 902. Также убедитесь в том, что в строках `cert` и `key` указан сертификат, который был сгенерирован на шаге 2.

## Листинг 2. Для утилиты **stunnel** требуется собственный конфигурационный файл

```
chroot  = /var/lib/stunnel/  
pid     = /stunnel.pid  
setuid  = stunnel  
setgid  = stunnel  
  
debug   = 7  
output  = /var/log/messages  
  
client  = no  
cert    = /etc/stunnel/stunnel.pem
```

```

key      = /etc/stunnel/stunnel.pem

# Accept SSL connections on port 901 and funnel it to
# port 902 for swat.
[swat]
accept   = 901
connect  = 902

```

5. Создайте новую запись для SWAT в файле конфигурации `xinetd` или `inetd`.

Эта конфигурация должна выглядеть как обычная конфигурация без поддержки шифрования, за исключением того, что должен прослушиваться порт 902, а вместо имени `swat` должно использоваться имя `swat-stunnel`. Подключения должны быть разрешены только с интерфейса локального компьютера (`localhost`).

6. Удалите или закомментируйте исходную конфигурацию SWAT в файле `xinetd` или `inetd`.

7. Отредактируйте файл `/etc/services`:

- Скопируйте строку `swat`.
- Переименуйте скопированную строку в `swat-stunnel`.
- Измените номер порта на 902.

8. В некоторых дистрибутивах для запуска `stunnel` может потребоваться отредактировать файл `/etc/default/stunnel4`. В частности, необходимо изменить значение параметра `ENABLED` с 0 на 1.

9. Перезапустите супер-сервер.

10. Запустите `stunnel`.

Для однократного запуска `stunnel` можно выполнить команду `/etc/init.d/stunnel start` от имени пользователя `root`, а для регулярного запуска может потребоваться изменить конфигурацию `SysV`.

После того, как вы внесли все изменения, вы должны получить возможность подключаться к SWAT с использованием шифрованного соединения, используя вместо протокола `http://` протокол `https://`. Обратите внимание на то, что при первом подключении браузер может сообщить о том, что сертификат не является доверенным, поскольку вы генерировали его сами. Разрешите использование этого сертификата.

## Использование SWAT

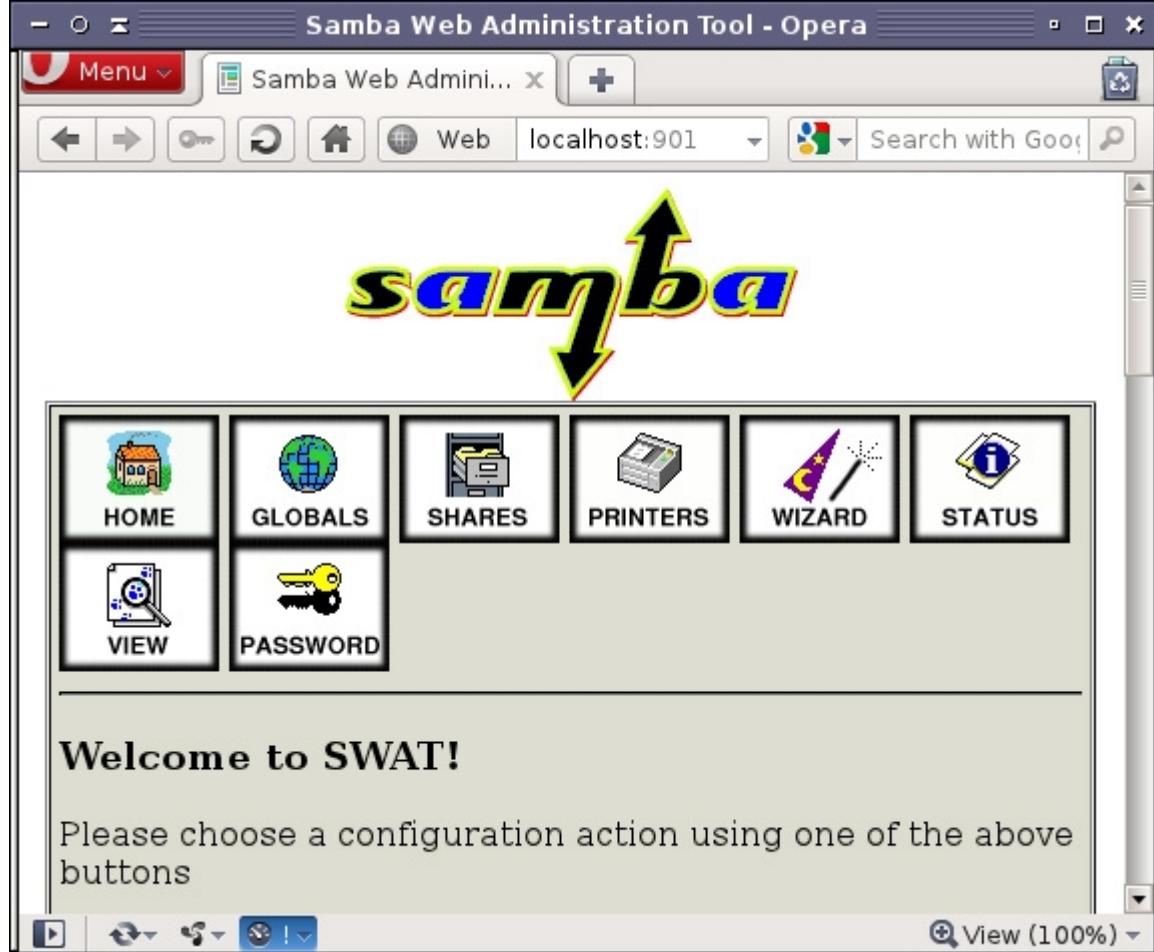
После того, как SWAT сконфигурирован, можно приступить к его использованию. Подключиться к SWAT можно через любой Web-браузер. Подключившись к SWAT, можно управлять сервером Samba.

## Подключение к SWAT

В любом браузере, в котором вы привыкли работать, наберите URL-адрес `http://localhost:901`; если вы подключаетесь к SWAT с удаленного компьютера, то вместо `localhost` укажите имя сервера, на котором запущен SWAT. Если вы настроили SSL-шифрование, то замените префикс `http://` на `https://` в начале URL-адреса. Если SWAT разрешено принимать учетные данные с других удаленных компьютеров, то для подключения к SWAT можно использовать Web-браузер любой операционной системы – Linux, Mac OS X, Windows и других. Можно даже использовать браузер мобильного телефона.

При первом подключении к серверу браузер запросит у вас имя пользователя и пароль. Введите в качестве имени пользователя `root` и пароль для этой учетной записи. После этого вы попадете на главную страницу SWAT, как показано на рисунке 1.

**Рисунок 1. Главная страница SWAT, позволяющая выполнять различные действия**



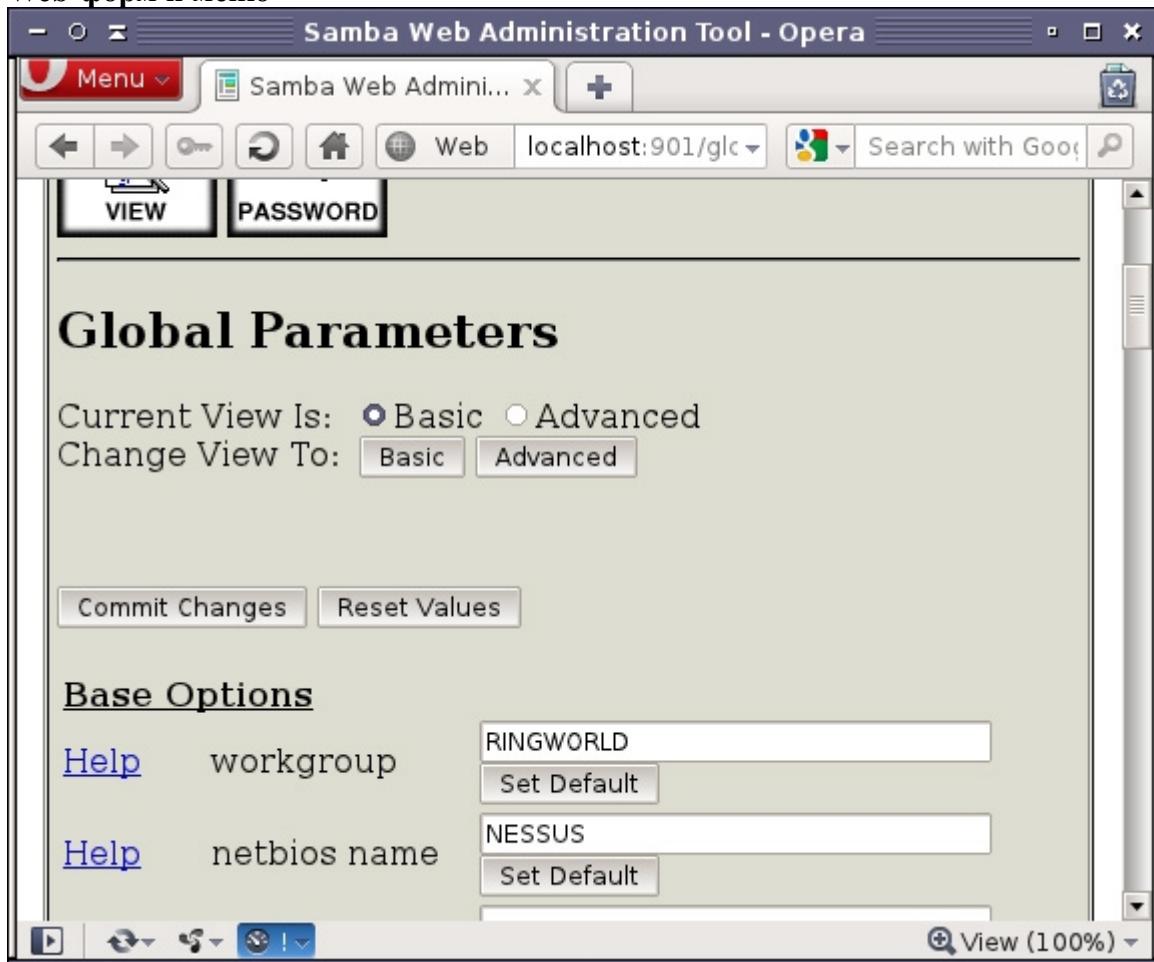
Кнопки, расположенные под логотипом Samba, настраивают различные параметры – раздел [global], общие файловые ресурсы, общие ресурсы печати и т. д. Как правило, если вы знакомы с конфигурационным файлом Samba smb.conf, то сможете легко настроить Samba с помощью SWAT.

Если прокрутить домашнюю страницу SWAT, то вы увидите ссылки на документацию Samba. Эта документация состоит из man-страниц различных демонов Samba, конфигурационных файлов, утилит и т. д.

### Использование возможностей SWAT

Рассмотрим пример использования SWAT. Нажмите кнопку **GLOBALS**. На обновившейся странице вы увидите доступные для настройки параметры раздела [global] файла smb.conf, как показано на рисунке 2. Нажмите кнопку **Advanced**, если вы хотите получить доступ к дополнительным параметрам; по умолчанию (в режиме Basic) отображаются только общие параметры.

**Рисунок 2.** SWAT позволяет настраивать параметры Samba с помощью различных Web-форм и меню



Внимательно ознакомьтесь с параметрами, содержащимися на этой странице, и попробуйте изменить некоторые из них. Чтобы внести изменения в файл smb.conf и перезагрузить конфигурацию, необходимо нажать кнопку **Commit Changes**.

Если нажать кнопку **SHARES** или **PRINTERS**, то можно редактировать общие файловые ресурсы и ресурсы печати, соответственно. Интерфейс идентичен интерфейсу раздела **[globals]** за исключением того, что нужно выбрать из выпадающего списка настраиваемый общий ресурс и нажать кнопку **Choose Share**. Можно также создать новый общий ресурс, указав его имя в текстовом поле напротив кнопки **Create Share** и нажав ее.

Дополнительные кнопки позволяют выполнять следующие действия:

- Кнопка **WIZARD** открывает страницу, на которой вы можете быстро выполнить некоторые общие настройки, например, настроить конфигурацию домена.
- Кнопка **STATUS** показывает текущий статус сервера Samba, например, сколько активных подключений обрабатывается, и к каким файлам происходят обращения. На этой же странице можно запустить, остановить или перезапустить серверы **smbd**, **nmbd** и **winbindd**.
- Кнопка **VIEW** показывает текущий конфигурационный файл smb.conf.
- Кнопка **PASSWORD** позволяет управлять пользователями – изменять пароли, добавлять и удалять пользователей, и т. д. Можно управлять как локальными пользователями, так и пользователями другого сервера Server Message Block (SMB)/Common Internet File System (CIFS).

Большинство возможностей SWAT доступны только пользователю root. Обычные

пользователи могут просматривать страницы **HOME**, **STATUS**, **VIEW** и **PASSWORD**, однако некоторые параметры на этих страницах им недоступны. Вероятно, обычным пользователям будет доступна страница **PASSWORD**, поскольку они могут использовать ее для смены собственных паролей.

## Предостережения относительно использования SWAT

SWAT – это полезный инструмент, тем не менее, он имеет определенные ограничения и недостатки. Основное и самое серьезное – это отсутствие в SWAT поддержки директивы `include`, которая используется в файле smb.conf для разделения конфигурации на несколько отдельных файлов (можно даже иметь отдельные конфигурационные файлы для различных клиентов). Если вы используете такие конфигурации, то SWAT окажется для вас бесполезен, по крайней мере, для настройки конфигурации Samba (тем не менее, пользователи могут использовать SWAT для смены своих паролей).

Другая основная проблема SWAT связана с безопасностью. Можно свести риск к минимуму, разрешив подключения только с локального компьютера или используя SSL-шифрование при удаленных подключениях, как это было описано выше. Но даже с учетом этих мер, *любой* сервер является потенциальной угрозой безопасности: программное обеспечение может содержать скрытые ошибки или быть неправильно настроено. Необходимо учитывать эти риски и принимать соответствующие меры предосторожности, например, использовать TCP Wrappers, параметры `xinetd` или правила брандмауэра `iptables` для ограничения доступа к компьютеру. Чем больше пользователей могут получить доступ к серверу из Интернета, тем более подвержена риску его конфигурация.

## Что дальше

Следующая статья этой серии содержит материалы цели 312.6 темы 312. В ней рассматриваются вопросы локализации Samba, включая использование кодовых страниц и других возможностей для поддержки сервером Samba файлов, в названиях которых используются языки, отличные от английского.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): SWAT configuration](#) (EN).
- В [официальной документации Samba SWAT](#) (EN) вы найдете дополнительную информацию о настройке и использовании SWAT.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.

## Получить продукты и технологии

- [Web-сайт stunnel](#) (EN) – загрузите программу stunnel и документацию по ней.

- [Web-сайт OpenSSL](#) (EN) – загрузите OpenSSL и документацию по этому программному обеспечению.

# Изучаем Linux, 302 (смешанные среды): Локализация

*Базовые основы локализации Samba в неанглоязычной среде*

[Трейси Бост](#), консультант и преподаватель, Свободный писатель

**Описание:** Если вы работаете в смешанной среде, в которой используются символы, отличные от английских, то необходимо иметь представление о символьных кодах и кодовых страницах языка используемой среды. Также необходимо понимать, что в средах Linux и Windows пространства имен интерпретируются по-разному. Хотя Samba поддерживает локализацию, при работе со старыми клиентами Windows или Samba версий 2.x, а также в случаях использования наборов символов, отличных от Unicode, необходимо выполнить дополнительные настройки. В зависимости от используемого языка операционной системы может возникнуть необходимость в компоновке и применении исправлений для библиотек перекодировки. Из этой статьи вы узнаете о том, как работать с локализацией в среде Linux.

**Дата:** 28.06.2012

**Уровень сложности:** средний

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

## Краткий обзор

В этой статье рассматриваются следующие темы:

- Символьные коды и кодовые страницы.
- Работа Windows-клиентов с наборами символов.
- Библиотеки перекодировки.
- Настройка локализации Samba.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 312.6 темы 312. Цель имеет вес 1.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и в общих чертах понимает назначение Samba (о чем рассказывалось в предыдущей статье "[Изучаем Linux, 302: основные принципы](#)"). Для выполнения примеров этой статьи на вашем компьютере должно быть инсталлировано программное обеспечение Samba. Ваш компьютер должен быть подключен к локальной сети

и Интернету, и на нем должны быть проинсталлированы библиотеки компилятора GNU Compiler Collection. Для проверки работы с языком, отличного от английского, полезно иметь в сети клиента под управлением Windows.

### Зачем нужна локализация

Весьма вероятно, что при работе с файлами и директориями в смешанной среде пользователи будут использовать собственные локальные настройки. *Локальные настройки* – это просто набор параметров, определяющих язык, страну и другие установки, которые пользователь может использовать в программной среде. Если программное обеспечение поддерживает работу с локальными настройками пользователя, то обычно это называется *локализацией*.

### Символьные коды

Представьте, что вы просматриваете на вашем компьютере директории с помощью файлового менеджера Nautilus в Linux или с помощью Windows Explorer в Windows и встретили директорию с именем *01100001 01110000 01110000 01101100 01101001 01100011 01100001 01110100 01101001 01101111 01101110 01110011*. Или, предположим, компьютер отображает имя директории в виде *97 112 112 108 105 99 97 116 105 111 110 115* или *61 70 70 6C 69 63 61 74 69 6F 6E 73*. Если вы не умеете читать двоичный, десятичный или шестнадцатеричный код, или у вас под рукой нет программы-переводчика, то вы никогда не узнаете, что эта директория является общим файловым ресурсом с именем *applications*. Тем не менее, компьютер умеет понимать числа. Более того, числа – это единственное, что умеет понимать компьютер.

К счастью, вам не нужно изучать двоичную, десятичную, шестнадцатеричную или другие системы счисления для того, чтобы пользоваться компьютером, поскольку программы-трансляторы отображают уже преобразованные читаемые символы. В основе этих преобразований лежат таблицы символов. *Таблица символов* – это представление определенного символа в числовом формате, причем каждому символу сопоставлено определенное числовое значение. В таблице 1 представлена таблица символов ASCII (American Standard Code for Information Interchange), содержащая символы из названия директории нашего примера.

**Таблица 1. Символьные коды ASCII для имени директории "applications"**

Двоичное значение	Десятичное значение	Шестнадцатеричное значение	Отображаемый символ
01100001	97	61	а
01110000	112	70	р
01110000	112	70	р
01101100	108	6C	l
01101001	105	69	и
01100011	99	63	с
01100001	97	61	а
01110100	116	74	т
01101001	105	69	и
01101111	111	6F	о
01101110	110	6E	н
01110011	115	73	с

Этот пример будет полезен, если ваши локальные настройки работают с таблицей ASCII. Тем не менее, многие пользователи предпочитают использовать *собственные локальные настройки* для выбранной локализации.

## Стандарт Unicode

Если вы работаете в современной операционной системе и используете новое программное обеспечение, то, вероятно, вам приходилось использовать Unicode, даже если вы не знакомы с этим стандартом. В наши дни редко можно встретить статью о локализации, в которой бы не упоминалось о Unicode. В настоящее время Unicode является стандартом кодировки, де-факто использующимся для локализации символов. Он разработан для замены различных таблиц кодировки на всех уровнях путем кодировки абстрактных символов всех известных языков:

- Unicode используется по умолчанию в большинстве дистрибутивов Linux.
- Unicode используется по умолчанию в Samba версии 3.x.
- Unicode (UTF-16) используется по умолчанию на компьютерах Windows с конца 1990x.

UTF-8 – это наиболее широко используемая кодировка Unicode. В этой кодировке для каждого ASCII-символа используется один байт, что позволяет использовать те же коды символов, что и в ASCII. Тем не менее, для поддержки обратной совместимости системный администратор Linux должен уметь работать с различными кодовыми страницами, поскольку Unicode не всегда является лучшим или возможным решением для определенной среды, в которой используются символы, отличные от английских.

Вспомните те времена, когда компьютерные сети только начали появляться. Большинство программных продуктов было разработано с учетом только английского языка. По-существу, компьютеры без проблем могли отображать английские символы из стандартной ASCII-кодировки. В стандарте ASCII каждому английскому символу назначено десятичное число от 0 до 127, занимающее один байт. По мере возникновения необходимости использовать большее число символов (например, символы, которые содержатся в французском или испанском языках, или математических уравнениях) в таблицу ASCII был добавлен дополнительный бит, позволивший расширить ее дополнительными 128 символами, которым были присвоены десятичные числа от 128 до 255. Распространенными расширениями стандарта ASCII являются такие стандарты, как ISO Latin I, Extended Binary-Coded Decimal Interchange Code (EBCDIC, используется IBM) и Extended ASCII (используется Microsoft и в операционных системах DOS).

Но что, если пользователь работает в среде с использованием китайского, японского, венгерского, словацкого или другого языка, для которого символов из таблицы ASCII недостаточно? В работе с такими локальными настройками могут помочь различные кодовые страницы.

## Кодовые страницы

*Кодовая страница* – это сопоставление чисел и определенных символов в соответствии с тем набором символов, которые предполагается использовать в одной или нескольких локальных настройках. Традиционно кодовые страницы называют также *кодированием, набором символов и набором кодированных символов*. Хотя технически различные термины могут слегка отличаться, в этой статье термины *кодовая страница, набор символов и кодирование* тождественны.

Китайский, японский, словацкий и многие другие языки имеют собственные кодовые страницы. В таблице 2 перечислены некоторые наиболее распространенные кодовые страницы.

**Таблица 2. Распространенные кодовые страницы**

Кодовая страница	Представление
850	MS-DOS латинская 1 (Западноевропейская)

437	DOS-US, OEM-US
932	MS-DOS японская кодировка Shift-JIS
852	Центральноевропейские языки, использующие латинский шрифт
1252	Западноевропейские языки для Windows
950	MS-DOS традиционный китайский
65001	UTF-8 (Unicode)
28591	ISO-8859-1

## Работа с пространствами имен в среде с языком, отличным от английского

Поскольку Samba версии 2.x не имела поддержки Unicode, то вся поддержка наборов символов для различных языков заключалась в использовании кодовой страницы определенных локальных настроек. Старые Windows-клиенты используют однобайтовые кодовые страницы (в отличие от многобайтовых). Однако протокол Server Message Block (SMB)/Linux Common Internet File System (CIFS) не поддерживает перекодировку. Таким образом, при взаимодействии Samba и старых Windows-клиентов следует использовать одинаковые кодовые страницы.

Если в вашей среде необходимо использовать определенную кодовую страницу, то вы должны знать, что означают некоторые термины, использующиеся в Samba:

- **Кодировка UNIX** – кодировка, использующаяся внутри Linux.
- **Кодировка DOS** – кодировка, используемая Samba при взаимодействии со старыми Windows-клиентами.
- **Экранная кодировка** – кодировка, используемая для вывода информации на экран.

Если на вашем компьютере с Linux установлена библиотека `iconv` (скорее всего, это так), то можно посмотреть доступные кодовые страницы, выполнив команду `iconv -l`, как показано в листинге 1.

### Листинг 1. Часть списка доступных кодовых страниц

```
[tbost@samba ~]# iconv -l
The following list contain all the coded character sets known. This does
not necessarily mean that all combinations of these names can be used for
the FROM and TO command line parameters. One coded character set can be
listed with several different names (aliases).
```

```
437, 500, 500V1, 850, 851, 852, 855, 856, 857, 860, 861, 862, 863, 864, 865,
866, 866NAV, 869, 874, 904, 1026, 1046, 1047, 8859_1, 8859_2, 8859_3, 8859_4,
8859_5, 8859_6, 8859_7, 8859_8, 8859_9, 10646-1:1993, 10646-1:1993/UCS4,
ANSI_X3.4-1968, ANSI_X3.4-1986, ANSI_X3.4, ANSI_X3.110-1983, ANSI_X3.110,
ARABIC, ARABIC7, ARMSCII-8, ASCII, ASMO-708, ASMO_449, BALTIC, BIG-5,
BIG-FIVE, BIG5-HKSCS, BIG5, BIG5HKSCS, BIGFIVE, BRF, BS_4730, CA, CN-BIG5,
CN-GB, CN, CP-AR, CP-GR, CP-HU, CP037, CP038, CP273, CP274, CP275, CP278,
CP280, CP281, CP282, CP284, CP285, CP290, CP297, CP367, CP420, CP423, CP424,
CP437, CP500, CP737, CP775, CP803, CP813, CP819, CP850, CP851, CP852, CP855,
CP856, CP857, CP860, CP861, CP862, CP863, CP864, CP865, CP866, CP866NAV,
CP868, CP869, CP870, CP871, CP874, CP875, CP880, CP891, CP901, CP902, CP903,
CP904, CP905, CP912, CP915, CP916, CP918, CP920, CP921, CP922, CP930, CP932,
```

Для вывода текущих локальных настроек компьютера можно использовать команду `locale`. Если необходимо изменить локальные настройки, обратитесь к документации вашего дистрибутива, чтобы выяснить местоположение файла локальных настроек. После изменения локальных настроек необходимо перезагрузить компьютер. В листинге 2 приведен пример

локальных настроек по умолчанию на компьютере под управлением Linux.

## Листинг 2. Локальные настройки по умолчанию (Unicode UTF-8) на компьютере Linux

```
[tbost@samba ~]# locale
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=
```

Обратите внимание на то, что в листинге 2 локальные настройки имеют имена, позволяющие легко понять, что означает каждая из них (в отличие от многих общепринятых названий кодовых таблиц).

### Работа с наборами символов

Старые методы кодовых страниц DOS, используемые в Windows 9x и Samba 2.x, поддерживают использование расширенных наборов символов, но только по отдельности. Например, нельзя одновременно использовать поддержку для испанского, английского и французского языков. Помните об этом ограничении, если перед вами стоит задача обеспечить поддержку нескольких локальных настроек в этих средах.

Если вы обновляете Samba с версии 2.x на версию 3.x или изменяете локальные настройки Samba по умолчанию после использования настроек, отличных от английских, то файлы, содержащие специальные символы, могут оказаться неузнаваемыми. Обычно имена этих файлов превращаются в бессвязный набор символов. Как правило, это касается символов умлаутов и ударений, поскольку они содержатся только в ранее использовавшейся кодовой странице.

Если вы собираетесь назвать сервер Samba, используя символы, отличные от английских, то убедитесь, что локальные настройки Samba совпадают с локальными настройками компьютера Linux. Именно здесь директива UNIX `charset` играет важную роль в правильном конфигурировании Samba.

### Использование библиотек перекодировки

`iconv (libiconv)` – это распространяемая по лицензии GNU программа для преобразования символов из одной кодировки в другую. Для выполнения всех необходимых процедур по преобразованию символов Samba использует программу `iconv`, которая должна быть установлена на компьютере Linux. Хотя эти преобразования не всегда проходят безупречно, в целом программа неплохо справляется со своими обязанностями.

Если какой-либо символ отсутствует в таблице символов, то, вероятнее всего, имя файла будет преобразовано в бессвязный набор символов. Тем не менее, если определенный символ не поддерживается в одинаковых кодовых страницах для Linux или Windows, то, скорее всего, вместо него будет подставлен знак вопроса (?). Обычно в таких случаях в файл журнала Samba записываются события об ошибках, которые помогают разобраться с

источником проблемы. При возникновении таких ситуаций необходимо более подробно разобраться в том, как коды символов преобразуются на сервере Samba с использованием определенных кодовых страниц.

Также может возникнуть необходимость в компиляции библиотеки `libiconv`, которая обеспечивает поддержку определенной кодовой страницы, или применении исправлений при использовании сложных многобайтовых символов. Например, такая необходимость может возникнуть при использовании японского языка. Кодовая страница CP932 (также известная как `Shift_jis` и `Windows-31J`) – это кодовая страница Microsoft, используемая для японского языка. Библиотека `libiconv` содержит конвертер кодировки CP932, который преобразует кодовую страницу Windows 932 в Unicode. Тем не менее, для выполнения правильных преобразований необходимо установить исправление. В листинге 3 приведен пример, в котором устанавливается нужная библиотека.

### Листинг 3. Применение исправления, компиляция и инсталляция библиотеки `libiconv` для кодовой страницы CP932

```
[tbost@samba ~]# wget http://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.13.tar.gz
[tbost@samba ~]#
wget http://www2d.biglobe.ne.jp/~msyk/software/libiconv/libiconv-1.13-cp932.patch.gz
[tbost@samba ~]# tar -xvf libiconv-1.13.tar.gz
[tbost@samba ~]# cd libiconv-1.13
[tbost@samba ~]# gzip -dc ./libiconv-1.13-cp932.patch.gz | patch -p1
[tbost@samba libiconv-1.13]# ./configure --prefix=/usr/local/lib/libiconv
[tbost@samba libiconv-1.13]# make
[tbost@samba libiconv-1.13]# sudo make install
[tbost@samba libiconv-1.13]# /usr/local/lib/libiconv/bin/iconv -l | egrep -i '(-31j|-ms)|
EUC-JP-MS EUCJP-MS EUCJP-WIN EUCJPMS'
```

В листинге 3 выполняется следующая последовательность действий:

1. Загрузка исходного кода `libiconv`.
2. Загрузка исправления для CP932.
3. Распаковка исходного кода `libiconv`.
4. Смена директории на только что созданную директорию `libiconv-1.13`.
5. Задание в качестве директории для установки файлов директории `/usr/local/lib/libiconv`.
6. Компиляция исходного кода и установка программы с использованием разрешений `sudo`.
7. Проверка корректной установки исправления.

### Преобразование существующих файлов и директорий

Если файлам и директориям уже были присвоены имена в какой-то кодировке, то для поддержки целостности системы имен вы можете захотеть преобразовать их в другую кодировку. Для этого можно использовать инструмент `convmv`, написанный на Perl.

В листинге 4 выполняется загрузка запакованного tarball-файла и извлечение его содержимого. Поскольку `convmv` – это сценарий Perl, то его не нужно компилировать. Последняя команда в листинге 4 указывает `convmv` рекурсивно преобразовать все файлы в кодировке iso-8859-8 (Latin/Hebrew) в кодировку UTF-8.

### Листинг 4. Преобразование имен файлов с помощью `convmv`

```
[tbost@samba /]# wget http://www.j3e.de/linux/convmv/convmv-1.14.tar.gz
[tbost@samba /]# tar -xzvf convmv-1.14.tar.gz
```

```
[tbost@samba /]# cd convmv-1.14
[tbost@samba convmv-1.14]# sudo ./convmv -f iso-8859-8 -t utf8
-r --notest --replace /applications
```

## Настройка локализации Samba

Начиная с Samba версии 3 кодировкой по умолчанию является Unicode, что обеспечивает поддержку локализации без дополнительных настроек при условии, что все клиенты успешно работают с Unicode. Тем не менее, при использовании Samba версии 2.x или в случаях, когда в сети присутствуют старые Windows-клиенты, требуется внести дополнительные настройки в конфигурационный файл Samba, указав на необходимость использования нужных локальных настроек.

Когда все необходимые библиотеки перекодировки установлены, настройка Samba на использование локализации не представляет особых сложностей. Помните о том, что протокол CIFS изначально поддерживает символы, отличные от английских, и не нуждается в дополнительной настройке.

### Включение символьных наборов

Предположим, вы хотите настроить в Samba поддержку Windows-клиента, использующего испанский язык (если вам нужны другие локальные настройки, используйте соответствующие параметры кодовых страниц DOS и UNIX). В этом случае конфигурация будет выглядеть следующим образом.

Для включения символьных наборов выполните следующие действия:

1. Для безопасности сделайте резервную копию файла smb.conf.
2. Откройте файл smb.conf в любом текстовом редакторе.
3. В разделе глобальных параметров добавьте следующие директивы:

```
#===== Global Settings =====
[global]
dos charset = CP850
unix charset = IS08859-1
```

Эти параметры конфигурации представляют собой пример использования кодовой страницы 850 на Windows-клиентах в то время, как на сервере Samba используются локальные настройки IS08859-1. Ваша конфигурация, вероятно, будет использовать другие кодовые страницы и локальные настройки.

4. Проверьте новую конфигурацию на предмет синтаксических ошибок или неподдерживаемых символов:

```
[tbost@samba /]# testparm -v
Load smb config files from /etc/samba/smb.conf
rlimit_max: rlimit_max (1024) below minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

Вы должны увидеть сообщение `Loaded services file OK`. Если вы видите предупреждения или сообщения об ошибках, ссылающиеся на преобразование символов, то убедитесь, что библиотека `libiconv` поддерживает необходимую кодовую страницу.

## 5. Перезапустите Samba или перезагрузите конфигурационный файл.

Теперь попробуйте подключиться к Windows-клиенту и просмотреть директории, содержащие символ ударения или другие символы, отличные от английских:

```
[tbost@samba /]# smbclient -U tbost //windowsclientname/applications  
Enter tbost's password:
```

Здесь `windowsclientname` – это NetBIOS-имя Windows-клиента в сети, а `applications` – имя общего файлового ресурса на Windows-клиенте. После того, как вы подключились к общему ресурсу, найдите директории, содержащие символы, отличные от английских, и убедитесь в том, что они отображаются корректно.

## Ресурсы

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Internationalization](#) (EN).
- Ознакомьтесь со списком [идентификаторов кодовых страниц IBM](#)(EN) и узнайте больше о том, как IBM классифицирует кодовые страницы для различных языков.
- Ознакомьтесь со списком [идентификаторов кодовых страниц Microsoft](#) (EN) и узнайте больше о доступных кодовых страницах Windows для различных языков.
- В [главе 30](#) (EN) руководства Samba обсуждается использование Unicode в Samba 3.x и исправление `iiconv` для поддержки японского языка.
- Узнайте о [настройке SWAT для поддержки локализации](#) (EN) из руководства Samba и используйте SWAT для управления неанглоязычными средами.
- Узнайте о [библиотеке GNU libiconv](#) (EN) и о том, как она конвертирует символьные наборы.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- В [разделе Linux сайта developerWorks](#) можно найти сотни [пошаговых инструкций и руководств](#), загрузить программные продукты, а также получить ссылки на форумы и многие другие ресурсы, ориентированные на разработчиков и администраторов Linux.

# Изучаем Linux, 302 (смешанные среды): Управление учетными записями пользователей и групп

*Планирование системы управления учетными записями пользователей и групп*

Трейси Бост, консультант и преподаватель, Свободный писатель

**Описание:** Если вам доводилось работать с учетными записями пользователей и групп, то вы наверняка знаете, что в смешанных средах их применение не всегда проходит гладко для пользователей, что является типичным источником проблем как для пользователей, так и для системных администраторов. К счастью в составе Samba есть инструменты, которые помогают управлять этим процессом. Из этой статьи вы узнаете о том, как управлять учетными записями пользователей и групп в смешанной среде.

**Дата:** 28.06.2012

**Уровень сложности:** средний

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Учетные записи UNIX.
- Управление учетными записями Samba.
- Сопоставление учетных записей.
- Принудительная установка прав доступа к файлам и директориям для учетных записей.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 313.1 темы 313. Цель имеет вес 4.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и в общих чертах понимает назначение Samba (о чем рассказывалось в предыдущей статье "[Изучаем Linux, 302: основные принципы](#)"). Для выполнения примеров этой статьи на вашем компьютере должно быть инсталлировано программное обеспечение Samba, и вы должны иметь доступ к сетевому клиенту под управлением Windows.

## **Понимание учетных записей пользователей и групп UNIX**

Ваш сервер Samba не существует сам по себе. Пользователи должны получать доступ к файлам и директориям, но прежде, чем они смогут это сделать, они должны пройти аутентификацию. Пользователи могут подключаться с рабочих станций, работающих под управлением как Linux, так и Windows. Так или иначе, каждый пользователь должен иметь учетную запись, которую может распознать сервер Samba.

После того, как пользователь прошел аутентификацию, он должен получить права доступа к файлам, директориям и службам печати. *Группы* – это один из поддерживаемых Samba компонентов, который позволяет более эффективно управлять этими разрешениями.

Внутренняя база данных SAM – это посредник между локальными учетными записями UNIX и учетными записями удаленных компьютеров. Существует несколько методов, позволяющих пользователям пройти аутентификацию на сервере Samba, но прежде чем перейти к рассмотрению учетных записей Samba, необходимо твердо понимать основы управления учетными записями пользователей и групп в UNIX.

### **Учетные записи пользователей**

Когда вы создаете локальную учетную запись на компьютере Linux, например, с помощью команды `useradd`, то информацию об учетной записи сохраняется в файле `/etc/passwd`. В этом файле хранится такая информация, как имя пользователя, домашняя директория, командная оболочка и некоторые комментарии, связанные с учетной записью. Обычно эти учетные записи называются *локальными учетными записями UNIX*. В этой статье термины *учетная запись UNIX* и *локальная учетная запись* эквивалентны.

В листинге 1 создается локальная учетная запись с именем пользователя *monty* и описанием *Monty Python* в разделе комментариев (-C), задается ее домашняя директория (-m) и назначается командная оболочка по умолчанию `/bin/bash` (-s).

### **Листинг 1. Создание локальной учетной записи**

```
[tbost@samba ~]$ sudo useradd -c'Monty Python' -m -s /bin/bash monty
[tbost@samba ~]$ less /etc/passwd | grep monty
monty:x:504:504:Monty Python:/home/monty:/bin/bash
[tbost@samba ~]$
```

Каждая строка в файле `/etc/passwd` содержит одну учетную запись и состоит из семи полей, разделенных двоеточиями (:). Для управления учетными записями Samba представляют интерес три поля: первое (имя пользователя), третье (идентификатор пользователя, UID) и четвертое (идентификатор группы, GID).

### **Учетные записи групп**

Учетные записи групп играют важную роль, упрощая администрирование компьютеров с несколькими пользователями. Если вы управляете сервером Samba, то типичной задачей является назначение группам прав доступа к определенным файлам, директориям и принтерам.

Так же, как и в случае с учетными записями пользователей, если вы работаете с локальной конфигурацией учетных записей Samba, то в большинстве случаев необходимо создать учетные записи групп UNIX на локальном сервере Samba. Информация об учетных записях групп UNIX хранится в файле `/etc/group`. В некоторых дистрибутивах Linux для каждого нового пользователя создается локальная *частная группа*. Именно так обстоит дело в случае с пользователем *monty*:

```
[tbost@samba ~]$ less /etc/group | grep monty
monty:x:504:
[tbost@samba ~]$
```

Из приведенного листинга видно, что для пользователя `monty` была создана учетная запись частной группы. Если вы работаете в смешанной среде, где есть компьютеры Windows, то помните о том, что Windows не позволяет присваивать одинаковые имена учетным записям пользователей и групп.

Так же, как и в случае с учетными записями пользователей, для того чтобы Samba могла использовать учетные записи групп, они должны существовать на локальном сервере UNIX. Группы создаются с помощью команды `groupadd` (см. листинг 2) или путем редактирования файла `/etc/group` с помощью любого текстового редактора, например, `vim`.

## Листинг 2. Создание группы и добавление в нее пользователя

```
[tbost@samba ~]$ sudo groupadd accounting
[tbost@samba ~]$ sudo usermod -G accounting monty
[tbost@samba ~]$ less /etc/group | grep accounting
accounting:x:506:monty
[tbost@samba ~]$
```

Для создания группы и добавления в нее пользователя в листинге 2 используются команды `/sbin/groupadd` и `/sbin/usermod`. Если необходимо добавить в группу несколько пользователей, то можно создать сценарий или добавить пользователей непосредственно в файл `/etc/group`. Члены группы должны быть перечислены в последнем поле и разделяться запятыми ( , ). Если вы создаете группы вручную, то не забывайте о том, что каждая группа должна иметь уникальный идентификатор (GID).

## Управление учетными записями Samba

В стандартной конфигурации Samba информация об учетных записях хранится в одной из следующих внутренних баз данных паролей:

- `smbpasswd`
- `tdbsam`
- `ldapsam`

### Использование `smbpasswd` и `tdbsam`

База данных `smbpasswd` используется по умолчанию во всех версиях Samba ниже 3.4. В Samba версии 3.4 `smbpasswd` была объявлена устаревшей и вместо нее используется база данных `tdbsam` (эта база данных также рекомендуется для сред, содержащих менее 250 пользователей).

База данных `tdbsam` считается лучше масштабируемой по сравнению с `smbpasswd`. Если вы работаете с версией Samba, в которой по умолчанию используется `smbpasswd`, то можете заменить эту базу данных на `tdbsam`, указав в файле `smb.conf` параметр `passdb = tdbsam` в разделе `global`.

Однако `smbpasswd` – это не просто база данных, но еще и инструмент из состава пакета Samba, позволяющий управлять учетными записями Samba в простых конфигурациях. Для создания учетной записи Samba необходимо иметь права пользователя `root`. Прежде чем создавать учетную запись Samba, она должна существовать на локальном сервере Linux. В листинге 3 приведен пример создания учетной записи пользователя Samba с помощью

smbpasswd.

### **Листинг 3. Создание учетной записи пользователя Samba с помощью smbpasswd**

```
[tbost@samba ~]$ sudo smbpasswd -a monty  
New SMB password:  
Retype new SMB password:  
Added user monty.
```

Пользователи могут использовать **smbpasswd** для смены своих паролей, как показано в листинге 4.

### **Листинг 4. Локальный пользователь может изменять свой пароль с помощью smbpasswd**

```
[monty@samba ~]$ smbpasswd  
Old SMB password:  
New SMB password:  
Retype new SMB password:  
Password changed for user monty  
[monty@samba ~]$
```

Можно также так настроить Samba для синхронизации паролей, чтобы при каждом изменении пользователем пароля своей локальной учетной записи также обновлялся и пароль Samba:

```
[global]  
unix password sync = yes
```

Если в течение какого-то времени пользователю не нужен доступ к серверу Samba, то можно временно отключить учетную запись и включить ее позже. Если пользователю вообще не нужен доступ к Samba, то его учетную запись можно удалить. В листинге 5 показано, как это сделать.

### **Листинг 5. Отключение, включение и удаление учетной записи Samba с помощью smbpasswd**

```
[tbost@samba ~]$ sudo smbpasswd -d monty  
Disabled user monty.  
[tbost@samba ~]$ sudo smbpasswd -e monty  
Enabled user monty.  
[tbost@samba ~]$ sudo smbpasswd -x monty  
Deleted user monty.  
[tbost@samba ~]$
```

## **Использование pdbedit**

В составе Samba имеется многофункциональный инструмент под названием **pdbedit**. Этот инструмент может работать с учетными записями, хранящимися в любой из трех вышеперечисленных баз данных. Помимо создания, изменения и удаления пользователей,

**pdbedit** позволяет выполнять следующие действия:

- Отображать список учетных записей пользователей.
- Указывать домашние директории.
- Импортировать учетные записи пользователей.
- Назначать политики учетных записей.

При работе с базой данных **tdbsam** можно использовать как **pdbedit**, так и **smbpasswd** (листинг 6). Для выполнения всех команд **pdbedit** необходимо обладать правами пользователя root.

#### Листинг 6. Выполнение различных действий с внутренней базой данных при помощи **smbpasswd** и **pdbedit**

```
[tbost@samba ~]$ sudo smbpasswd -a monty
New SMB password:
Retype new SMB password:
Added user monty.
[tbost@samba ~]$ sudo pdbedit -L
monty:504:Monty Python
[tbost@samba ~]# sudo pdbedit -L --verbose
Unix username:      monty
NT username:
Account Flags:      [U          ]
User SID:            S-1-5-21-2247757331-3676616310-3820305120-1001
Primary Group SID:  S-1-5-21-2247757331-3676616310-3820305120-513
Full Name:          Monty Python
Home Directory:     \\samba\monty
HomeDir Drive:
Logon Script:
Profile Path:       \\samba\monty\profile
Domain:             SAMBA
Account desc:
Workstations:
Munged dial:
Logon time:         0
Logoff time:        never
Kickoff time:       never
Password last set: Tue, 24 May 2011 14:19:46 CDT
Password can change: Tue, 24 May 2011 14:20:16 CDT
Password must change: Tue, 24 May 2011 14:20:16 CDT
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

В листинге 6 продемонстрировано, как можно создать пользователя с помощью **smbpasswd**, а затем вывести список пользователей Samba с помощью **pdbedit**.

Утилиту **pdbedit** можно также использовать для задания политик учетных записей. Ниже перечислены названия политик, которыми можно управлять:

- **min password length** (минимальная длина пароля)
- **password history** (сохранять историю паролей)
- **user must logon to change password** (требовать смену пароля при следующем входе в систему)
- **maximum password age** (максимальный срок действия пароля)
- **minimum password age** (минимальный срок действия пароля)

- **lockout duration** (время блокировки при неудачном входе)
- **reset count minutes** (сброс счетчика минут блокировки учетной записи)
- **bad lockout attempt** (блокировка при неудачной попытке входа)
- **disconnect time** (время до отключения)
- **refuse machine password change** (отклонять изменение пароля компьютера)

В листинге 7 мы устанавливаем минимальную длину пароля равной восьми символам, а затем устанавливаем максимальный срок действия пароля равным 30 дням. Опция **-P** принимает строковый аргумент, который в точности должен совпадать с именем одной из вышеперечисленных политик, а опция **-C** задает значение для выбранной политики.

### Листинг 7. Управление учетными записями с помощью pdbedit

```
[tbost@samba ~]$ sudo pdbedit -P 'min password length' -C 8
account policy "min password length" description: Minimal password length (default: 5)
account policy "min password length" value was: 5
account policy "min password length" value is now: 8
[tbost@samba ~]$ sudo pdbedit -P 'maximum password age' -C 30
...
account policy "maximum password age" value was: 4294967295
account policy "maximum password age" value is now: 30
```

Для получения дополнительной информации о доступных командах утилиты **pdbedit** обратитесь к ее man-странице или выполните команду **pdbedit -h**.

### Использование ldapsam

Если ваша среда содержит более 250 пользователей, то можно использовать внутреннюю базу данных **ldapsam**. Из всех трех вышеперечисленных баз данных только **ldapsam** может хранить данные об учетных записях групп. Если вся информация о пользователях и группах хранится во внутренней базе данных **ldap**, то идентификаторы пользователей и групп (UIDs и GIDs, соответственно) будут согласованы между всеми вашими серверами. Поскольку настройка LDAP выходит за рамки этой статьи, я ограничусь информацией о том, что расположение LDAP-сервера определяется параметром **idmap backend** в файле **smb.conf**.

В следующем примере параметр **idmap backend** говорит Samba о том, что в качестве хранилища учетных данных она должна использовать службу каталогов LDAP, запущенную на узле с именем **directory-services.example.org**. В этом случае необходимо иметь рабочий сервер LDAP, предварительно настроенный на работу с Samba (более подробно утилита **idmap** будет рассмотрен в следующем разделе).

```
[global]
idmap backend = ldap:ldap://directory-services.example.org:636
```

### Сопоставление учетных записей

Если ваш сервер Samba является рядовым сервером одного домена, то, вероятно, вы просто будете использовать файлы сопоставлений. Однако, если в вашей среде есть пользователи, которые подключаются к серверу Samba из другого домена, то корректное сопоставление идентификаторов пользователей и групп помогает выполнить утилита **idmap**.

### Сопоставление пользователей с помощью sampasswd и TDB -файлов

Если имена пользователей Windows, подключающихся к серверу Samba, совпадают с именами на сервере Samba, то необходимость в файле сопоставлений отсутствует. В противном случае можно создать файл сопоставлений, который связывает имена

пользователей. Не забывайте о том, что хотя в Linux все имена и команды различают регистр, в Windows это *не так*. Таким образом, имя Windows-пользователя *TBost* и локальная учетная запись *tbost* – это не то же самое. В таблице 1 показано сопоставления учетных записей Windows и UNIX.

### Таблица 1. Учетные записи Windows и UNIX для сопоставления

#### Windows UNIX

Monty	monty
bosst	tbost
sue.george	sue

Когда вы создаете учетные записи Samba, используйте имена учетных записей Windows. В файле smb.conf можно указать местоположение файла, содержащего сопоставления учетных записей с соответствующими учетными записями UNIX. В листинге 8 показано сопоставление учетных записей в UNIX.

### Листинг 8. Простое сопоставление учетных записей в UNIX

```
[tbost@samba ~]$ sudo vi /etc/samba/smb.conf
[global]
username map = /etc/samba/smbusers
...
...
...
[tbost@samba ~]$ sudo vi /etc/samba/smbusers
# Unix_name = SMB_name1 SMB_name2 ...
root = administrator admin
nobody = guest pcguest smbguest
monty = Monty
tbost = bosst
sue = sue.george
```

Команда в листинге 8 настраивает параметр `username map` на использование файла `/etc/samba/smbusers` в качестве файла сопоставлений. Процедура сопоставления учетных записей достаточно простая: слева указываются учетные записи UNIX, справа – учетные записи Samba, а между ними ставится знак равенства (=). При подключении пользователя Samba выполняет сопоставление соответствующей учетной записи.

### Сопоставление групп

В типовой среде Samba сопоставление групп настраивается при помощи команды `groupmap` из состава Samba. Предположим, что пользователи Monty, bosst и sue.george являются членами групп Domain Admins, Domain Users и Domain Guests. Если вы хотите установить для групп этих пользователей такие же разрешения, что и для групп UNIX на сервере Samba, то добавьте имена учетных записей пользователей UNIX в каждую группу:

```
adm:x:4:root,adm,daemon,monty,tbost,sue
users:x:100:monty,tbost,sue
guests:x:507:monty,tbost,sue
```

Это лишь часть списка групп на сервере Samba. Группы adm и users были созданы во время инсталляции операционной системы Linux. Вам необходимо добавить каждого пользователя в соответствующую группу (таблица 2).

**Таблица 2. Учетные записи групп Windows и UNIX для сопоставления**

Windows	UNIX	Windows relative ID (RID)	UNIX GID
Domain Admins	adm	512	4
Domain Users	users	513	100
Domain Guests	guests	514	507

Команда `net groupmap` может выполнять сопоставление доменных групп (листинг 9), а команда `net groupmap list` выводит список этих сопоставлений. Начиная с Samba версии 3.x, доступна новая функциональность, предназначенная для сопоставления относительных идентификаторов Windows (RID) и идентификаторов групп UNIX (GID).

### Листинг 9. Сопоставление групп с помощью команды `groupmap`

```
[tbost@samba ~]$sudo net groupmap add ntgroup="Domain Admins" unixgroup=adm \
rid=512 type=d
Successfully added group Domain Admins to the mapping db as a domain group
[tbost@samba ~]$ sudo net groupmap add ntgroup="Domain Users" unixgroup=users \
rid=513 type=d
Successfully added group Domain Users to the mapping db as a domain group
[tbost@samba ~]$sudo net groupmap add ntgroup="Domain Guests" unixgroup=guests \
rid=514 type=d
Successfully added group Domain Guests to the mapping db as a domain group
[tbost@samba ~]$sudo net groupmap list
Domain Users (S-1-5-21-2247757331-3676616310-3820305120-513) -> users
Domain Guests (S-1-5-21-2247757331-3676616310-3820305120-514) -> guests
Domain Admins (S-1-5-21-2247757331-3676616310-3820305120-512) -> adm
```

В листинге 9 выполняется следующая последовательность действий для сопоставления групп:

1. Выполнение команды `net groupmap add` с правами пользователя root для сопоставления Windows-группы Domain Admin (`ntgroup='Domain Admin'`) с группой UNIX adm (`unixgroup=adm`).  
Выполнение этих действий для сопоставления каждой группы.
2. Последняя команда листинга 9 выводит список сопоставлений групп.

### Сопоставление идентификационных данных

Рассмотренные сопоставления оказываются достаточными в большинстве сред. Тем не менее, если вы управляете более сложной средой, например, средой с несколькими серверами Samba или рабочими станциями из различных доменов, которые подключаются к серверу Samba, то следует знать о сопоставлении идентификационных данных (IDMAP) и Winbind. IDMAP может помочь решить проблемы функциональной совместимости между идентификаторами безопасности Windows (SID) и локальными идентификаторами пользователей (UID) или групп (GID) UNIX.

Если сервер Samba является членом домена Windows, то для сопоставления идентификаторов SID и UID (или GID) можно использовать Winbind. В файле smb.conf можно настроить диапазон значений параметра `idmap` и указать время, в течение которого Winbind должен кэшировать информацию об учетных записях:

```
[global]
idmap uid = 20000-50000
idmap gid = 20000-50000
```

```
winbind cache time = 300
```

Параметры, заданные в этом примере, указывают Winbind использовать диапазон локальных идентификаторов UID 20000-50000 и диапазон идентификаторов GID 20000-50000. Эта конфигурация относительно безопасна для сервера Samba, на котором не предполагается размещать несколько тысяч локальных учетных записей пользователей или групп. Параметр `winbind cache time = 300` говорит Winbind о том, что информация об учетных записей должна кэшироваться 300 секунд. По умолчанию Winbind хранит сопоставления в файле `winbind_idmap.tdb`.

### Принудительное назначение учетных записей по умолчанию

Вместо добавления каждого пользователя в группу более удобным может оказаться использование параметров `force user` и `force group`. Когда эти параметры заданы, они говорят Samba о том, что авторизованный пользователь при подключении должен иметь те права доступа, которые были установлены для указанного пользователя и группы. Это особенно полезно при настройке общего ресурса, доступного многим пользователям, для которых достаточно использовать общие права доступа:

```
[global]
username map = /etc/samba/smbusers
force user = guest
force group = +employees
```

В приведенном примере параметр `force user` рассматривает всех подключившихся пользователей, обращающихся к файлам, в качестве пользователя `guest`. При этом каждый пользователь должен подключаться с использованием действующей учетной записи. В рассмотренном примере в качестве учетных записей пользователей будет принудительно использоваться учетная запись `guest`, а в качестве учетных записей групп – учетная запись `employees`.

## Ресурсы

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Managing user accounts and groups \(EN\)](#).
- [Базы данных для хранения учетных записей Samba \(EN\)](#) – глава 11 руководства Samba 3.x.
- [Сопоставление групп \(EN\)](#) – глава 12 руководства Samba 3.x.
- Детальное описание [инструмента pdbedit \(EN\)](#) из руководства `pdbedit`.
- [Сопоставление идентификационных данных \(IDMAP\) \(EN\)](#) для автономных и основных контроллеров домена – глава 14 руководства Samba.
- На Web-сайте [программы сертификации LPIC \(EN\)](#) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302 \(EN\)](#), а также [примеры заданий и вопросов \(EN\)](#).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI \(EN\)](#) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.

# Изучаем Linux, 302 (смешанные среды): Аутентификация и авторизация

*Механизмы проверки подлинности и настройка контроля доступа*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") для получения сертификата системного администратора Linux. Из этой статьи вы узнаете, как задавать и хранить пароли, интегрировать Samba с LDAP и использовать списки контроля доступа для защиты операционной системы Linux.

[Больше статей из этой серии](#)

**Дата:** 05.07.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Настройка локальной базы данных паролей.
- Формат файла smbpasswd.
- Синхронизация паролей между Samba и другими системами.
- Другие хранилища паролей.
- Интеграция Samba с протоколом Lightweight Directory Access Protocol (LDAP).
- Списки контроля доступов (Access control lists, ACLs).

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 313.2 темы 313. Цель имеет вес 8.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. Кроме этого, у вас должен быть доступ к среде Windows, которую можно использовать для проверки работы механизмов проверки подлинности и авторизации.

## Механизмы аутентификации Samba

Samba позволяет по-разному хранить пароли во *внутренних хранилищах паролей*, как с хранением данных хранятся на локальном диске, так и с использованием сетевых функций. Кроме того, в Samba имеются механизмы, позволяющие серверу Linux использовать систему аутентификации подлинности Samba при входе пользователей в систему.

Поскольку Samba существует на протяжении почти двух десятилетий, в ее арсенале накопился определенный технический инструментарий. Несмотря на то, что на смену старым механизмам аутентификации пришли новые, старые механизмы все еще поддерживаются. В

документации Samba вы найдете ссылки на эти старые механизмы. Эта путаница еще более усложняют инструменты командной строки, названия которых похожи на названия определенных технологий.

Несмотря на свое название, *внутреннее хранилище паролей* хранит не только пароли, но также различную информацию об учетных записях и другие атрибуты. Сами данные могут также хранится на сервере LDAP, а внутреннее хранилище Samba в этом случае выступает посредником между Samba и LDAP.

Все становится еще интереснее в результате того, что операционные системы Microsoft используют хэши паролей, формат которых отличается от формата паролей UNIX. Эти пароли *хэшируются*, т. е. подвергаются одностороннему шифрованию, в результате чего их невозможно сопоставить паролям в формате UNIX и Samba. В результате аутентификация клиента Samba в базе данных паролей UNIX становится невозможной.

### **Параметры локальной базы данных паролей**

*Локальной базой данных паролей* называется хранилище паролей, которое хранит информацию на сервере, а не осуществляет проверку подлинности через сеть. У таких хранилищ низкая производительность, но зато они просты в использовании.

В литературе можно встретить упоминания о хранилищах *открытых паролей*. Много лет назад клиенты Windows передавали на сервер учетные данные в незашифрованном (открытом) виде. Тогда было возможно хэшировать пароли в UNIX-формате и сопоставлять полученные результаты с локальной базой данных паролей. Сегодня клиенты Windows не передают по сети пароли в открытом виде (по крайней мере, без дополнительного вмешательства в системный реестр), и все стараются избегать этого. Таким образом, вы можете встретить упоминания о хранилищах открытых паролей, но вряд ли стоит использовать их на практике.

Большинство старых систем аутентификации используют хранилище паролей *smbpasswd*. Эта база данных хранит информацию об учетных записях в простом текстовом файле. Эта информация включает в себя имя учетной записи, хэши паролей и некоторую базовую информацию. Эта базовая информация достаточно простая и не содержит дополнительных атрибутов, с которыми работают различные инструменты администрирования Microsoft.

На сегодняшний день предпочтительным локальным хранилищем является *tdbsam*. Вспомните файлы Trivial Database (TDB), которые рассматривались в теме 310.3 (см. раздел Ресурсы): TDB-файлы позволяют получать быстрый и надежный доступ к информации, хранящейся в виде пар "ключ-значение". В *tdbsam* информация хранится в формате, похожем на формат базы данных Microsoft Windows NT Security Account Manager (SAM), поэтому почти все, что может хранится в SAM, сможет понять сервер Samba. Именно поэтому *tdbsam* обеспечивает высокий уровень совместимости с операционными системами Microsoft.

Недостатком *tdbsam* является то, что информация хранится в двоичном формате, поэтому невозможно быстро просмотреть ее, просто заглянув внутрь файла. В статье "[Изучаем Linux, 302 \(смешанные среды\): файлы базы данных Trivial Database](#)" (developerWorks, март 2011 г.) показано, как можно извлекать из TDB-файла отдельные ключи и их значения, но как использовать это на практике – предстоит решать вам. Этот файл называется *passdb.tdb*.

### **Использование базы данных smbpasswd**

При новой инсталляции Samba вы вряд ли захотите использовать базу данных *smbpasswd*, но в некоторых старых инсталляциях она все еще применяется, поэтому важно знать принципы ее работы.

База данных для хранения паролей настраивается с помощью параметра *passdb backend*. В следующем примере показано, как выбрать в качестве хранилища базу данных *smbpasswd*:

```
[global]
passdb backend=smbpasswd:/etc/samba/smbpasswd
```

Мы видим, что в данном примере используется глобальный параметр `passdb backend`, которому присвоены значение `smbpasswd` и путь к файлу, разделенные двоеточием (:). Двоеточие и путь не являются обязательными, однако лучше использовать их. Если этого не сделать, то Samba разместит файл в директории по своему усмотрению. Samba создает этот пустой файл при перезапуске.

Для добавления пользователя используется команда `smbpasswd`. В листинге 1 приведен пример добавления пользователя и показан итоговый результат в файле `smbpasswd`.

### Листинг 1. Добавление пользователя в файл `smbpasswd`

```
# smbpasswd -a sean
New SMB password:
Retype new SMB password:
Added user sean.
# cat smbpasswd
sean:1001:01FC5A6BE7BC6929AAD3B435B51404EE: \
0CB6948805F797BF2A82807973B89537:[U           ]:LCT-4DCDE4D8:
```

В листинге 1 мы сначала добавляем пользователя *sean*, используя для этого флаг `-a`. Этот пользователь должен существовать в локальной базе данных UNIX, в противном случае выполнение команды завершится с ошибкой. После этого в файле `smbpasswd` появляется единственная строка (в листинге 1 она для удобства разбита на две строки), содержащая поля, разделенные двоеточиями. Перечислим эти поля по очереди:

- Имя пользователя.
- UNIX-идентификатор пользователя.
- Устаревший LM-хэш пароля (поскольку этот хэш содержит уязвимости, то его использование равносильно использованию открытых паролей).
- Безопасный хэш Windows NT (необходим для аутентификации современных клиентов).
- Флаги учетной записи (в нашем примере содержится единственный флаг `U`, указывающий на учетную запись пользователя); информацию обо всех флагах можно найти на *man*-странице `smbpasswd(5)`.
- Время последнего изменения (Last Changed Time, LCT) учетной записи (шестнадцатеричное значение является закодированной версией временной метки UNIX).

Пользователи могут менять свои пароли, запуская команду `smbpasswd`, а пользователь `root` может сменить пароль любого пользователя, указав его имя. Флаг `-a` нужен только для добавления учетной записи.

Файлы с открытыми паролями имеют свои ограничения, связанные со скоростью извлечения данных и хранением метаданных учетных записей. Чтобы обойти эти ограничения, была создана база данных `tdbsam`.

### Использование базы данных `tdbsam`

База данных `tdbsam` была разработана в качестве замены `smbpasswd` и может хранить больше информации об учетных записях. Кроме того, благодаря использованию простой базы

данных, она работает гораздо быстрее.

К сожалению, нет четкой информации об ограничениях масштабируемости базы данных `tdbsam`. Официальная документация рекомендует переходить на использование базы данных LDAP, если количество пользователей превышает 250, а в других источниках приводятся примеры, когда `tdbsam` работала с тысячами клиентов. Вероятно, важным фактором является количество одновременных запросов к базе данных, на которое влияет число пользователей, рабочая нагрузка и задержки между клиентами и сервером. Как и в случае со многими приложениями, лучше всегда отслеживать ключевые показатели производительности (такие, как задержка, загрузка процессора и операции дискового ввода/вывода) и по ним заранее предсказывать образование "узких мест" в сети.

Настройка базы данных `tdbsam` так же проста, как и настройка `smbpasswd`. Задайте параметр `passdb backend=tdbsam` и необязательное имя файла в разделе `global` и перезапустите Samba. По умолчанию TDB-файл, в котором будет храниться информация об аутентификации, будет называться `passdb.tdb`.

Для управления пользователями в базе данных `tdbsam` также можно использовать утилиту `smbpasswd`. В листинге 2 показано содержимое базы данных `tdbsam` до и после добавления учетной записи пользователя с помощью `smbpasswd`.

## Листинг 2. Создание учетной записи пользователя с помощью `smbpasswd` и просмотр базы данных `tdbsam`

```
# tdbdump passdb.tdb
{
key(13) = "INFO/version\00"
data(4) = "\03\00\00\00"
}
# smbpasswd -a sean
New SMB password:
Retype new SMB password:
Added user sean.
# tdbdump passdb.tdb
{
key(13) = "RID_00000bba\00"
data(5) = "sean\00"
}
{
key(10) = "USER_sean\00"
data(205) = "\00\00\00\00....EC\04\00\00"
}
{
key(13) = "INFO/version\00"
data(4) = "\03\00\00\00"
}
```

Изначально файл `passdb.tdb` содержит единственный ключ, содержащий строку с версией. Затем в базу данных добавляется учетная запись пользователя, так же, как и в [предыдущем примере](#). Утилита `smbpasswd` обращается к файлу `smb.conf` и на его основе определяет, каким образом следует добавлять пользователя. Теперь база данных паролей содержит два дополнительных ключа: первый сопоставляет идентификатор Microsoft Relative ID (RID) имени пользователя, а второй содержит информацию о пользователе. Основная часть данных хранится в двоичном формате, поэтому для их декодирования нужно выяснить структуру размещения данных, обратившись к исходному коду Samba.

## Проверка подлинности с помощью LDAP

Файловый формат TDB обеспечивает хорошую производительность и подходит почти для всех организаций с небольшим количеством пользователей. Если же использование TDB-файлов уже не соответствует требованиям организации, то необходимо переходить к проверке подлинности с помощью LDAP-сервера. Каталог LDAP хорошо подходит для проверки подлинности по своей природе, поскольку имеет древовидную структуру. На базе LDAP построены службы Active Directory® Domain Services компании Microsoft.

### Настройка LDAP-сервера

Интеграция Samba и LDAP начинается с настройки LDAP-сервера. Подробное описание этого процесса выходит за рамки нашей статьи, но вы можете подробно прочитать об этом в материалах для подготовки к экзамену LPIC 301 (см. раздел [Ресурсы](#)). Поскольку для интеграции Samba и LDAP необходимо иметь представление о LDAP, то мы советуем перечитать и освежить в памяти указанные материалы.

Самым распространенным LDAP-сервером для Linux является OpenLDAP. Его конфигурация хранится в файле slapd.conf. В листинге 3 приведен пример конфигурационного файла, полностью готового для интеграции с Samba.

### Листинг 3. Конфигурационный файл slapd.conf

```
# Include the core schema files, and the perquisites for samba.schema
include          /etc/openldap/schema/core.schema
include          /etc/openldap/schema/cosine.schema
include          /etc/openldap/schema/inetorgperson.schema
include          /etc/openldap/schema/nis.schema
include          /etc/openldap/schema/samba.schema

database        bdb
# Configure the tree and the admin user
pidfile         /var/run/openldap/slapd.pid
suffix          "dc=ertw, dc=com"
rootdn          "cn=admin, dc=ertw, dc=com"
rootpw          linux
directory       /var/lib/ldap

# Indexes
index objectclass           eq
index cn                  pres,sub,eq
index sn                  pres,sub,eq
# For storing Unix accounts in LDAP
index uidNumber            eq
index gidNumber            eq
index memberUid            eq
# Samba specific
index uid                  pres,sub,eq
index displayName          pres,sub,eq
index sambaSID             eq
index sambaPrimaryGroupSID eq
index sambaDomainName      eq

index default              sub
```

В листинге 3 для всех объектов используется суффикс `dc=ertw, dc=com`. В первом разделе загружаются все элементы схемы, необходимые для интеграции с Samba. Схема `core.schema` требуется для выполнения базовых функций OpenLDAP, а схема `samba.schema` дополняет

список классами Samba `objectClass`. Другие схемы являются зависимостями для схемы `samba.schema` и должны быть перечислены раньше, поскольку файлы обрабатываются по очереди.

Следующий раздел, начинающийся с `database bdb`, говорит о том, что будет использоваться база данных Berkeley, и показывает некоторую информацию о дереве, включая пользователя с правами администратора и директорию. В остальной части конфигурационного файла настраиваются индексы, ускоряющие поиск в дереве; эти индексы наследуются из конфигурации OpenLDAP по умолчанию и документации Samba.

Теперь необходимо заполнить дерево LDAP различными контейнерами для хранения пользователей, компьютеров и групп. Также необходимо сопоставить идентификатор безопасности (SID) вашего сервера с некоторыми записями. Это довольно непросто, но, к счастью, этот процесс можно упростить с помощью инструментария `smbldap-tools`. Скорее всего, ваш дистрибутив должен содержать пакет с этими инструментами, в противном случае их необходимо загрузить и установить вручную (ссылку на загрузку вы найдете в разделе [Ресурсы](#)).

После установки инструментов `smbldap-tools` откройте конфигурационный файл, укажите необходимые данные (например, имя домена или рабочей группы) и задайте административные параметры для подключения к LDAP-серверу. После этого запустите команду `smbldap-populate`, которая построит дерево. Вывод показан в листинге 4.

#### Листинг 4. Заполнение дерева LDAP

```
# smbldap-populate
Populating LDAP directory for domain BOB (S-1-5-21-2287037134-1443008385-640796334)
(using builtin directory structure)

entry dc=ertw,dc=com already exist.
adding new entry: ou=People,dc=ertw,dc=com
adding new entry: ou=Groups,dc=ertw,dc=com
adding new entry: ou=Computers,dc=ertw,dc=com
adding new entry: ou=Idmap,dc=ertw,dc=com
adding new entry: uid=root,ou=People,dc=ertw,dc=com
...
adding new entry: cn=Replicators,ou=Groups,dc=ertw,dc=com
adding new entry: sambaDomainName=BOB,dc=ertw,dc=com

Please provide a password for the domain root:
Changing UNIX and samba passwords for root
New password:
Retype new password:
```

Из листинга 4 видно, что команда `smbldap-populate` определила идентификатор SID и имя домена локального компьютера, и выполняет построение структуры каталогов LDAP. Наконец, утилита запрашивает пароль пользователя `root` и синхронизирует его с деревом LDAP.

#### Подключение Samba к LDAP

При настройке Samba на использование LDAP необходимо указать серверу, как следует выполнить привязку к дереву LDAP. В листинге 5 показана минимальная конфигурация Samba, позволяющая выполнять проверку подлинности в LDAP.

## Листинг 5. Минимальная конфигурация для проверки подлинности Samba в LDAP

```
[global]
passdb backend = ldapsam:ldap://ldap.ertw.com
ldap suffix = dc=ertw,dc=com
ldap user suffix = ou=People
ldap group suffix = ou=Groups
ldap admin dn = uid=samba_service,ou=People,dc=ertw,dc=com
```

В листинге 5 задействована глобальная область параметров, а в качестве локального сервера указывается LDAP-сервер, расположенный по адресу ldap.ertw.com. Три суффикса сообщают Samba о базовом имени дерева и об именах ветвей для пользователей и групп, соответственно. Наконец, настраивается уникальное имя (DN) пользователя с правами администратора. Учетная запись этого пользователя будет использоваться для подключения к дереву при проверке подлинности других пользователей. В листинге 6 представлена информация об этом пользователе в формате обмена данными LDAP (LDAP Data Interchange Format, LDIF).

## Листинг 6. Данные учетной записи администратора в формате LDIF

```
dn: uid=samba_service,ou=People,dc=ertw,dc=com
uid: samba_service
objectclass: person
objectclass: uidobject
description: Service account to allow Samba to authenticate
cn: samba_service
sn: samba_service
userPassword: {SSHA}tQNdW/bNxQGz2iGoLz5zFL5wJ8px43v5
```

Необходимо задать пароль для DN администратора с помощью команды `smbpasswd -w` пароль. Хэш `userPassword` из листинга 6 является тем же самым паролем, но генерированным с помощью команды `slappasswd`. Теперь нужно перезапустить Samba.

### Управление пользователями

Как правило, необходимо позаботиться об управлении сопоставлениями между идентификаторами пользователей UNIX и идентификаторами безопасности Microsoft с учетом SID домена. Эта процедура может часто приводить к ошибкам, поэтому здесь нам поможет команда `smbldap-useradd`. Для добавления пользователя с именем `sean` запустите команду `smbldap-useradd -a sean`. Параметр `-a` говорит о том, что к объекту LDAP необходимо добавить классы Samba `objectClass`, которые позволяют пользователю подключаться к домену Microsoft. Наконец, с помощью команды `smbldap-passwd` с указанием имени пользователя для указанного пользователя задается пароль.

К этому моменту вы должны иметь возможность подключаться к серверу Samba, используя учетные данные, которые только что были добавлены в дерево LDAP. Пакет `smbldap-tools` содержит множество других инструментов, помогающих управлять учетными записями пользователей и планировать их более масштабные перемещения.

### Аутентификация UNIX в Samba

Linux и Microsoft хранят пароли по-разному, поэтому настройка *механизма единого входа* (single sign-on) или синхронизацию паролей между этими операционными системами – это

сложная процедура. Пакет `smbldap-tools` справляется с этой задачей, работая одновременно с двумя операционными системами. Другой способ добиться нужного результата – это передать управление паролями сетевому окружению Microsoft.

В Linux широко используется принцип *подключаемых модулей аутентификации* (Pluggable Authentication Modules, PAM). PAM-модули позволяют администраторам управлять проверкой подлинности служб и изменять способы аутентификации обычным редактированием конфигурационных файлов; при этом нет необходимости разбираться в том, как работают приложения. Для проверки подлинности пользователей приложения обращаются к библиотекам PAM. PAM-модули, в свою очередь, обращаются к конфигурационным файлам с тем, чтобы определить, каким образом необходимо выполнять проверку пользователей того или иного приложения. Результат аутентификации (успех либо отказ) возвращается приложению.

За последнее время было разработано множество модулей проверки подлинности, предоставляющих результаты другим службам – от локальных файлов паролей до LDAP, Kerberos и даже сетевого окружения Microsoft. Модуль, предоставляющий эту службу, называется `pam_smb`.

### Настройка `pam_smb`

Ваш дистрибутив должен содержать самую последнюю версию `pam_smb`; если это не так, то вы можете загрузить исходный код вручную (ссылки на дистрибутив есть в разделе [Ресурсы](#)). Конфигурация модуля находится в файле `/etc/pam_smb.conf`. Формат этого файла достаточно простой: сначала указывается имя домена, а затем – один или два сервера, выполняющих проверку подлинности. Вот пример простой конфигурации:

```
MYGROUP
ALICE
BOB
```

В этом примере аутентификация пользователей домена или рабочей группы MYGROUP выполняется серверами ALICE и BOB. Если аутентификация выполняется лишь одним сервером, то в конфигурационный файл будет содержать только две строки – имя домена и имя единственного сервера аутентификации.

Далее необходимо вставить модуль `pam_smb` в стек проверки подлинности. В директории `/etc/pam.d` есть несколько конфигурационных файлов, относящихся к определенному способу проверки подлинности для отдельной службы. Большинство этих файлов похожи, поскольку они часто включают в себя общий файл, а не дублируют одну и ту же конфигурацию. Найдите имя файла, которое указано в директиве `include`. Например, в Fedora этот файл будет называться или `password-auth`, или `system-auth`. В листинге 7 показан раздел файла `password-auth`, относящийся к проверке подлинности.

### Листинг 7. Раздел файла `password-auth`, отвечающий за проверку подлинности

```
auth      required      pam_env.so
# Use samba authentication
auth      sufficient    pam_smb_auth.so debug
auth      sufficient    pam_unix.so nullok try_first_pass
auth      requisite     pam_succeed_if.so uid >= 500 quiet
auth      required      pam_deny.so
```

Все строки листинга 7 обрабатываются поочередно. В первой строке выполняется вызов

модуля  `pam_env`, который устанавливает переменные окружения. Во второй строке модуль `pam_smb` пытается выполнить запрос проверки подлинности в режиме расширенной отладки. Если аутентификация завершается успешно, то проверка считается выполненной, и пользователь подключается. В противном случае управление передается модулю `pam_unix`, который проверяет учетные данные в файле паролей UNIX.

Экспериментируя с PAM, всегда оставляйте открытый сеанс пользователя `root` и держите под рукой рабочие копии нужных файлов. Если вы заблокируете себе доступ к системе, то сможете быстро восстановить его, скопировав рабочие копии вместо неправильных файлов.

## Службы имен

В Linux есть файл `/etc/nsswitch.conf`, управляющий тем, как системные библиотеки сопоставляют идентификаторы пользователей и групп с их именами. Этот файл содержит конфигурацию службы Name Service Switch (NSS). Большинство операционных систем содержат ссылки на локальные файлы (например, `/etc/passwd` и `/etc/group`), LDAP- или NIS-сервер. Можно также использовать службу Winbind, обеспечив перенаправление запросов системных библиотек в сетевое окружение Microsoft.

В следующей статье этой серии данная тема будет рассмотрена более подробно. Сейчас просто достаточно понимать, что проверка подлинности пользователей выполняется через PAM, но имена пользователей и групп проверяются системой NSS.

## Списки контроля доступов

Серверы Microsoft поддерживают достаточно мощный набор прав доступа, позволяющих администраторам управлять доступом к файлам и директориям с большой точностью. Даже не смотря на то, что некоторые UNIX-системы имеют поддержку списков контроля доступов (ACL) для файловой системы, эта поддержка слабо распространена. Фактически, Linux-системы часто ограничены традиционной поддержкой битов Чтение/Запись/Выполнение для пользователя группы и всех остальных. Тем не менее, Samba должна обеспечивать интерфейс ACL для клиентов Microsoft и сопоставлять его с файловыми разрешениями UNIX; при этом некоторая информация может храниться в TDB-файле.

Сопоставлением файловых разрешений UNIX и Windows NT управляют несколько параметров. Двумя важными параметрами являются `force security mode` и `security mask`. Эти параметры работают совместно и предназначены для установки и удаления битов прав доступа к файлам, соответственно.

## Главное о правах доступа к файлам

Права доступа к файлам в UNIX представляют собой восьмеричные числа: 1 – выполнение (execute), 2 – запись (write) и 4 – чтение (read). Эти числа соответствуют трем битам, необходимым для построения восьмеричного числа. Первое из трех восьмеричных чисел определяет права доступа для владельца файла или директории, второе число – права доступа для группы и третье – глобальные разрешения (для всех остальных пользователей). Если для файла установлено значение 750, то это означает, что владелец файла может читать, записывать и запускать его на выполнение, члены группы-владельца – только читать и выполнять, а всем остальным доступ к файлу запрещен.

Для установки и сброса прав доступа можно выполнять над восьмеричными значениями бинарные операции. Операция `OR` устанавливает действующие биты, а операция `AND` очищает бит. Результатом операции `1 OR 4` будет 5, поскольку изначально установлен бит 1, а затем мы устанавливаем бит 4. Аналогично, результатом операции `5 OR 4` будет 5, поскольку 5 – это комбинация битов 1 и 4, поэтому установка бита, который уже и так установлен, не приводит к каким-либо результатам.

Операция **AND** противоположна операции **OR**. Для того, чтобы иметь в результате операции **AND** установленный бит, необходимо, чтобы он был установлен в обоих ее operandах. Результатом операции **1 AND 1** является **1**, поскольку установлены оба бита **1**. Результатом операции **5 AND 1** также является **1**, поскольку бит **1** установлен в обоих operandах, но бит **4** установлен только в одном из них (в левом). Если аккуратно пользоваться операциями **AND** и **OR**, то биты всегда будут правильно установлены или сброшены.

## Применение операций с битами к Samba

Параметр **force security mode** принудительно устанавливает биты для файла при его создании или когда клиент пытается изменить разрешения для него. По умолчанию этот параметр имеет значение **000**, что означает, что биты принудительно не устанавливаются. Если изменить значение параметра **force security mode** на **700**, то для пользователя всегда будут принудительно устанавливаться биты чтения, записи и выполнения, что позволит ему всегда прочитать свои файлы, даже если он попытается удалить разрешения.

Аналогично, параметр **security mask** выполняет операцию **AND** и сбрасывает биты. По умолчанию этот параметр имеет значение **777**, что означает, что никакие биты не сбрасываются. Значение параметра **775** будет сбрасывать бит записи файла для всех остальных пользователей (**Everyone**), что не позволит пользователям создавать файлы, доступные для записи каждому.

Параметры **security mask** и **force security mode** можно устанавливать как глобально, так и на уровне общего ресурса. Наиболее полезными эти параметры оказываются при работе с публичными директориями и директориями для групп пользователей, когда необходимо автоматически устанавливать нужные групповые или глобальные права доступа к файлам без вмешательства пользователей.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Authentication and authorization](#) (EN).
- Чтобы освежить свои знания о TDB-файлах, прочтите статью "[Изучаем Linux, 302 \(смешанные среды\): файлы базы данных Trivial Database](#)" (developerWorks, март 2011 г.)
- Если вы подзабыли о принципах устройства LDAP, прочтите [руководства developerWorks для подготовки к экзамену LPI 301](#) (EN).
- Узнайте больше о функции [umask](#) (EN), которая ведет себя так же, как и параметр **force security mask**. Этот материал поможет вам получить навыки в использовании двоичной математики.
- Если вы большой энтузиаст, загрузите и разберите несколько [примеров конфигураций сервера OpenLDAP](#) (EN).
- Если вы не знакомы с PAM, то вам будет полезно прочитать [руководство администратора системы Linux-PAM](#) (EN).
- Более подробно [управление доступом](#) (EN) рассматривается в документации Samba.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).

- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.

#### Получить продукты и технологии

- [smbldap-tools](#) (EN) – пакет, сильно упрощающий интеграцию Samba с LDAP.
- [pam\\_smb](#) (EN) – модуль, позволяющий подключаться к серверу Linux с использованием учетных данных Microsoft.
- Загрузите [последнюю версию Samba](#) (EN) и используйте все ее самые новые возможности.

## Изучаем Linux, 302 (смешанные среды): Winbind

*Использование контроллера домена Windows для управления учетными записями Linux*

[Майк Бойерсмит](#), Штатный инженер-программист, IBM

**Описание:** Если в вашей сети есть контроллер домена под управлением Windows или сервера Samba, то базу данных контроллера домена можно использовать вместо или в дополнение к локальной базе данных учетных записей на компьютере Linux . Для этого необходимо использовать набор инструментов под названием *Winbind*. Этот инструментарий будет полезен даже на тех компьютерах Linux, на которых не запущен сервер Samba или отсутствуют общие файлы или принтеры – в этом случае пользователи с учетными записями домена Windows могут подключаться к консоли или использовать SSH для доступа к различным функциям, присущим исключительно Linux, использовать почтовые серверы POP или IMAP, авторизуясь с учетными данными Windows, и т. д.

**Дата:** 17.07.2012

**Уровень сложности:** средний

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

### Краткий обзор

В этой статье рассматриваются следующие темы:

- Инсталляция Winbind

- Конфигурирование Winbind

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 313.3 темы 313. Цель имеет вес 2.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и имеет общее представление о Samba (см. предыдущую статью [Изучаем Linux, 302: основные принципы](#)), а также основы ее настройки, включая настройку Samba на использование контроллера домена. Вы должны знать общую структуру конфигурационного файла smb.conf и иметь представление о работе серверов, включая использование сценариев запуска System V (SysV) и супер-серверов. Для выполнения примеров этой статьи необходимо иметь доступ к домену Windows (можно работать в домене как под управлением Samba, так и под управлением операционной системы Windows Server).

## Что такое Winbind

Как упоминалось в статьях [Изучаем Linux, 302 \(смешанные среды\): роли Samba](#) и [Изучаем Linux, 302 \(смешанные среды\): управление доменом](#), в сетевом окружении Windows часто присутствует *контроллер домена* – компьютер, выполняющий аутентификацию всех компьютеров Windows в сети. Такая конфигурация существенно облегчает работу в сети, содержащей большое количество пользователей и компьютеров, поскольку в этом случае нет необходимости дублировать (и поддерживать в актуальном состоянии) информацию об учетных записях на разных компьютерах. Вместо этого вся информация об учетных записях Windows хранится на одном компьютере, позволяя администраторам централизованно управлять учетными данными, а пользователям – единожды менять свои пароли и использовать их на любых рабочих станциях домена.

С помощью различных опций сервер Samba можно настроить таким образом, чтобы он являлся рядовым сервером домена. Типичный набор глобальных опций для присоединения к домену может выглядеть следующим образом:

```
security = Domain
password server = CONTROL
domain logons = No
encrypt passwords = Yes
```

Если контроллер домена поддерживает функциональность служб Microsoft Active Directory Domain Services, то вместо опции **security = Domain** можно использовать опцию **security = ADS**. Кроме того, сервер Samba необходимо присоединить к домену с помощью команды **net join member -U adminuser**, где **adminuser** – учетная запись администратора домена. За более подробным описанием процесса настройки сервера Samba в качестве члена домена обратитесь к статье [Изучаем Linux, 302 \(смешанные среды\): роли Samba](#).

Все это хорошо подходит для сервера Samba, но, конечно, это не единственный сервер, который можно запустить на компьютере Linux. Другие серверы, а также локальные службы входа, теоретически могут получать преимущества от использования контроллера домена. Например, если некоторые компьютеры в сети являются рабочими станциями Linux, или если вы запускаете сервер электронной почты, работающий по протоколу Post Office Protocol (POP), на компьютере Linux, то для аутентификации удобнее использовать контроллер домена. Здесь в дело вступает Winbind, позволяющий пользователям и компьютерам Linux использовать учетные записи домена Windows для выполнения задач аутентификации,

отличных от задач Samba.

Однако прежде, чем приступать к настройке Winbind, следует знать о его ограничениях. Поскольку аутентификация в доменах Microsoft ориентирована прежде всего на Windows, то она не выполняет некоторых функций, необходимых для работы с учетными записями Linux, например, не присваивает значения идентификаторам пользователей и групп (UID и GID) UNIX®. Вместо этого в доменах Windows используются идентификаторы безопасности, которые нельзя напрямую сопоставить идентификаторам пользователей и групп Linux. Аналогично, контроллеры домена не хранят информацию о домашних директориях UNIX/Linux. Таким образом, Winbind должен генерировать часть этой информации локально. Этот способ подходит для многих ситуаций, однако, он означает, что на двух компьютерах Linux, использующих Winbind, для одного и того же пользователя могут быть запросто созданы различные идентификаторы UID и GID. Если данные компьютеры совместно работают с файлами, расположеннымными в сетевой файловой системе NFS, которая предполагает использование одинаковых UID и GID на всех серверах, то это может привести к негативным последствиям. В таких ситуациях лучше будет настроить в сетевом окружении сервер LDAP и использовать его в качестве хранилища как учетных записей Linux, так и доменных учетных записей Windows.

Winbind использует несколько различных компонентов:

- Опции файла smb.conf, относящиеся к Winbind.
- Опции конфигурации подсистемы подключаемых модулей аутентификации PAM, которая является стандартной частью современных дистрибутивов Linux.
- Опции конфигурации подсистемы Net Service Switch (NSS), которая является стандартной частью современных дистрибутивов Linux.
- Серверная программа `winbindd`.

Таким образом, для настройки Winbind необходимо внести изменения в каждую из вышеперечисленных конфигураций. Для использования Winbind не обязательно запускать демоны Samba `smbd` и `nmbd`, однако, возможно, потребуется инсталлировать Samba, чтобы получить все необходимые вспомогательные файлы. Может потребоваться инсталлировать отдельный пакет Winbind, который обычно называется `winbind` или `samba-winbind`. Также может потребоваться установить пакет `samba-winbind-clients`. Убедитесь, что после инсталляции всех необходимых пакетов в вашей системе присутствуют файлы `/lib/security/pam_winbind.so` и `/lib/libnss_winbind.so.2`.

### Настройка опций Winbind в файле smb.conf

Для настройки Winbind сначала необходимо присоединить компьютер к домену; об этом кратко упоминалось в этой статье, а более подробно этот материал изложен в статье [Изучаем Linux, 302 \(смешанные среды\): роли Samba](#). После этого необходимо настроить в файле smb.conf ряд опций, относящихся к Winbind.

### Обзор опций Winbind

Ниже перечислены некоторые опции smb.conf, которые, возможно, потребуется настроить:

- **winbind separator.** В домене Windows имя учетной записи пользователя состоит из имени домена и имени пользователя, разделенных определенным символом. Параметр `winbind separator` устанавливает этот символ. По умолчанию используется обратная косая черта (\), а распространенной альтернативой является знак плюс (+).
- **winbind cache time.** Winbind кэширует данные аутентификации на определенный промежуток времени, по умолчанию равный 300 секундам (т. е. 5 минутам). При проверке конфигурации Winbind можно уменьшить это значение.
- **template shell.** Эта опция устанавливает командную оболочку пользователя. По

умолчанию используется значение `/bin/false`, что подходит для систем, не поддерживающих доступ к командной оболочке. Если же вы хотите предоставить пользователям возможность подключаться (локально или удаленно, например, через SSH) к системе, то задайте для этой опции значение `/bin/bash` или имя любой другой командной оболочки Linux.

- **template homedir**. Для каждого пользователя должна быть указана домашняя директория по умолчанию. Обычно для этого используется одна или несколько переменных окружения Samba, например, `%U` (имя пользователя) и `%D` (имя домена). По умолчанию опция `template shell` имеет значение `/home/%D/%U`.
- **winbind enum users**. Эта логическая опция включает или отключает поддержку определенных системных вызовов, позволяющих приложениям перечислять пользователей. По умолчанию она установлена в `Yes`; ее установка в `No` может повысить производительность, однако некоторые программы, например `finger`, будут работать некорректно.
- **winbind enum groups**. Эта опция работает так же, как и опция `winbind enum users`, но применяется не к пользователям, а к группам.
- **winbind use default domain**. Если эта опция установлена в `Yes`, то Winbind удаляет из имен пользователей доменную часть, что зачастую является предпочтительным действием, поскольку имена становятся короче (например, `rexx` вместо `MYDOMAIN\rexx`). Использовать значение по умолчанию `No` имеет смысл в тех случаях, когда вы работаете с несколькими доменами.
- **idmap uid**. Значением этой опции является диапазон идентификаторов пользователей (UID), разделенных знаком тире (-), например `10000-20000`. Убедитесь, что указанный диапазон не перекрывается с диапазоном локальных идентификаторов UID, определенных в системе.
- **idmap gid**. Эта опция работает так же, как и опция `idmap uid`, но определяет не идентификаторы пользователей (UIDs), а идентификаторы групп (GIDs).

## Разбор простого примера

В качестве примера рассмотрим листинг 1, содержащий ряд описанных выше опций файла `smb.conf`.

### Листинг 1. Фрагмент файла `smb.conf`, содержащего конфигурацию Winbind

```
winbind separator = +
winbind cache time = 60
template shell = /bin/bash
template homedir = /home/%U
winbind enum users = Yes
winbind enum groups = Yes
winbind use default domain = Yes
idmap uid = 10000-20000
idmap gid = 10000-20000
```

Конечно, опции вашей конфигурации будут отличаться от приведенной. Например, как было отмечено ранее, для серверов, не предоставляющих доступ к командной оболочке (например, для серверов электронной почты или FTP-серверов), больше подойдет значение опции `template shell`, определенное по умолчанию.

## Настройка PAM

После того, как вы настроили параметры в файле smb.conf, необходимо разобраться с конфигурацией PAM. Мы уже рассказывали о PAM в статье [Изучаем Linux, 302 \(смешанные среды\): проверка подлинности и авторизация](#), но для утилит, не связанных с Samba, есть смысл добавить в качестве инструмента проверки подлинности Winbind, а не проверять пользователей Samba с помощью PAM. Эта процедура имеет свои тонкости, поскольку в различных дистрибутивах PAM настраивается по-разному, поэтому изменения, прекрасно работающие в одном дистрибутиве могут совершенно не работать в другом.

### Что представляет собой PAM

PAM – это набор библиотек, которые могут использоваться приложениями, требующими проверки подлинности, например, программами `login` (управляет входом в систему в текстовом режиме), X Display Manager (управляет входом в систему в графическом режиме) или POP3-почтовым сервером. Для создания более гибкой системы PAM можно настраивать с помощью конфигурационных файлов, о чем будет рассказано ниже.

PAM – это сложная система, а поскольку в различных дистрибутивах она имеет разные конфигурации, и их невозможно полностью описать в этой статье (ссылки на дополнительную документацию PAM вы можете найти в разделе [Ресурсы](#)). Конфигурация PAM настраивается в файлах директории /etc/pam.d, в большинстве из которых описывается, как работает PAM с определенными программами, например, с программой `login` или сервером SSH. Тем не менее, большинство дистрибутивов содержит глобальные конфигурационные файлы PAM, такие как `system-auth` или `common-auth`, где `имястека` – имя одного из четырех стеков PAM (каждый стек соответствует определенному типу действий, выполняемых PAM).

### Изменение стека PAM

Если вы собираетесь изменять конфигурацию PAM, то необходимо определиться с тем, должны ли вносимые изменения влиять на все службы входа или только на некоторые из них (чтобы узнать, службы входа каких типов установлены в вашей системе, просмотрите файлы в директории /etc/pam.d, чтобы выяснить, все ли они должны использовать Winbind). Если необходимо изменить только одну или две службы (например, POP3-сервер) и не использовать Winbind для остальных служб (например, для FTP-сервера), то редактируйте только файл для той службы, которую нужно изменить. Если же необходимо применить изменения ко всем службам, то редактируйте общий файл, например, `system-auth`.

Типичный стек PAM выглядит примерно так, как показано в листинге 2 (это файл /etc/pam.d/common-auth из дистрибутива Ubuntu 10.10).

### Листинг 2. Пример стека PAM

```
auth  [success=1 default=ignore]  pam_unix.so nullok_secure
auth  requisite                  pam_deny.so
auth  required                   pam_permit.so
```

К сожалению, синтаксис этого примера не очень понятен. Модуль `pam_unix.so` управляет аутентификацией, используя файлы локальной базы данных паролей, а параметр `success=1` в этой же строке означает, что PAM пропускает одну строку, когда этот модуль возвращает сообщение об успешной аутентификации. Таким образом, модуль `pam_deny.so` (который всегда возвращает код ошибки аутентификации) всегда пропускается в случае успешного выполнения модуля `pam_unix.so`, и вход в систему выполняется успешно.

Чтобы изменить эту конфигурацию и внедрить использование Winbind, необходимо добавить ссылку на модуль `pam_winbind.so` и изменить количество пропускаемых строк в строке `pam_unix.so`. Конечная конфигурация приведена в листинге 3 (изменения выделены жирным шрифтом).

### Листинг 3. Стек PAM с поддержкой Winbind

```
auth  [success=2 default=ignore]  pam_unix.so nullok_secure
auth  [success=1 default=ignore]  pam_winbind.so cached_login try_first_pass
auth  requisite                  pam_deny.so
auth  required                   pam_permit.so
```

В дистрибутивах, отличных от Ubuntu, необходимо внести в конфигурацию PAM другие изменения, но перед этим нужно полностью разобраться в исходной конфигурации. Учтите также, что нужно изменить все четыре стека – `auth`, `account`, `session` и `password`. Все эти стеки могут находиться в одном файле либо быть разнесены по нескольким. Если вам необходимо изменить отдельные службы, то в конфигурационных файлах каждой из них необходимо изменить все четыре стека.

В некоторых дистрибутивах необходимые изменения в конфигурации PAM вносятся при установке пакета Winbind. Таким образом, вам вообще не нужно вносить никаких изменений вручную.

### Настройка NSS

Вторая служба, которую необходимо настроить для включения аутентификации с помощью Winbind, это NSS. К счастью, конфигурация NSS значительно проще, чем конфигурация PAM. Необходимо изменить следующие три строки в файле `/etc/nsswitch.conf`:

```
passwd:      compat
group:       compat
shadow:      compat
```

В некоторых системах вместо слова `compat` в этих трех строках содержится слово `files` и, возможно, дополнительные элементы. Эти строки указывают NSS на использование в процессе работы локальных файлов, содержащих списки пользователей и групп для программ, которым это необходимо. Чтобы добавить в систему Winbind, добавьте ссылку на него в каждой из этих трех строк:

```
passwd:      compat winbind
group:       compat winbind
shadow:      compat winbind
```

### Запуск Winbind

После того, как все необходимые изменения внесены, можно, наконец, запускать и проверять работу Winbind. Обычно Winbind запускается с помощью сценария запуска SysV, как показано ниже:

```
# /etc/init.d/winbind start
```

**Предупреждение.** Если на вашем компьютере работает демон Name Service Cache Daemon

(NSCD), то завершите его работу, прежде чем запускать демон Winbind. NSCD может вмешиваться и нарушать нормальную работу Winbind.

Если вы хотите запускать демон автоматически каждый раз при загрузке операционной системы, то убедитесь, что его сценарий запуска SysV правильно настроен. Для этого можно использовать утилиты `chkconfig`, `ntsysv`, `update-rc.d` и другие инструменты.

Если Winbind запущен, то его работу можно проверить с помощью команды `wbinfo`. Опция `-U` выводит список пользователей:

```
$ wbinfo -u
grogers
fastaire
mikhail
```

В этом примере мы видим трех пользователей. Для получения списка пользователей можно также использовать команду `getent passwd`, по существу, являющуюся эквивалентом файла /etc/passwd. Эта команда показывает всех пользователей системы, зарегистрированных как локально, так и с помощью Winbind и других служб, которые могут использоваться системой, например, LDAP.

**Примечание.** Иногда командам `wbinfo -u` и `getent passwd` не удается возвратить полный список пользователей домена, особенно если задана опция `winbind enum users = No`. Если эти команды не работают, то проверьте, удается ли вам сделать `login`.

Конечно, полная проверка работоспособности Winbind заключается в том, чтобы проверить его способность правильно аутентифицировать пользователей служб, настроенных на использование PAM. Таким образом, следует выполнять проверку, используя учетную запись домена Windows, а не учетную запись локального компьютера. Если проверка не прошла, попробуйте просмотреть файлы журналов как на клиенте, так и на контроллере домена Windows. Помните о том, что если задан параметр `winbind use default domain = No`, то локальные имена пользователей будут иметь вид `ДОМЕН\имяпользователя` (или нечто похожее с использованием разделителя, указанного в параметре `winbind separator`).

## Что дальше

Следующая статья этой серии [Изучаем Linux, 302 \(смешанные среды\): интеграция с протоколом CIFS](#) содержит материалы цели 314.1 темы 314. В ней рассматривается интеграция компьютеров Linux в сеть Server Message Block (SMB)/Common Internet File System (CIFS) в качестве клиентов общих файловых ресурсов. Вы узнаете об использовании инструментария автономных клиентов, а также о монтировании общих ресурсов SMB/CIFS в иерархии стандартной файловой системы Linux.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Winbind](#) (EN).
- В статье "[Understanding and configuring PAM](#)" (EN) (developerWorks, март 2009 г.) PAM рассматривается более детально.
- В [официальной документации Samba Winbind](#) (EN) вы найдете дополнительные сведения о настройке Winbind.
- Руководство [Linux PAM System Administrators' Guide](#) (EN) содержит подробную информацию о PAM, хотя оно было написано достаточно давно, поэтому в нем отсутствуют сведения о новейших возможностях PAM.

- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.
- В [разделе Linux сайта developerWorks](#) можно найти сотни [пошаговых инструкций и руководств](#), загрузить программные продукты, а также получить ссылки на форумы и многие другие ресурсы, ориентированные на разработчиков и администраторов Linux.

# Изучаем Linux, 302 (смешанные среды): Интеграция с протоколом CIFS

*Использование Linux в качестве клиента серверов SMB/CIFS*

Родерик Смит (Roderick Smith), автор и консультант, IBM

**Описание:** Компьютеры под управлением Linux могут работать в Windows-сетях как серверы и/или клиенты. Можно использовать `ftp` и аналогичные программы для передачи файлов и модификации сервера или смонтировать на компьютере Linux общий ресурс Samba или Windows-сервера, позволив обычным приложениям напрямую обращаться к файлам на сервере. Выполняя такие действия, не забывайте об особенностях оригинального протокола SMB и его новой модификации CIFS, в особенности при обращении к компьютерам под управлением Windows Server, поскольку вам могут быть недоступны некоторые функции файловой системы, поддерживаемые компьютерами Linux.

**Дата:** 26.07.2012

**Уровень сложности:** средний

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Протоколы Server Message Block (SMB) и Common Internet File System (CIFS).
- Возможности и преимущества использования CIFS.
- Монтирование общих файловых ресурсов CIFS на клиентах Linux.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 314.1 темы 314. Цель имеет вес 3.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux, знает основы конфигурирования Samba и имеет общее представление о структуре конфигурационного файла `smb.conf`. Необходимо также знать основы монтирования локальных и удаленных файловых систем (с помощью команды `mount` и файла `/etc/fstab`). Знания о команде `ftp`, входящий в стандартный набор текстовых команд Linux приветствуются, хотя не являются обязательными.

## Что такое SMB/CIFS

Прежде чем переходить к рассказу о том, как использовать Linux в качестве клиента сервера SMB/CIFS, полезно рассказать об особенностях этих протоколов и выяснить, насколько полно они обеспечивают использование файловой системы при работе с Linux . Мы объясним, как работает изучение оригинальный протокол SMB и какие новые функции реализованы в его модификации CIFS . Вы можете обратиться к статье developerWorks, содержащей материалы цели 310.1 экзамена LPI, в которой рассматриваются некоторые основные принципы SMB/CIFS (см. ссылку в разделе [Ресурсы](#)).

### Основные возможности SMB

SMB обладает несколькими уникальными возможностями с точки зрения работы в сети, включая собственную систему именования компьютеров (Network Basic Input/Output System, NetBIOS), рабочие группы и протоколы аутентификации. Для того чтобы понять, как SMB и CIFS работают с Linux-клиентами общих файловых ресурсов, нужно рассказать о наиболее важной функции этих протоколов, а именно, о наборе предоставляемых ими метаданных.

*Метаданные* – это данные, связанные с файлом, но не являющиеся его частью. Примером метаданных являются метка времени, владелец, права доступа и даже имя файла.

Несомненно, вы знаете о некоторых способах использования метаданных на компьютерах Linux и, возможно, об их некоторых отличиях в Linux и других операционных системах, например, Windows. Поскольку протокол SMB был разработан для DOS, Windows и IBM Operating System/2® (OS/2), то он содержит много метаданных, специфичных для этих операционных систем. Однако более важно то, что SMB не поддерживает такие метаданные UNIX® и Linux, как владельцы, группы и большинство прав доступа. Кроме того, SMB не поддерживает символические и жесткие ссылки, а также другие специальные типы файлов, такие как файлы устройств. SMB содержит несколько типов метаданных, не распознаваемых Linux в обычном режиме, например, биты *hidden* (скрытый) , *archive* (архивный) и *system* (системный). Бит Read-only (только чтение) можно сопоставить биту разрешений Write (запись) в Linux.

Для того чтобы клиенты Linux могли работать с SMB с учетом вышеперечисленных особенностей, они должны либо игнорировать их, либо иметь возможность использовать "поддельные" данные. Эти возможности похожи на те, что используются при монтировании в Linux файловых систем NTFS или FAT. К счастью, протокол CIFS предоставляет более широкий набор инструментов, позволяющих обходить некоторые из этих ограничений.

Необходимо также знать и о сетевых портах, используемых протоколом SMB. Это UDP-порты (User Datagram Protocol) 137 и 138, а также TCP-порт 139 (службы сеансов – другими словами, передача файлов). Эта информация потребуется при отладке SMB с помощью низкоуровневых утилит диагностики сети.

### Расширения CIFS для протокола SMB

В середине 1990-х годов компания Microsoft® изменила название протокола SMB на *CIFS* и одновременно добавила в него ряд новых возможностей, включая поддержку символьических и жестких ссылок, а также файлов большого объема. Также CIFS поддерживает доступ к серверу по защищенному TCP-порту 445 в дополнение к стандартному порту 139.

Не менее важными, чем собственные расширения Microsoft для SMB, оказались и другие расширения CIFS. В частности, ряд функциональных возможностей, известный как *UNIX extensions* (расширения UNIX), обеспечивает поддержку владельцев и прав доступа к файлам наряду с другими типами метаданных UNIX. Если и клиенты, и сервер поддерживают эти расширения, то использование протокола CIFS вместо протокола SMB может обеспечить намного более эффективную работу клиентов под управлением Linux. Как и можно было ожидать, эти расширения не поддерживаются операционными системами семейства Windows

Server®, поэтому полезны они только тогда, когда клиенты Linux подключаются к серверу Samba. Сервер должен быть настроен с помощью следующего глобального параметра:

```
unix extensions = Yes
```

По умолчанию этот параметр был установлен в **No** во всех версиях Samba, меньше 3.0, но в Samba 3.0 по умолчанию он установлен в **Yes**, что избавляет от необходимости настраивать его вручную.

## Использование smbclient

Возможно, самый простой способ получить доступ к серверу SMB/CIFS с клиента Linux – это использовать утилиту командной строки **smbclient**. Эта утилита похожа на классическую команду **ftp**, поэтому если вы знакомы с **ftp**, то вы легко освоите и **smbclient**. Если вы не знакомы с **ftp**, то достаточно знать, что эта программа обеспечивает соединение с сервером *без* стандартного монтирования общих файловых ресурсов. Вместо этого для просмотра, удаления, загрузки или передачи файлов пользователь выполняет различные команды.

Для использования **Smbclient** необходимо набрать в командной строке имя этой команды и имя службы в следующем формате: *// СЕРВЕР / СЛУЖБА*. Например, если необходимо получить доступ к общему ресурсу GORDON на сервере TANGO, то следует указать имя *//TANGO/GORDON*. В зависимости от конфигурации сервера может потребоваться ввести пароль. Если введен правильный пароль, то можно вводить различные команды для доступа к файлам, хранящимся на сервере. В таблице 1 перечислены некоторые наиболее важные команды **smbclient**; для получения информации о других более экзотических командах обратитесь к тан-странице этой утилиты.

**Таблица 1. Наиболее важные команды smbclient**

Команда	Действие
? или help	Выводит список всех команд
cd	Изменяет рабочую директорию на удаленном сервере
del	Удаляет файл
dir или ls	Выводит список файлов в текущей (или указанной) директории
exit или quit	Завершает сеанс работы
get	Передает файл с сервера клиенту
lcd	Изменяет рабочую директорию на локальном компьютере
md или mkdir	Создает директорию на удаленном сервере
mget	Передает несколько файлов с сервера клиенту
more	Выводит список удаленных файлов с помощью локальной команды постраничного вывода
mput	Передает несколько файлов с клиента на удаленный сервер
put	Передает файл с клиента на удаленный сервер
rd или rmdir	Удаляет директорию
rename	Переименовывает файл на удаленном сервере
rm	Удаляет один или несколько файлов на удаленном сервере

По умолчанию для подключения к серверу **smbclient** использует текущее имя пользователя, однако можно указать имя явно с помощью опции **-U**. На самом деле, можно использовать несколько опций командной строки, включая опции, позволяющие передавать файлы без входа в интерактивный режим **smbclient**. Таким образом, **smbclient** можно использовать в сценариях для выполнения автоматической передачи файлов. За

дополнительной информацией обратитесь к тан-странице этой утилиты.

Сеанс работы с `smbclient` выглядит примерно следующим образом:

### Листинг 1. Пример сеанса работы с `smbclient`

```
$ smbclient //TANGO/GORDON/
Enter gordon's password:
Domain=[RINGWORLD] OS=[Unix] Server=[Samba 3.4.12]
smb: \> cd mystuff
smb: \mystuff\> ls
.
..
xv-3.10a-1228.1.src.rpm      D      0  Mon May 16 19:20:08 2011
License.txt                  D      0  Mon May 16 19:18:12 2011
xorg.conf                     3441259 Tue May 18 19:09:26 2010
                             27898 Mon May 16 19:17:15 2011
                             1210  Fri Jan 21 04:18:13 2011

      51198 blocks of size 2097152. 2666 blocks available
smb: \mystuff\> get xorg.conf
getting file \mystuff\xorg.conf of size 1210 as xorg.conf (9.4 KiloBytes/sec)
(average 9.4 KiloBytes/sec)
smb: \mystuff\> exit
```

**Совет.** Утилита `smbclient` является превосходным инструментом отладки. Не смотря на свою простоту, она позволяет получить доступ к сетевому окружению без монтирования ресурсов, что упрощает поиск и устранение проблем.

### Монтирование файловых ресурсов SMB/CIFS

Не смотря на всю свою эффективность, `smbclient` не позволяет получить такой же прозрачный доступ к серверу, как при работе с Windows-клиентом. Если вам необходим именно такой доступ, то необходимо использовать другие средства, позволяющие монтировать общие ресурсы SMB/CIFS. Это можно сделать с помощью стандартной команды Linux `mount` или редактируя файл `/etc/fstab` для автоматического монтирования ресурсов SMB/CIFS при загрузке компьютера.

### Временное монтирование общих ресурсов

Файловый ресурс SMB/CIFS можно смонтировать с помощью команды `mount`, которая также используется для монтирования локальных томов или совместно используемых ресурсов NFS. Можно указать тип файловой системы `cifs` или в большинстве случаев `mount` определит необходимость использования того или иного драйвера на основе синтаксиса команды. Кроме того, можно напрямую вызвать вспомогательную программу `mount.cifs`. По сути, монтирование локальной и удаленной файловой системы отличается лишь типом монтируемого устройства; таким образом, для монтирования ресурса GORDON, расположенного на сервере TANGO, достаточно выполнить от имени пользователя `root` следующую команду:

```
# mount //TANGO/GORDON /mnt
```

На практике такая команда может создать проблему: в качестве имени пользователя она передает на сервер имя `root`, и если этому пользователю не разрешено подключаться к серверу, то монтирование завершится с ошибкой. Эту проблему можно исправить, используя опцию `-o user=имя` для передачи имени пользователя на сервер.

```
# mount -o user=gordon //TANGO/GORDON /mnt
```

Password:

Можно использовать несколько других опций монтирования, передаваемых команде `mount` с помощью опции `-o`. Наиболее полезные из них перечислены в таблице 2. За дополнительной информацией об остальных опциях обратитесь [man-странице `mount.cifs`](#).

**Таблица 2. Наиболее важные опции `mount.cifs`**

Опция	Действие
<code>user=name</code> or <code>username=name</code>	Определяет имя пользователя, передаваемое на сервер.
<code>password=pass</code>	Определяет пароль для передачи на сервер. Если пароль не указан, то <code>mount.cifs</code> использует значение переменной окружения <code>PASSWD</code> ; если значение <code>PASSWD</code> не задано, то программа запрашивает пароль у пользователя.
<code>credentials=file</code> <code>ename</code>	Определяет файл, содержащий имя пользователя, пароль и необязательное имя рабочей группы. Каждое значение указывается в отдельной строке, начинающейся с <code>username=</code> , <code>password=</code> и <code>workgroup=</code> , соответственно.
<code>uid=UID</code>	Определяет идентификатор (ID) пользователя, который будет являться владельцем файлов смонтированного ресурса.
<code>gid=GID</code>	Аналогична опции <code>uid=UID</code> , но применяется к идентификаторам групп (GID), а не к идентификаторам пользователей (UID).
<code>file_mode=mode</code>	Устанавливает режим файлов (разрешения) в числовой форме, который будет назначен файлам на сервере.
<code>dir_mode=mode</code>	Аналогична опции <code>file_mode=mode</code> , но применяется не к файлам, а к директориям.
<code>guest</code>	Предотвращает запросы на ввод пароля. Обычно эта опция работает только в том случае, если для ресурса поддерживается гостевой доступ. Если сервер становится недоступен, то процессы, пытающиеся получить доступ к расположенным на нем файлам, остаются в зависшем состоянии до тех пор, пока доступ к серверу не будет возобновлен.
<code>hard</code>	Если сервер становится недоступен, то процессы, пытающиеся получить доступ к расположенным на нем файлам, получают сообщения об ошибках. Это действие используется по умолчанию.
<code>soft</code>	Параметры <code>uid</code> , <code>gid</code> , <code>file_mode</code> и <code>dir_mode</code> обычно не являются обязательными при подключении к серверу с поддержкой расширений UNIX, реализованных в CIFS. Тем не менее, эти параметры можно использовать для переопределения значений, установленных на сервере. Также отметим, что все эти параметры влияют на то, как файлы видны <i>клиенту</i> ; они не влияют на разрешения и права владения файлами на сервере.

После того, как общий ресурс SMB/CIFS смонтирован, можно получить к нему доступ точно так же, как и к локальному диску или тому NFS. Можно копировать и удалять файлы командами `Cp` и `Rm`, редактировать их в текстовых редакторах или других программах и т. д. Однако помните о том, что если сервер не поддерживает определенные возможности, то вы не сможете их использовать. Например, невозможно изменить режим файла с помощью `chmod`, если сервер не поддерживает расширения UNIX (частным исключением в случае с `chmod` является возможность изменения разрешений на запись – эти разрешения инверсно сопоставлены биту "только чтение" протокола SMB).

Когда работа с общим ресурсом закончена, его можно размонтировать с помощью команды

`umount`, как обычную локальную файловую систему:

```
# umount /mnt
```

## Монтирование файловых ресурсов с помощью SMB

Ядра Linux до версии 2.6.37 содержали отдельные драйверы SMB и CIFS и позволяли смонтировать общий ресурс с использованием оригинальных протоколов SMB; для этого нужно было либо указать в качестве типа файловой системы `smbfs`, либо использовать команду `smbmount`. При этом все работало почти так же, как и при использовании типа файловой системы `cifs` или команды `mount.cifs`, хотя имелись некоторые отличия в деталях. Использование протокола SMB делало невозможным использование функций, присущих только CIFS, например, расширений UNIX.

Раньше иногда имело смысл использовать SMB; например, можно было монтировать файловые ресурсы очень старых компьютеров под управлением Windows 9x/Me, используя драйвер Linux `smbfs`, но не используя `cifs`. Сегодня такие ситуации встречаются очень редко, поскольку в современной реализации `cifs` была устранена большая часть его когда-то существовавших ограничений. Тем не менее, если вы столкнулись с подобной проблемой, то попробуйте инсталлировать ядро Linux с версией до 2.6.37 и проверить, поможет ли драйвер `smbfs` решить ее.

### Постоянное монтирование общих ресурсов

Если необходимо смонтировать файловый ресурс SMB/CIFS на постоянной основе, то для этого нужно добавить запись в файл `/etc/fstab`. Эти изменения похожи на все другие изменения файла `/etc/fstab`, которые возникают в процессе работы команды `mount`. Однако в [таблице 2](#) есть одна опция, которая заслуживает особого внимания в данной ситуации, а именно `credentials`. Поскольку большинство SMB/CIFS серверов используют для аутентификации пароли, то для монтирования файловых ресурсов с помощью `/etc/fstab` необходимо хранить постоянный пароль. Хотя пароль можно хранить непосредственно в файле `/etc/fstab` (используя опцию `password`), делать этого не рекомендуется – поскольку файл `/etc/fstab` должен быть доступен для чтения всем пользователям, то любой пользователь может также увидеть и пароль. Использование опции `credentials` позволяет хранить пароли в файле, доступном для чтения только пользователю `root`, что повышает их защищенность.

Рабочая запись в файле `/etc/password` для общего ресурса SMB/CIFS может выглядеть следующим образом:

```
//TANGO/BACKUPS /saveit cifs credentials=/etc/samba/creds.txt 0 0
```

Связанные с ней учетные данные могут выглядеть так:

```
username=buuser  
password=Iw2bUmS[ t
```

**Предупреждение.** Не забудьте установить для файла подходящие права доступа (обычно 0600 или 0400) и назначить владельца (пользователь `root` или другой пользователь, учетные данные которого хранятся в файле).

После выполнения необходимых настроек файловый ресурс `//TANGO/BACKUPS` должен автоматически монтироваться каждый раз при перезагрузке компьютера или выполнении

команды `mount -a`. Если это не сработает, то убедитесь в том, что указаны правильные имя пользователя и пароль, проверьте с помощью команды `mount` и выполните другие стандартные действия для устранения неполадок.

## Что дальше

Следующая статья этой серии "[Изучаем Linux, 302 \(смешанные среды\): NetBIOS и WINS](#)" содержит материалы цели 314.2 темы 314. В ней рассматриваются вопросы разрешения имен с помощью службы Windows Internet Name Service (WINS) и просмотр сетевого окружения, что позволяет компьютерам обнаруживать сетевые ресурсы в древовидной иерархии компьютеров и ресурсов.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): CIFS integration](#) (EN).
- В статье "[Изучаем Linux, 302 \(смешанные среды\): Основные принципы](#)" (developerWorks, февраль 2011 г.) изложены основные принципы протоколов SMB и CIFS.
- Вики-страницы Samba содержат [раздел о расширениях UNIX протокола CIFS](#) (EN), в котором описаны технические подробности этой функциональности.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.

# Изучаем Linux, 302 (смешанные среды): NetBIOS и WINS

*Конфигурирование Samba для управления системой имен и просмотра сетевого окружения*  
[Родерик Смит \(Roderick Smith\)](#), автор и консультант, IBM

**Описание:** В сетях SMB/CIFS используется нестандартная схема имен. Хотя для обращения друг к другу современные клиенты могут использовать доменные Интернет-имена, более старые клиенты используют для этого разработанную компанией Microsoft службу *Windows Internet Name Service (WINS)*, которая также известна как сервер имен NetBIOS (NetBIOS Name Server, NBNS). Таким образом, важную роль играют правильная настройка Samba для разрешения имен, а также настройка просмотра сетевого окружения – механизма, позволяющего серверам узнавать, какие общие ресурсы доступны в сети.

**Дата:** 26.07.2012

**Уровень сложности:** средний

# Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

## Краткий обзор

В этой статье рассматриваются следующие темы:

- Windows Internet Name Service (WINS).
- Сетевая базовая система ввода-вывода (NetBIOS).
- Функции локального главного браузера.
- Функции главного браузера домена.
- Разрешение имен.
- Конфигурирование и использование Samba в качестве сервера WINS/NBNS.
- Просмотр сетевого окружения, выборы браузера и объявление служб NetBIOS.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 314.2 темы 314. Цель имеет вес 7.

## Предварительные требования

Предполагается, что читатель умеет работать с командной строкой Linux, знает основы конфигурирования Samba и имеет общее представление о структуре конфигурационного файла smb.conf. Также будет полезно понимание основ системы доменных имен (DNS), поскольку с этой технологией связаны некоторые темы, рассматриваемые в этой статье.

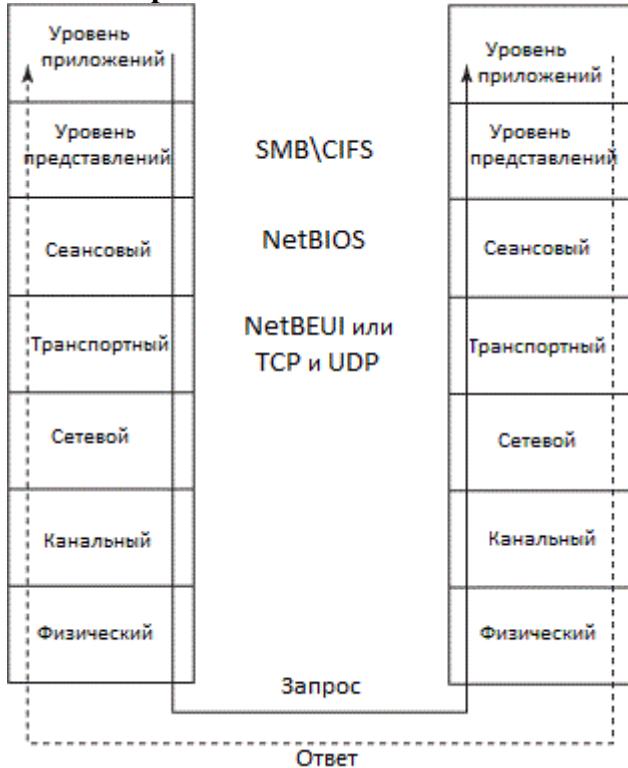
## Что такое разрешение имен NetBIOS

Обычно имена компьютеров не зависят от используемого протокола. Тем не менее, при использовании протоколов SMB/CIFS возникают определенные проблемы с именами компьютеров, поскольку эти протоколы разрабатывались без привязки к стеку TCP/IP. Отдельно следует отметить то, что протоколы SMB/CIFS начинали разрабатываться в сетях, в которых использовались протоколы NetBIOS и NetBEUI (NetBIOS Extended User Interface) со своими собственными правилами. Когда протоколы SMB/CIFS были модифицированы для работы с TCP/IP, то некоторые старые соглашения о NetBIOS-именовании также были адаптированы для работы с TCP/IP. Иногда при работе с современными клиентами и серверами SMB/CIFS возникает необходимость в соблюдении этих схем именования, поэтому следует понимать определенные правила, включая различные способы, с помощью которых NetBIOS интерпретирует имена, и роль WINS серверов в сети.

## Что такое NetBIOS

На рисунке 1 изображены два сетевых стека (для клиента и для сервера), использующих модель OSI. В этой модели SMB/CIFS можно рассматривать как стек протоколов уровня представлений. В исходной реализации (до адаптации к TCP/IP) протоколы SMB/CIFS использовали протокол NetBIOS на сеансовом уровне, который, в свою очередь, использовал протокол NetBEUI на транспортном уровне.

**Рисунок 1. NetBEUI и NetBIOS – протоколы сеансового и транспортного уровня стека сетевых протоколов**



В операционной системе Linux протоколы NetBIOS и NetBEUI отсутствуют. Все функции протоколов SMB/CIFS и NetBIOS выполняет Samba. Вместо использования NetBEUI Samba осуществляет сетевые взаимодействия через протоколы TCP и UDP. Иногда комбинацию двух протоколов NetBIOS over TCP/IP называют механизмом *NBT*, который отвечает за реализацию следующих служб:

- **Служба имен.** В сетях TCP/IP компьютеры имеют имена, которые назначили им люди, но взаимодействуют они с помощью IP- и MAC-адресов. В NetBEUI не используются IP-адреса, а используются только имена и MAC-адреса. Таким образом, NBT берет на себя задачу создания IP-адресов, которые не требуются для работы в SMB/CIFS over NetBIOS и NetBEUI сетях. Этому вопросу посвящена первая половина нашей статьи.
- **Служба датаграмм.** Эта служба представляет собой отдельный тип сетевого взаимодействия, при котором не создается постоянного подключения. В сетях TCP/IP ее логическим эквивалентом является протокол UDP, поэтому NBT использует UDP для функций, не требующих использования служб датаграмм (например, выборы браузера, о которых пойдет речь во второй половине этой статьи).
- **Служба сеансов.** Операции NetBIOS создают долговременные подключения для большинства запросов на передачу данных, например, при монтировании файлового ресурса пользователем. Логически этот тип служб соответствует протоколу TCP стека TCP/IP, поэтому для таких подключений NBT использует протокол TCP.

На практике NBT упоминается не часто: при упоминании имен, используемых в сетях SMB/CIFS, и их сопоставлениях с IP-адресами обычно говорят об *NetBIOS-именах* или о *разрешении имен NetBIOS*. NetBIOS-имена разделяются на две группы:

- **Имя компьютера.** Это имя, так же, как и DNS-имя компьютера, относится к определенному компьютеру. На практике удобнее, чтобы имя компьютера совпадало с его DNS-именем. Имя компьютера не должно быть уникальным во всем мире, но должно быть уникальным в пределах вашей сети (по крайней мере, в пределах рабочей группы или домена).

- **Имя рабочей группы.** Это имя охватывает все компьютеры в сети. В простой конфигурации SMB/CIFS (без использования домена) имя рабочей группы – это просто способ организации компьютеров, позволяющий находить их в сетевом окружении, о чем будет рассказано во второй половине этой статьи. В доменном сетевом окружении имя рабочей группы назначается домену, предоставляющему различные сервисы на уровне домена, о чем рассказывается в статье "[Изучаем Linux, 302 \(смешанные среды\): управление доменом](#)".

Такая двухуровневая система именования является одной из причин, по которым работа в сетях NetBIOS ограничена локальной сетью, в то время как с глобальная сеть Интернет, построена на TCP/IP. При использовании лишь двух уровней именования не имеет смысла создавать имена в таком количестве, которое обеспечило бы их уникальность среди миллионов компьютеров.

В сетях NetBIOS каждый компьютер имеет собственное имя, которое он анонсирует в сеть с помощью *широковещательной передачи*. Этот децентрализованный подход к именованию упрощает настройку для начинающих пользователей и администраторов при условии, что они не пытаются использовать уже задействованное имя.

NetBIOS-имена могут иметь длину до 15 символов, содержать буквы, цифры и следующие символы:

! @ # \$ % ^ & ( ) - ' { } . ~

На практике лучше всего ограничиться буквами и, возможно, цифрами. Регистр символов в именах не учитывается. В этой статье я использую заглавные буквы для NetBIOS-имен и строчные – для DNS-имен.

Хотя операционные DOS, Windows и IBM Operating System/2 (OS/2) поддерживают NetBEUI, сегодня практически везде они используют NBT (даже в тех сетях, в которых все компьютеры могут использовать NetBEUI). Основные принципы работы NBT и системы разрешения имен NetBIOS одинаковы для всех этих платформ.

### Способы разрешения NetBIOS-имен

Предположим, что пользователь работает за клиентским компьютером и хочет получить доступ к серверу с именем **SPEAKER**. Он запускает соответствующее клиентское приложение и вводит в качестве имени сервера **SPEAKER**. Что происходит в этот момент? В сетях NBT существует четыре различных способа разрешения имен:

- **Широковещательное разрешение имен.** Клиент может послать широковещательное сообщение, получаемое всеми компьютерами в сети и содержащее запрос IP-адреса, связанного с именем хоста SPEAKER. После этого компьютер с указанным именем ответит ему и тогда клиент сможет начать установку соединения для последующей передачи данных. Этот тип разрешения имен иногда называется *B-режимом* разрешения имен (B mode), а использующий его компьютер – *B-узлом* (B-node).
- **Использование сервера имен.** Клиент может обратиться к серверу NBNS, или WINS (эти два названия являются синонимами). Так же, как и DNS-сервер, WINS-сервер хранит сопоставления имен компьютеров и их IP-адресов. Однако WINS-сервер получает эти данные на основе широковещательных сообщений, рассылаемых компьютерами при их загрузке, а не путем просмотра локального конфигурационного файла или обращений к другим серверам. Этот способ разрешения имен иногда называется *P-режимом* разрешения имен (P-mode).
- **Использование файла lmhosts.** Сопоставления имен компьютеров и их IP-адресов хранятся в конфигурационном файле с именем *lmhosts* (обычно он хранится в

директории /etc/samba в системах с Samba-сервером или в директории C:\WINDOWS\SYSTEM32\DRIVERS\ETC в системах Windows). Этот файл похож на файл /etc/hosts, который выполняет те же самые функции для имен хостов TCP/IP.

- **Использование DNS.** Хотя система DNS и не является частью NBT, можно настроить Samba на ее использование для разрешения NetBIOS-имен (это же относится к современным операционным системам Windows).

Многие компьютеры будут настроены на использование нескольких методов разрешения имен для повышения шансов нахождения требуемого компьютера с минимальными усилиями. В качестве основного метода разрешения NetBIOS имен старые операционные системы Microsoft Windows 9x или Windows Me по умолчанию используют В-режим, а большинство современных версий Windows – систему доменных имен DNS.

WINS-сервер (независимо от того, является ли он Linux-компьютером с установленным сервером Samba или Windows-сервером) должен прослушивать широковещательные сообщения с запросами на разрешение имен, хранить сопоставления NetBIOS-имен и IP-адресов и отвечать на запросы клиентов. На практике это означает, что WINS-сервер должен быть всегда доступен (или почти всегда, насколько это возможно). Часто на роль WINS-сервера отлично подходит контроллер домена. Если на каком-либо компьютере работает собственный DNS-сервер, то этот компьютер также может быть кандидатом на эту роль. Необходимо явно настроить выбранный компьютер на работу в качестве WINS-сервера, о чем будет рассказываться в следующем разделе применительно к серверам Samba.

### Конфигурирование системы разрешения имен в Samba

Определившись с требованиями и способами разрешения NetBIOS-имен, можно приступать к настройке системы разрешения имен в Samba. Эта задача делится на две части: настройка параметров сервера Samba, работающего в качестве клиента, и настройка параметров Samba, работающего в качестве WINS-сервера.

#### Настройка Samba в качестве клиента

Следующие параметры файла smb.conf влияют на то, как Samba себя в сети и выполняет разрешение NetBIOS-имен:

- **netbios name.** С помощью этого параметра задается NetBIOS-имя, заявляемое сервером. Если этот параметр не задан, то по умолчанию используется часть DNS-имени компьютера, соответствующая имени хоста; например, если DNS-имя компьютера *tunesmith.example.com*, то NetBIOS-именем будет *TUNESMITH*.
- **netbios aliases.** С помощью этого параметра один компьютер может заявлять несколько NetBIOS-имен, например, *aliases = PRILL HALRL0PRILLALAR*. Эти имена заявляются в дополнение к любому имени, заданному с помощью параметра *netbios name*.
- **workgroup.** С помощью этого параметра задается имя рабочей группы или домена. Для просмотра сетевого окружения и правильной работы службы домена необходимо задать для этого параметра правильное значение. По умолчанию значением этого параметра является *WORKGROUP*.
- **name resolve order.** Этот параметр принимает от одной до четырех опций и управляет тем, какой из четырех способов разрешения имен будут использовать программы пакета Samba. Для серверной части этот параметр не имеет большого практического значения, однако он влияет на то, как будут выполнять разрешение имен такие утилиты, как, например, *smbclient*.
- **wins server.** С помощью этого параметра задаются имена или IP-адреса компьютеров, настроенных на работу в качестве WINS-серверов. Заметим, что установка этого параметра не гарантирует, что к этим серверам будут выполняться обращения – эта функциональность настраивается с помощью параметра *name*

## **resolve order.**

Обычно для нормальной работы достаточно задать значение параметра **netbios name** без настройки параметра **netbios aliases**, однако при определенных обстоятельствах требуется использовать оба этих параметра. Например, если сервер Samba должен выполнять функции серверов, которые выводятся из эксплуатации, то параметр **netbios aliases** поможет настроить его так, что он будет отвечать на запросы, направляемые этим компьютерам. Если вы хотите использовать для NetBIOS-адресации одно имя, а для всех остальных задач – другое, то вам помогут параметры **netbios name** и **netbios aliases**.

При использовании параметра **name resolve order** указываются одна или несколько из четырех опций, разделенные пробелом:

- **lmhosts.** Эта опция указывает на использование файла /etc/samba/lmhosts, который будет рассмотрен более подробно ниже.
- **host.** Эта опция указывает на традиционный способ разрешения имен TCP/IP, включающий в себя использование DNS-сервера, файла /etc/hosts или других методов в зависимости от локальной конфигурации среды TCP/IP.
- **wins.** Эта опция указывает на использование WINS-сервера, который указывается с помощью параметра **wins server**.
- **bcast.** Если указать эту опцию, то клиенты Samba будут выполнять разрешение имен с использованием широковещательных запросов NetBIOS.

По умолчанию параметр **name resolve order** использует четыре метода в указанном выше порядке. Эта очередность подходит для многих ситуаций, однако, ее можно изменить. Например, если в сетевом окружении не используется файл lmhosts, а вместо методов DNS предпочтительнее использовать методы NetBIOS, то можно указать следующее значение:

```
name resolve order = wins bcast host
```

Если для разрешения имен используется WINS-сервер, то следует убедиться, что конфигурация содержит параметр **wins server**. В противном случае Samba не сможет обнаружить WINS-сервер, и результат будет такой же, как если бы в параметре **name resolve order** не была указана опция **wins**.

Если вы планируете использовать и обслуживать файл lmhosts, то следует понимать его формат, который соответствует формату файла /etc/hosts. Для каждого компьютера в этом файле имеется строка, начинающаяся с IP-адреса и заканчивающаяся NetBIOS-именем. Комментарии начинаются с символа решетки (#). Файл lmhosts может выглядеть примерно так:

```
# Edit this file when adding or removing machines
192.168.7.5 NESSUS
192.168.7.8 PRILL
192.168.7.56 TUNESMITH
```

## **Настройка Samba в качестве WINS-сервера**

В дополнение к параметрам Samba, влияющим на разрешение NetBIOS-имен, можно задать ряд параметров, позволяющих серверу Samba работать в качестве WINS-сервера:

## Репликация WINS

WINS-серверы Windows могут выполнять процесс, который называется *репликацией WINS* и позволяет обмениваться данными между несколькими WINS-серверами, расположенными в разных подсетях. К сожалению, в Samba версии 3 репликация WINS не поддерживается, хотя она запланирована в Samba версии 4. Какое-то время поддержку репликации WINS обеспечивала утилита `wrepl`, однако она никогда не имела статуса законченного продукта, и впоследствии от нее отказались.

На практике отсутствие репликации WINS не является серьезной проблемой для большинства конфигураций, поскольку возможно использовать параметры `dns proxy` или `wins proxy`; кроме того, в наши дни для разрешения имен в большинстве случаев используется система DNS.

- **wins support.** Если этот логический параметр установлен в `Yes`, то Samba работает в качестве WINS-сервера. По умолчанию этот параметр установлен в `No`.
- **wins proxy.** Этот логический параметр определяет, будет ли Samba отвечать на широковещательные запросы разрешения всех имен, кроме собственного. Установка этого параметра в `Yes` (по умолчанию он установлен в `No`) может повысить надежность разрешения NetBIOS-имен, если сетевое окружение разделено на несколько подсетей или если часть компьютеров настроена на использование широковещательных запросов, а часть – на использование WINS-серверов.
- **dns proxy.** Если этот логический параметр установлен в `Yes` (это значение используется по умолчанию), то происходит следующее: если WINS-сервер Samba не содержит нужного NetBIOS-имени в файле, отвечающем за разрешение имен, то поиск этого имени выполняется в системе DNS с применением локально настроенных параметров DNS.
- **max wins ttl.** При работе с WINS-сервером Samba хранит накопленную информацию об именах в течение определенного временного интервала, устанавливаемого на основе клиентского запроса и данного параметра. Значение параметра задается в секундах и по умолчанию составляет `518400` (6 дней).
- **min wins ttl.** Этот параметр работает так же, как и параметр `min wins ttl`, но задает минимальный временной интервал для хранения информации о полученных именах. Значение этого параметра по умолчанию составляет `21600` (6 часов).

Как правило, для настройки работы Samba в качестве WINS-сервера достаточно задать параметр `wins server = Yes`. Поскольку не существует способа автоматического обнаружения клиентами WINS-серверов, то необходимо настроить клиентские компьютеры и указать для них требуемый WINS-сервер. Самый простой способ сделать это (по крайней мере, на Windows-клиентах) – это настроить необходимые параметры на DHCP-сервере. Популярный сервер доменных имен `named` можно настроить с помощью его конфигурационного файла `dhcpd.conf`, обычно расположенного в директории `/etc`. Необходимо задать два параметра, которые уже могут присутствовать в начале конфигурационного файла:

```
option netbios-name-servers 192.168.7.2;
option netbios-node-type 8;
```

Первая из этих двух строк определяет IP-адрес WINS-сервера Samba. Вторая строка задает *тип узла*, определяя способ разрешения имен. В этой строке можно указать значения от `1` до `8`, однако полезными являются только четыре из них:

- **1.** Клиентские компьютеры должны использовать только широковещательные сообщения, т. е. являются В-узлами.

- **2.** Клиентские компьютеры должны использовать только WINS-сервер, т. е. являются Р-узлами.
- **4.** Клиентские компьютеры должны сначала использовать широковещательные сообщения, а если это не срабатывает, то использовать WINS-сервер. Такие клиентские компьютеры иногда называют *M*-узлами.
- **8.** Клиентские компьютеры должны сначала использовать WINS-сервер, а если это не срабатывает, то использовать широковещательные сообщения. Такие клиентские компьютеры иногда называют *H*-узлами.

В общем случае рекомендуется настраивать клиентские компьютеры в качестве Н-узлов, поэтому на DHCP-сервере рекомендуется указывать тип NetBIOS-узла 8.

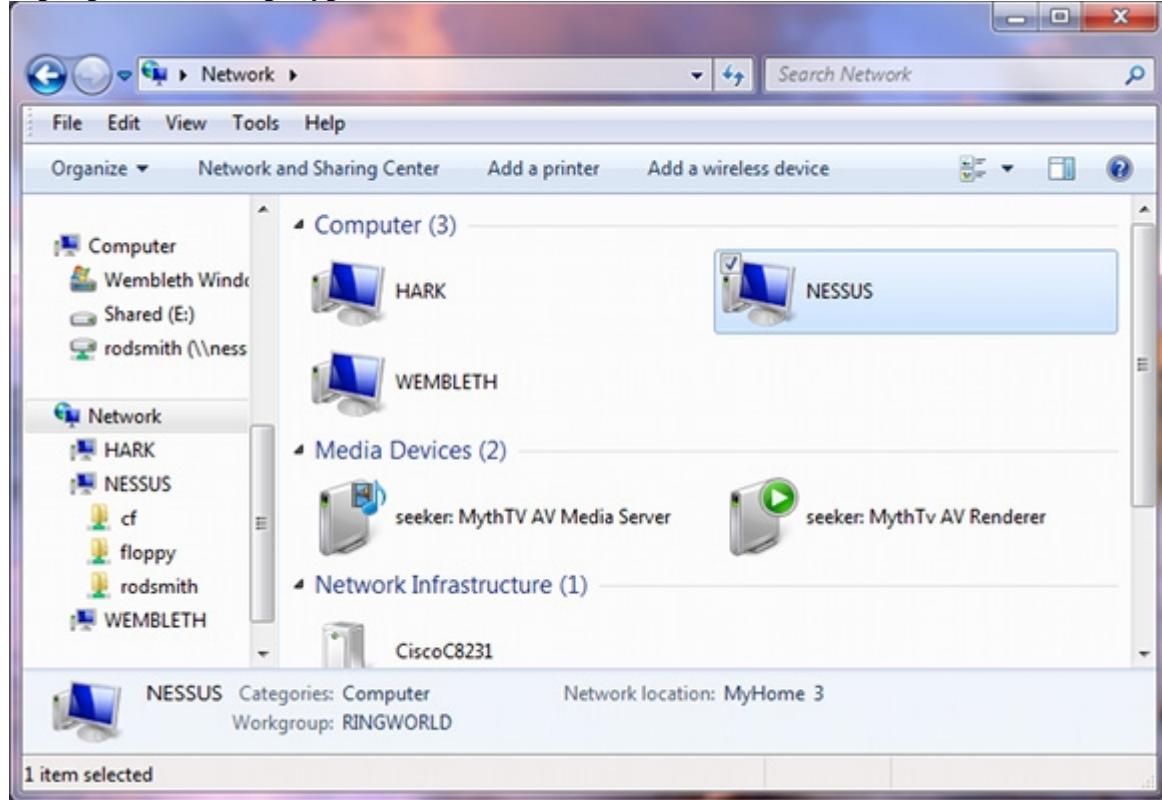
### **Что такое просмотр сетевого окружения**

В компьютерном мире термин *browsing* (просмотр) обычно относится к всемирной WWW-сети; для просмотра Web-сайтов мы используем Web-браузеры, как вы делаете это сейчас. В контексте SMB/CIFS термин *browsing* означает нечто другое (хотя и похожее по смыслу): просмотр сетевого окружения означает процесс обнаружения доступных в сети ресурсов. Samba изначально имеет параметры конфигурации, влияющие на то, как она взаимодействует с клиентскими компьютерами и просматривает сетевое окружение. Однако, для того чтобы настраивать эти параметры, необходимо понимать ключевые принципы, на которых основаны механизмы просмотра сетевого окружения, например, принципы работы *главного браузера* и проведения *выборов браузера*.

### **Обзор механизма просмотра сетевого окружения**

Если вы работали с общими файлами в Windows, то, вероятно, вы просматривали сетевое окружение, как показано на рисунке 2. На этом рисунке изображено окно **Computer** операционной системы Windows: узел **Network** в левой панели окна содержит список доступных в сети компьютеров – HARK, NESSUS и WEMBLETH. Можно щелкнуть мышью на каждом из этих компьютеров, чтобы посмотреть, какие общие ресурсы он содержит, щелкнуть на значке общего ресурса компьютера, чтобы узнать, какие на нем есть файлы, и т. д.

**Рисунок 2. Просмотр сетевого окружения позволяет пользователям быстро находить серверы и общие ресурсы**



С точки зрения пользователя все это происходит автоматически, но, конечно же, все это работает, благодаря значительным усилиям, которые не видны с первого взгляда. Просмотр сетевого окружения основан на использовании компьютера, который называется *главным браузером* и формирует список компьютеров. Конечно же, главный браузер – это сервер. Этот сервер может не нуждаться в сложном конфигурировании, но для того чтобы избежать неполадок при просмотре сети (например, внезапное исчезновение из списка всех компьютеров без видимых причин), необходимо понимать, как работает механизм просмотра сетевого окружения.

Официально существует два основных типа браузеров:

- **Локальные главные браузеры** – обслуживают одну подсеть; прослушивают широковещательные сообщения, в которых компьютеры сообщают о себе и передают свои NetBIOS-имена, а также управляют своими списками просмотра, формируемыми на основе этих сообщений. Также они отвечают на широковещательные запросы обнаружения главных браузеров. Локальные главные браузеры выбираются на основе процесса, который называется *выборами*, поэтому в принципе, локальным главным браузером может стать любой компьютер (напомним, что протоколы SMB/CIFS были разработаны для использования в одноранговых конфигурациях, в которых все компьютеры являлись одновременно и клиентами, и серверами).
- **Главные браузеры доменов.** Поскольку локальные главные браузеры формируют свои списки серверов, прослушивая широковещательные сообщения, то они не могут работать в нескольких подсетях (по крайней мере, без дополнительной помощи). В таких случаях в дело вступают главные браузеры доменов, которые взаимодействуют с локальными главными браузерами нескольких подсетей и обмениваются списками просмотра, позволяя компьютерам из различных подсетей видеть друг друга (хотя такой подход приводит к некоторым задержкам при внесении и удалении серверов из списков просмотра). Главные браузеры доменов должны быть указаны явным образом – они не назначаются в результате выборов. Обычно в роли главного браузера доменов

выступают контроллеры доменов.

В разделе [Настройка параметров просмотра сетевого окружения в Samba](#) описываются параметры Samba, позволяющие локальным главным браузерам обмениваться списками просмотра с главными браузерами других подсетей. Это может пригодиться в тех случаях, когда имеется сеть из нескольких сегментов, но не хочется создавать доменную конфигурацию.

Главные браузеры могут иметь своих "заместителей", которые называются *резервными браузерами*. Резервные браузеры обслуживают списки просмотра, но в обычных ситуациях клиентские компьютеры к ним не обращаются. Назначение резервного браузера – взять на себя роль главного браузера, если последний окажется недоступен.

## Выборы главного браузера

Выборы начинаются в тот момент, когда компьютер запрашивает об этом; обычно это происходит тогда, когда предыдущий главный браузер перестает отвечать на запросы, но некоторые системы (например, в Samba) могут инициировать выборы в момент загрузки или в любой другой момент времени. Как правило, вся процедура выборов занимает несколько секунд, в течение которых просмотр сетевого окружения становится невозможен, поэтому не стоит настраивать Samba на слишком частые запросы процедуры выборов. Выборы начинаются тогда, когда компьютер, запрашивающий их, рассыпает широковещательный пакет с критериями выборов. Эти критерии можно ранжировать, как будет показано ниже. Любой компьютер, приоритет критерия которого выше, чем приоритет полученного критерия, посыпает в ответ широковещательное сообщение со своим критерием. Компьютер, который считает себя победившим в выборах, продолжает рассыпать свои широковещательные критерии выборов каждые 200-800 миллисекунд до тех пор, пока это не будет сделано четыре раза, после чего считается, что он победил в выборах, и он становится локальным главным браузером.

Критерии, используемые при выборе главного браузера, можно упорядочить следующим образом (в порядке убывания приоритета):

1. **Версия протокола выборов.** Теоретически можно использовать несколько протоколов выборов, но на сегодняшний день используется только версия 1. Таким образом, этот критерий мог бы играть важную роль, но на практике это не так.
2. **Приоритет операционной системы.** Каждая операционная система имеет свой приоритет от 0 до 255. Операционные системы с большим приоритетом являются более предпочтительными. Большинство операционных систем Microsoft имеют приоритеты от 1 до 32. Можно настроить приоритет для Samba, как будет описано в следующем разделе [Настройка параметров просмотра сетевого окружения в Samba](#).
3. **Статус основного контроллера домена.** Если компьютер является действующим основным контроллером домена, то он побеждает в выборах среди всех компьютеров с тем же самым приоритетом ОС и версией протокола выборов.
4. **Статус WINS-сервера.** Если компьютер является действующим WINS-сервером, то он побеждает в выборах, если результат не был определен предыдущими факторами.
5. **Предпочитаемый главный браузер.** Компьютер может быть назначен *предпочитаемым главным браузером*, что дает ему преимущество для победы в выборах.
6. **Текущий главный браузер.** Если компьютер является действующим главным браузером, то получает преимущество для победы в выборах.
7. **Бывший главный браузер.** Если компьютер работал в качестве главного браузера, но проиграл последние выборы и был понижен до статуса резервного браузера, то он получает небольшое преимущество для победы в выборах.
8. **Работающий резервный браузер.** Если компьютер является действующим резервным браузером, то получает в выборах небольшой перевес в свою пользу.

Если по результатам выборов не получилось определить победителя, то используются два дополнительных критерия:

- **Время непрерывной работы.** В выборах побеждает тот компьютер, время работы которого больше, чем время работы остальных компьютеров.
- **NetBIOS-имя.** В выборах побеждает тот компьютер, имя которого стоит первым в списке, отсортированном по алфавиту.

## Настройка параметров просмотра сетевого окружения в Samba

В Samba имеется множество параметров, влияющих на ее участие в выборах главного браузера и возможность работы в качестве главного браузера домена. В этой статье я расскажу о параметрах, которые можно использовать для "фальсификации" выборов, просмотра сетевого окружения с несколькими подсетями и получения статуса главного браузера домена. Также я расскажу о параметрах серверов Samba (даже если они *не являются* главными браузерами) для настройки общих ресурсов, которые видят пользователи, заходя на ваш компьютер в сетевом окружении.

## Настройка Samba для победы в выборах

Следующие параметры могут повлиять на победу Samba в выборах локального главного браузера:

- **local master.** Если этот параметр установлен в **Yes** (значение по умолчанию), то Samba принимает участие в выборах локального главного браузера. Если же этот параметр установлен в **No**, то Samba не принимает участия в выборах и, следовательно, не может победить в них.
- **os level.** По умолчанию параметр **os level** в Samba равен **20**. Это значение позволяет Samba выигрывать выборы у большинства клиентских операционных систем Microsoft, но не у серверных операционных систем, приоритет которых приближается к **32**. Если вы хотите обеспечить абсолютную победу Samba в выборах, то задайте для этого параметра значение **255**, что позволит выиграть выборы у всех компьютеров, за исключением серверов Samba с аналогичной конфигурацией. Не выставляйте это значение где только возможно, поскольку в сети с несколькими серверами Samba, только один из них можно "фальсифицировать" таким образом.
- **domain logons.** Как рассказывалось в статье, содержащей материалы цели 312.4 экзамена LPIC, этот логический параметр настраивает сервер Samba на работу в качестве основного контроллера домена и, следовательно, влияет на выбор сервера Samba в качестве главного браузера.
- **wins support.** Поскольку статус WINS-сервера влияет на исход выборов, то этот логический параметр также влияет на выбор сервера Samba в качестве главного браузера.
- **preferred master.** Установка этого логического параметра в **Yes** указывает Samba послать запрос на проведение выборов при ее запуске. Также этот параметр устанавливает флаг "предпочитаемый главный браузер", используемый в процедуре выборов. По умолчанию этот параметр установлен в **No**.
- **browse list.** Этот логический параметр (по умолчанию установлен в **Yes**) определяет, обслуживает ли сервер список просмотра. Если это так, то сервер может стать резервным или главным браузером (если этот параметр установить в **No**, то Samba не будет принимать участие в выборах).

В большинстве случаев серверы Samba нормально работают с настройками по умолчанию: они выигрывают выборы в конфигурациях с рабочими группами, где нет серверов под управлением ОС Windows Server, и позволяют выиграть выборы серверам под управлением ОС Windows Server или специально настроенным для этого серверам Samba в тех

конфигурациях, где они есть.

Если вы столкнулись с проблемами просмотра сетевого окружения, то, возможно, придется изучить параметры конфигурации сервера Samba. Часто проблемы могут возникать в следующих ситуациях:

- **Ненадежные локальные главные браузеры.** Если компьютер, работающий в качестве главного браузера, будет выключен или отсоединен от сети, то это приведет к сбоям просмотра сетевого окружения. Такая ситуация может возникнуть в одноранговой сети, состоящей из множества компьютеров под управлением операционных систем с одинаковым приоритетом, в которой пользователи периодически выключают свои компьютеры. В такой ситуации хорошим решением будет являться настройка надежного сервера Samba, который будет выигрывать выборы (для этого необходимо поднять его приоритет и, возможно, изменить другие параметры).
- **Слишком много предпочтаемых главных браузеров.** Если в качестве предпочтаемого главного браузера настроено более одного компьютера, то это может привести к повторяющимся запросам на проведение выборов, каждый из которых, в свою очередь, будет приводить к кратковременным сбоям просмотра сетевого окружения. Убедитесь в том, что в каждом сегменте сети в качестве предпочтаемого главного браузера настроен только один сервер Samba. Если предпочтаемым главным браузером является компьютер под управлением Windows (например, если это контроллер домена), то убедитесь в том, что ни один сервер Samba не настроен в качестве предпочтаемого браузера.

### **Включение просмотра нескольких подсетей для рабочей группы**

Если вы используете конфигурацию рабочей группы, включающую в себя несколько подсетей, то просмотр сетевых ресурсов и разрешение имен может не работать между этими подсетями. Проблемы, связанные с разрешением имен, можно решить с помощью параметров WINS-сервера, которые обсуждались в первой половине этой статьи (можно также использовать для разрешения имен систему DNS). Однако для просмотра сетевого окружения может потребоваться использовать некоторые необычные параметры, которые есть только в Samba. Одним из способов является настройка WINS-сервера Samba в качестве главного браузера домена. В Samba это можно сделать без необходимости полной настройки домена, поскольку требуемая функциональность разделена на независимые модули. Тем не менее, если вы решите сделать это, то вам придется настроить один сервер Samba как в качестве WINS-сервера, так и в качестве главного браузера домена; если же вы решите не делать этого, то главный браузер домена не получит информацию обо всех остальных именах хостов, необходимых для его работы.

Если вы не можете настроить один компьютер в качестве WINS-сервера и главного браузера доменов, или вы не можете заставить все ваши компьютеры зарегистрироваться на WINS-сервере, то объединить несколько подсетей вам помогут следующие параметры:

- **remote browse sync.** В этом параметре можно указать IP-адреса серверов Samba или целых подсетей, после чего Samba будет обмениваться списками просмотра со всеми серверами Samba или главными браузерами, которые она обнаружит в указанных подсетях. Важным ограничением этого параметра является то, что он работает только с серверами Samba (т. е. Samba не может обмениваться списками просмотра с локальными главными браузерами под управлением Windows).
- **remote announce.** Samba может анонсировать информацию о себе в удаленные сети, передавая IP-адрес (включая широковещательный адрес подсети) и необязательное имя рабочей группы. Например, в сообщении 192.168.8.255/РАК Samba заявляет себя как локальный главный браузер для рабочей группы РАК в

подсети 192.168.8.0/24. В отличие от параметра `remote browse sync`, этот параметр позволяет передавать списки просмотра любому главному браузеру, а не только серверам Samba.

- **enhanced browsing.** Этот логический параметр (по умолчанию установлен в `Yes`) заставляет Samba искать и обмениваться списками просмотра с обнаруженными главными браузерами домена, используя для этого WINS-сервер или любые другие способы. Этот параметр позволяет обеспечить более надежный просмотр сетевого окружения, состоящего из нескольких подсетей, но также может не удалять несуществующие рабочие группы из списков просмотра, поэтому, если вы столкнулись с такой проблемой, то, возможно, следует установить этот параметр в `No`.

В общем случае обмен списками просмотра между подсетями можно обеспечить следующим способом: использовать в каждой подсети локальный главный браузер Samba, настроенный с помощью параметра `remote browse sync`, в котором указаны серверы Samba других подсетей. Если вы не знаете точно IP-адресов удаленных локальных главных браузеров, то можно указать широковещательный адрес подсети, например, `192.168.8.255` для сети `192.168.8.0/24`.

### Настройка параметров главного браузера домена

Обычно Samba берет на себя роль главного браузера домена, когда она настраивается в качестве контроллера домена. Это можно настроить принудительно, установив параметр `domain master` в `Yes`. Этот параметр может принимать логические значения `Yes` или `No`, а также значение `Auto` (значение по умолчанию), подставляющее в конфигурацию значение параметра `domain logins`, о котором рассказывалось в статье "[Изучаем Linux, 302 \(смешанные среды\): управление доменом](#)".

После того, как сервер Samba настроен в качестве контроллера домена и главного браузера домена, все члены домена не должны испытывать никаких сложностей с его использованием. Локальные главные браузеры из других подсетей могут обнаруживать главный браузер домена, используя запросы доменных имен, содержащие коды для идентификации главного браузера домена. Чтобы обеспечить еще более надежную работу, можно использовать на локальных главных браузерах Samba параметр `remote browse sync`.

Имейте в виду, что передача списков просмотра между подсетями не происходит мгновенно. Локальным главным браузерам необходимо время, чтобы сформировать списки просмотра, а их синхронизация с главным браузером домена происходит от случая к случаю. Все это может приводить к тому, что прежде, чем все компьютеры появятся во всех списках просмотра всех подсетей, может пройти несколько минут.

### Настройка локальных параметров просмотра

Большинство рассмотренных в этой статье параметров было связано с главными браузерами, поддерживающими списки компьютеров. Однако для того, чтобы можно было просматривать сеть, отдельные серверы должны сообщать о предоставляемых ими общих ресурсах. Обычно параметры по умолчанию обеспечивают нормальную работу, и большинство общих ресурсов появляются в списках просмотра. Частным исключением является ресурс `HOMES`: этот общий ресурс создается для каждого пользователя и обычно виден только тому пользователю, которому он принадлежит. Даже если сервер Samba не является главным браузером, для него также можно задать несколько параметров, отвечающих за настройку локальных файловых ресурсов и ресурсов печати:

- **preload.** С помощью этого параметра указывается список общих ресурсов, которые должны присутствовать в списках просмотра, даже если обычно они не доступны. Например, можно указать домашние директории определенных пользователей: параметр `preload = LOUIS` позволит всем просматривать домашнюю директорию

пользователя Louis. Этот параметр также называется `auto services`.

- **load printers.** Этот логический параметр (по умолчанию установлен в `Yes`) определяет, будут ли доступны общие ресурсы печати, созданные командой `PRINTERS`.
- **browsable.** Этот логический параметр по умолчанию установлен в `Yes` для большинства общих ресурсов, за исключением нескольких специальных типов. Если он установлен в `Yes`, то общий ресурс отображается в списках просмотра. Если же он установлен в `No`, то общий ресурс не отображается, но к нему могут получить доступ все пользователи, которые знают о его существовании.

## Проверка работы NetBIOS и WINS

После того, как сервер Samba настроен в качестве WINS-сервера или главного браузера (а, может, и раньше), может возникнуть желание проверить работу службы разрешения NetBIOS-имен и функций просмотра сетевого окружения. Для этого в Samba имеется несколько инструментов. Самые примечательные из этих инструментов – это `findsmb` и `smbtree` (работа обоих основана на использовании низкоуровневых утилит, таких как `nmblookup` и `smbclient`, выполняющих всю "тяжелую" работу).

Программа `findsmb` – это инструмент, посылающий широковещательные запросы на обнаружение NetBIOS-компьютеров в сети и возвращающий результаты поиска. Обычно эта программа запускается без каких-либо опций, хотя можно указать ей адрес удаленной опрашиваемой подсети или опцию `-r`, устраняющую недостатки операционной системы Microsoft Windows 95.

### Листинг 1. Пример вывода команды `findsmb`

```
$ findsmb
          *=DMB
          +=LMB
IP ADDR      NETBIOS NAME      WORKGROUP/OS/VERSION
-----
192.168.7.1   CISC0C8231    +[           ]
192.168.7.5   NESSUS       *[RINGWORLD] [Unix] [Samba 3.4.12]
192.168.7.9   HARK         [RINGWORLD] [Unix] [Samba 3.5.7]
192.168.7.56  TUNESMITH   +[RINGWORLD] [Unix] [Samba 3.0.28a-apple]
```

Этот листинг содержит не только имена и IP-адреса локальных компьютеров, но также информацию об их рабочих группах, операционных системах и версиях запущенных серверов Samba.

В отличие от программы `findsmb`, программа `smbtree` использует несколько опций. Эта программа используется для получения информации просмотра сетевого окружения. В простейшем случае она выводит список просмотра, аналогичный тому, который можно обнаружить в файловом проводнике Windows (как на [рисунке 2](#)), но только в текстовом виде, как показано в листинге 2. Обратите внимание на то, что `smbtree` запрашивает пароль пользователя, используемый для доступа к информации о ресурсах, которую некоторые серверы показывают только после успешного входа в систему.

### Листинг 2. Пример вывода команды `smbtree`

```
$ smbtree
Enter rodsmith's password:
```

```

RINGWORLD
    \\WEMBLETH
        \\WEMBLETH\IPC$           wembleth server (Samba, Ubuntu)
        \\WEMBLETH\programs       IPC Service (wembleth server (Samba, Ubuntu))
    \\SEEKER
        \\SEEKER\rodsmit          User programs
        \\SEEKER\IPC$             seeker server (Samba, Ubuntu)
        \\SEEKER\MYTHTV          Home Directories
    \\NESSUS
        \\NESSUS\rodsmit          IPC Service (seeker server (Samba, Ubuntu))
        \\NESSUS\hp4000            Home Directories
        \\NESSUS\Epson_RX500       HP4000 via Ethernet
        \\NESSUS\IPC$              EPSON Stylus Photo RX500
        \\NESSUS\cf                IPC Service (Nessus)
        \\NESSUS\floppy            Epson RX500 CF port
                                Floppy Drive

```

Большая часть информации, выводимой командой `smbtree`, говорит сама за себя и включает сведения о рабочей группе или домене (в нашем примере это RINGWORLD), серверах (WEMBLETH, SEEKER и NESSUS) и доступных для просмотра общих ресурсах (а также описания, которые задаются с помощью опции `comment`). Ресурс `IPC$` используется для выполнения фоновых задач и обычно скрыт от пользователей, однако `smbtree` показывает информацию о нем.

Можно использовать различные опции команды `smbtree`, влияющие на ее работу. В таблице 1 перечислены наиболее полезные из них. За дополнительной информацией о более продвинутых опциях обратитесь к `man`-странице `smbtree`.

**Таблица 1. Опции команды `smbtree`**

Опция	Действие
<code>-b</code> или <code>--broadcast</code>	Выполняет широковещательные запросы, а не запросы к локальному главному браузеру, для получения списка просмотра.
<code>-D</code> или <code>--domains</code>	Выводит список известных доменов; не выводит информацию о доступных серверах и общих ресурсах.
<code>-S</code> или <code>--servers</code>	Выводит список доменов и серверов; не выводит информацию об общих ресурсах.
<code>-d level</code> или <code>--debuglevel=level</code>	Устанавливает уровень детализации для записи в log-файлы; по умолчанию используется значение 0 (минимум информации). Для поиска и устранения проблем с сетью увеличьте уровень детализации.
<code>-N</code> или <code>--no-pass</code>	Отключает запрос пароля, что может отразиться на выводимой информации.
<code>-U username</code> или <code>--user=username</code>	Задает имя пользователя, используемое для доступа к общему ресурсу. Также можно добавить знак процента (%) и пароль; например, опция <code>teela%lucky</code> задает имя пользователя <code>teela</code> и пароль <code>lucky</code> (заметим, что <code>lucky</code> – это очень слабый пароль; позаботиться об этом должна не Teela).
<code>-S</code> или <code>--servers</code>	Выводит список доменов и серверов; не выводит информацию об общих ресурсах.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): NetBIOS and WINS \(EN\)](#).

- Прочтайте о настройке Samba в качестве контроллера домена в статье Родерика Смита "[Изучаем Linux, 302 \(смешанные среды\): управление доменом](#)" (developerWorks, август 2011 г.).
- Прочтайте о [параметрах просмотра сетевого окружения и WINS-сервера](#) (EN) в главе "Network Browsing" HOWTO-подборки, посвященной Samba.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.

## Получить продукты и технологии

- Исходный код утилиты [http://ftp.samba.org/pub/unpacked/standalone\\_projects/source4/libcli/wrepl/](http://ftp.samba.org/pub/unpacked/standalone_projects/source4/libcli/wrepl/) доступен на [Web-сайте Samba](#).

# Изучаем Linux, 302 (смешанные среды): Интеграция с Active Directory

*Интеграция и работа в доменной среде Active Directory*

[Трейси Бост](#), консультант и преподаватель, Свободный писатель

**Описание:** Начиная с операционной системы Windows 2000, компания Microsoft предлагает собственную службу каталогов под названием Active Directory, позже переименованную в Active Directory Domain Services (AD DS). Службы AD DS используют преимущества таких популярных протоколов, как LDAP (для управления ресурсами) и Kerberos (для аутентификации), и тесно интегрированы со службой доменных имен DNS. Если вы используете среду AD DS, то интеграция в нее сервера Linux поможет централизовать управление идентификационной информацией пользователей и обеспечить поддержку файловых служб и служб печати Linux. Однако при такой интеграции могут возникнуть серьезные проблемы. К счастью, в Samba имеется решение для интеграции Linux с AD DS, не требующее внесения изменений в службу каталогов. [Примечание. Мы изменили заголовок [листинга 4](#) и облегчили раздел "[Использование команды net](#)" на основании отзывов читателей.]

**Дата:** 01.08.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно

дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели экзаменов сертификации LPIC-3.*

В этой статье рассматриваются следующие темы:

- Доменные службы Active Directory (AD DS).
- Механизмы взаимодействия Samba со службами AD DS.
- Настройка Samba для работы со службами AD DS.
- Взаимодействие со службами AD DS.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 314.3 темы 314. Цель имеет вес 2.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и в общих чертах понимает назначение Samba (о чем рассказывалось в предыдущей статье "[Изучаем Linux, 302: основные принципы](#)"). Для выполнения примеров этой статьи на вашем компьютере должно быть инсталлировано программное обеспечение Samba. Кроме того, необходимо иметь доступ к компьютеру под управлением операционной системы Windows Server с запущенными службами AD DS.

## Что такое Active Directory

Если в вашей сети имеются множество Windows-клиентов или уже работают службы AD DS, то стоит подумать об интеграции Linux-серверов в доменное окружение Active Directory. Службы AD DS используются в качестве средства аутентификации и службы каталога со временем операционной системы Windows 2000. Существенным отличием от ранее используемых основного и резервного контроллеров домена стало то, что в AD DS используются контроллеры домена, которые можно реплицировать между собой.

Несмотря на существование различных методов интеграции Linux-серверов в домен AD DS, Samba позволяет упростить управление и настройку, не требуя внесения изменений в схему AD DS или инсталляции дополнительного программного обеспечения на компьютере с Windows Server. Сервер Samba не может работать в качестве контроллера домена Active Directory, но может работать в качестве рядового (member) сервера, взаимодействуя со службами AD DS.

Службы AD DS основаны на следующих стандартах сети Интернет:

- Система доменных имен DNS (Domain Name System), используемая для разрешения имен.
- Протокол Kerberos версии 5, предназначенный для выполнения аутентификации.
- Протокол LDAP (Lightweight Directory Access Protocol) версии 3, обеспечивающий работу служб каталогов.

## Протокол LDAP 3

Протокол LDAP появился как ответ на потребность в более легкой службе каталогов по сравнению с его предшественником, протоколом X.500. Первая версия LDAP была выпущена в 1993 году, и с тех пор этот протокол был существенно усовершенствован. На сегодняшний день он является Интернет-стандартом де-факто для служб каталогов.

Microsoft заявляет о стопроцентном соответствии спецификации LDAP. В таблице 1 перечислены RFC-документы, содержащие расширенную информацию о чтении и выполнении операций в LDAP.

**Таблица 1. RFC-документы, содержащие информацию о LDAP**

<b>Номер RFC</b>		<b>Поддерживается</b>
2251	LDAP v3	Начиная с Windows 2000
2252	Attribute Syntax Definitions	Начиная с Windows 2000
2253	UTF-8 String Representation of Distinguished Names	Начиная с Windows 2000
2254	LDAP Search Filters Using Strings	Начиная с Windows 2000
2255	The LDAP URL Format	Начиная с Windows 2000
2256	The X.500 User Schema for use with LDAPv3	Начиная с Windows 2000
2829	Authentication Methods for LDAP	Начиная с Windows 2000
2830	Extension for Transport Layer Security	Начиная с Windows 2000
2589	Extensions for Dynamic Directory Services	Начиная с Windows Server 2003
2798	Defines the <code>inetOrgPerson</code> LDAP Object Class	Начиная с Windows Server 2003
2831	Using Digest Authentication as an SASL Mechanism	Начиная с Windows Server 2003
2891	LDAP Control Extension for Server Side Sorting of Search Results	Начиная с Windows Server 2003

### Протокол Kerberos 5

Протокол Kerberos был разработан Массачусетским технологическим институтом в качестве сетевого протокола аутентификации в то время, когда в глобальных и локальных сетях остро всталася проблема безопасности. Этот протокол обеспечивает стойкую криптографическую защиту, позволяющую клиентам и серверам передавать свои учетные данные друг другу. Для такого обмена информацией используются мандаты доступа (tickets) и аутентификаторы (authenticators).

В AD DS протокол Kerberos версии 5 используется для аутентификации пользователей. Контроллер домена Active Directory выступает в роли центра распределения ключей аутентификации клиентов Kerberos.

### Система имен DNS

Службы AD DS тесно интегрированы с системой имен DNS, которая используется в следующих целях:

- Поиск контроллеров домена AD DS.
- Описание иерархической организационной структуры с использованием имен доменов, входящих в ее состав.
- Разрешение имен для области контроллера домена и для доменов AD DS.

Следует помнить о том, что сама по себе служба AD DS не является DNS-сервером и не выполняет его задач. Как правило, DNS-сервер хранит записи о зонах и доступных ресурсах, тогда как AD DS использует то же самое пространство имен для хранения объектов домена. В таблице 2 приведено сравнение типовых ролей, выполняемых системой DNS и службами AD DS.

**Таблица 2. Роли DNS и AD DS**

<b>DNS</b>	<b>AD DS</b>
Хранит имена доменов в качестве записей ресурсов DNS	Хранит DNS-имена в качестве объектов ( <code>dnsZone</code> )
Хранит имена компьютеров в качестве записей ресурсов DNS	Хранит имена компьютеров в качестве записей объектов

**Служебная запись**, или SRV-запись (service record) DNS содержит информацию, определяющую местоположение серверов для указанных служб. Для правильной работы служб AD DS серверы DNS должны поддерживать записи описания ресурсов (RR). Запись SRV RR сопоставляет имя службы и имя сервера, на котором она запущена. Контроллеры домена и клиенты служб AD DS используют SRV-записи для определения IP-адресов контроллеров домена.

### Настройка поддержки служб AD DS в Samba

Прежде чем Linux-сервер сможет взаимодействовать со службами AD DS, необходимо убедиться в том, что установленное программное обеспечение Samba поддерживает протоколы LDAP и Kerberos. Если вы инсталлировали готовую скомпилированную версию Samba, то, вероятно, в ней уже присутствует поддержка LDAP и Kerberos 5. Если же вы инсталлировали Samba из исходного кода, то необходимо убедиться в том, что была включена поддержка библиотек `kbr5` и `ldap`. Как правило, для этого достаточно отредактировать файл заголовков `include/config.h` перед запуском команды `make` следующим образом:

```
#define HAVE_KRB5 1  
#define HAVE_LDAP 1
```

В зависимости от дистрибутива Linux имена библиотек могут отличаться.

После инсталляции Samba на ваш компьютер, можно с помощью системного демона `smbd` выяснить, какие функции она поддерживает:

### Листинг 1. Часть списка функций Kerberos 5 и LDAP, поддерживаемых Samba

```
[tbost@samba3 ~]$ smbd -b | grep KRB  
HAVE_KRB5_H  
HAVE_KRB5_LOCATE_PLUGIN_H  
HAVE_ADDRTYPE_IN_KRB5_ADDRESS  
HAVE_DECL_KRB5_AUTH_CON_SET_REQ_CKSUMTYPE  
HAVE_DECL_KRB5_GET_CREDENTIALS_FOR_USER  
HAVE_INITIALIZE_KRB5_ERROR_TABLE  
HAVE_KRB5  
HAVE_KRB5_AUTH_CON_SETUSERUSERKEY  
HAVE_KRB5_AUTH_CON_SET_REQ_CKSUMTYPE  
HAVE_KRB5_C_ENCTYPE_COMPARE  
HAVE_KRB5_C_VERIFY_CHECKSUM  
HAVE_KRB5_DEPRECATED_WITH_IDENTIFIER  
HAVE_KRB5_ENCRYPT_BLOCK  
HAVE_KRB5_ENCRYPT_DATA  
HAVE_KRB5_ENCTYPE_TO_STRING  
....
```

```
[tbost@samba3 ~]$ smbd -b | grep LDAP  
HAVE_LDAP_H  
HAVE_LDAP  
HAVE_LDAP_ADD_RESULT_ENTRY  
HAVE_LDAP_INIT  
HAVE_LDAP_INITIALIZE  
HAVE_LDAP_SASL_WRAPPER  
HAVE_LDAP_SET_REBIND_PROC  
HAVE_LIBLDAP
```

## LDAP\_SET\_REBIND\_PROC\_ARGS

В листинге 1 показана поддержка библиотек `krb5` и `ldap` для дистрибутива Fedora. В зависимости от вашего дистрибутива вывод этих команд может отличаться. Несмотря на это, убедитесь, что в выводе команд присутствуют, как минимум, строки `HAVE_KRB5_H` и `HAVE_LDAP_H`.

## Kerberos и NTP

Протокол Kerberos предполагает, что сервер Samba самостоятельно синхронизирует время с доменом. Обычно службы AD DS используют в качестве сервера времени контроллеры домена. Вы можете настроить протокол NTP (протокол сетевого времени) на компьютере Linux, указав в качестве сервера времени контроллеры домена Windows.

## Samba и Kerberos

Samba может использовать протокол Kerberos для аутентификации пользователей в домене AD DS. Для настройки Samba найдите в директории `/etc` файл `krb5.conf`, в который нужно будет внести некоторые изменения. Как минимум, в разделе `realms` этого файла нужно указать имя домена, а также полное доменное имя (FQDN) сервера в домене Windows, выполняющего аутентификацию для служб AD DS, как показано в листинге 2:

### Листинг 2. Настройка конфигурации в файле `krb5.conf`

```
[realms]
```

```
LPIC302.LOCAL= {
    kdc = wins3.lpic302.local
    admin_server =wins3.lpic302.local
    default_domain = LPIC302.LOCAL
}
```

В листинге 2 показан простой пример конфигурации с использованием имени `LPIC302.LOCAL` в качестве имени домена AD DS. Имя домена должно быть набрано заглавными буквами, иначе Kerberos не сможет подключиться. Директива `kdc` указывает на контроллер AD DS с именем `wins3.lpic302.local`. Кроме того, контроллер домена указан в директиве `admin_server`. Параметр `default_domain` определяет для Kerberos имя домена, которое будет использоваться по умолчанию в тех случаях, когда оно не будет указано пользователем.

## Демон Winbind

Демон Winbind облегчает процесс аутентификации пользователей в домене AD DS. По существу, нужно настроить систему PAM-модулей на использование модуля `pam_winbind`, как показано в листинге 3.

### Листинг 3. Настройка PAM на использование `pam_winbind`

```
auth      sufficient  pam_winbind.so
auth      sufficient  pam_unix.so use_first_pass
auth      required    pam_stack.so service=system-auth
auth      required    pam_nologin.so
```

```
account      sufficient  pam_winbind.so
account      required    pam_stack.so service=system-auth
password     required    pam_stack.so service=system-auth
session      required    pam_stack.so service=system-auth
session      optional   pam_console.so
```

В листинге 3 показан измененный файл system-auth из директории /etc/pam.d в дистрибутиве на базе Fedora. В зависимости от вашего дистрибутива имя файла аутентификации может отличаться. Обычно именем файла является *services* или *login*.

Место размещения строки с модулем pam\_winbind.so играет важную роль. Если вы предполагаете, что пользователи в основном будут авторизоваться с использованием учетных записей AD DS, а не локального файла passwd, то строка с модулем pam\_winbind.so должна стоять первой. В противном случае вы обнаружите в файле auth.log множество сообщений о неудачных попытках входа.

### Служба Name Service Switch

Служба Name Service Switch (NSS) предоставляет стандартный механизм, с помощью которого Linux-компьютеры могут взаимодействовать с распространенными службами аутентификации. При использовании этих служб Linux-компьютер обращается к файлу /etc/nsswitch.conf. Для того чтобы Linux-компьютер мог использовать Winbind для аутентификации пользователей, необходимо изменить этот файл следующим образом.

В следующем коде жирным шрифтом выделено добавление поддержки Winbind, которое позволяет пользователем проходить аутентификацию в базе данных Kerberos 5 AD DS с помощью Winbind.

```
passwd: files winbind
group:   files winbind
```

### Файл smb.conf

Как можно было предположить, для работы Samba в домене AD DS необходимо выполнить определенные настройки в файле smb.conf. В простейшем случае нужно задать значения для параметров **realm** и **security**, как показано в листинге 4.

### Листинг 4. Настройка параметров файла smb.conf

```
[global]
realm = lpic302.LOCAL
security = ADS
password server = wins.lpic302.local
workgroup = lpic302
winbind use default domain = yes
idmap uid = 10000-20000
idmap gid = 10000-20000
winbind enum users = yes
winbind enum groups = yes
```

В листинге 4 параметру **realm** присваивается имя домена *lpic302.local*. Параметр **security** установлен в **ADS**. Значение ADS означает, что Samba будет работать в режиме безопасности AD DS Service. Чтобы избавиться от необходимости дополнять имена пользователей и других ресурсов доменным суффиксом при обращении к ним, можно задать

параметр `windbind use default domain = yes`. Например, вместо полного имени пользователя LPIC302.LOCAL/tbost можно будет указать просто имя `tbost` – Winbind сам определит, что именем домена является LPIC302.LOCAL.

## Взаимодействие с AD DS

После выполнения всех настроек, перезапуска Samba и запуска демона Winbind можно начинать работу с AD DS.

### Использование команды `net`

Утилита `net` чрезвычайно полезна для администраторов Samba. Если вам приходилось работать с командой `net` в Windows, то многие ее функции и параметры будут вам знакомы. При работе с AD DS используется команда `net ADS`. Первое, с чего следует начать – это присоединить компьютер к домену:

```
[tbost@samba3 ~]$ sudo net ADS join -U Administrator%password  
[tbost@samba3 ~]$ sudo net ADS testjoin  
[tbost@samba3 ~]$ sudo net ADS join -U Administrator createcomputer="ACCOUNTING/Servers"
```

Первая команда `net` в этом примере используется для присоединения компьютера к домену. В качестве альтернативного способа можно не использовать строку `%password` и ввести пароль учетной записи администратора Windows при получении запроса. Вторая команда выполняет проверку успешного присоединения сервера к домену. Третья команда создает (или перемещает объект из организационного подразделения Computers, используемого по умолчанию) учетную запись сервера Samba в AD DS в организационном подразделении ACCOUNTING/Servers, которое должно существовать в каталоге Active Directory перед выполнением этой команды. Если вы хотите узнать больше о команде `net`, то обратитесь к ее онлайновой man-странице, которая содержит много полезной информации; кроме того, можно выполнить команду `net help ADS`, как показано в листинге 5.

### Листинг 5. Справочная информация, выводимая командой `net help ADS`

```
[tbost@samba3 ~]$ net help ADS  
Usage:  
net ads info  
    Display details on remote ADS server  
net ads join  
    Join the local machine to ADS realm  
net ads testjoin  
    Validate machine account  
net ads leave  
    Remove the local machine from ADS  
net ads status  
    Display machine account details  
net ads user  
    List/modify users  
net ads group  
    List/modify groups  
net ads dns  
    Issue dynamic DNS update  
net ads password  
    Change user passwords  
net ads changetrustpw  
    Change trust account password  
net ads printer  
    List/modify printer entries  
net ads search
```

```
Issue LDAP search using filter
net ads dn
    Issue LDAP search by DN
net ads sid
    Issue LDAP search by SID
net ads workgroup
    Display the workgroup name
net ads lookup
    Find the ADS DC using CLDAP lookups
net ads keytab
    Manage local keytab file
net ads gpo
    Manage group policy objects
net ads kerberos
    Manage kerberos keytab
```

## Взаимодействие с помощью wbinfo

Утилита **wbinfo** из состава демона Winbind используется для получения информации о ресурсах AD DS:

```
[tbost@samba3 ~]$ wbinfo -p
[tbost@samba3 ~]$ wbinfo -u
[tbost@samba3 ~]$ wbinfo -g
```

В этом примере команда **wbinfo** используется для получения информации о домене.

Команда **wbinfo -p** посылает запрос демону Winbind, чтобы проверить, что он запущен.

Команда **wbinfo -u** возвращает список всех пользователей домена, а команда **wbinfo -g** – список всех групп домена. За дополнительной информацией о команде **wbinfo** обратитесь к ее справочному руководству.

## Управление списками контроля доступа с помощью smbcacls

Если вы знакомы с командами **setfacl** и **getfacl**, то у вас не будет особых проблем с командой **smbcacls** из клиентского пакета Samba. С помощью этой команды можно изменять группы и владельцев файлов и директорий, а также управлять разрешениями списков контроля доступа для общих ресурсов, расположенных в домене на компьютере под управлением Windows Server:

```
[tbost@samba3 ~]$ sudo smbcacls -G LPIC302.LOCAL\accounting \
//wins2.lpic302.local/budget private.doc
```

В этом примере с помощью команды **smbcacls** мы изменяем группу-владельца файла *private.doc*, хранящегося в общей директории *budget* на компьютере под управлением Windows Server, на группу *accounting* домена AD DS. Дополнительную информацию об этой команде можно получить, выполнив команду **smbcacls --help**.

## Ресурсы

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Integration with Active Directory \(EN\)](#).
- Узнайте больше об [интеграции Samba, Active Directory и LDAP \(EN\)](#) из руководства Samba 3.x.
- Узнайте, как [Samba использует DNS с AD DS \(EN\)](#) из руководства Samba.

- Узнайте, как настроить Samba для [аутентификации в AD DS](#) (EN).
- Узнайте, как [Winbind взаимодействует с AD DS](#) EN) из руководства Samba.
- Узнайте больше о протоколе[LDAPv3](#) (EN) из материалов рабочей группы LDAPv3.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.
- [Материалы для подготовки к исправленным экзаменам LPIC](#) (EN) содержат список дополнительных ресурсов института LPI, которые помогут вам при подготовке к получению сертификата.

## Изучаем Linux, 302 (смешанные среды): Работа с Windows-клиентами

*Использование Windows-клиентов для работы с серверами Samba*

[Родерик Смит \(Roderick Smith\)](#), автор и консультант, IBM

**Описание:** Несмотря на то, что клиентами Samba могут быть компьютеры под управлением UNIX и Linux, большинство из них использует операционную систему Windows®, поэтому нужно знать о том, как применять функционал ОС Windows, чтобы подключаться к серверам Samba. Например, с помощью определенных команд Samba на Linux-компьютере можно выявлять и устранять возникающие проблемы.

[Больше статей из этой серии](#)

**Дата:** 13.09.2012

**Уровень сложности:** средний

### Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Возможности Windows-клиентов.
- Использование списков просмотра в ОС Windows.
- Создание общих файловых ресурсов и ресурсов печати в ОС Windows.

- Использование утилиты `smbclient` для тестирования.
- Использование команды `net` операционной системы Windows.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 314.4 темы 314. Цель имеет вес 4.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. В частности, предполагается, что читатель умеет работать с командной строкой Linux и знает основы конфигурирования Samba. Необходимо уметь загружать и работать на компьютере с Windows и знать основные функции этой операционной системы, включая работу с окном командной строки. Все примеры этой статьи выполнены на компьютерах под управлением ОС Windows 7, однако в более ранних версиях Windows все действия будут аналогичными.

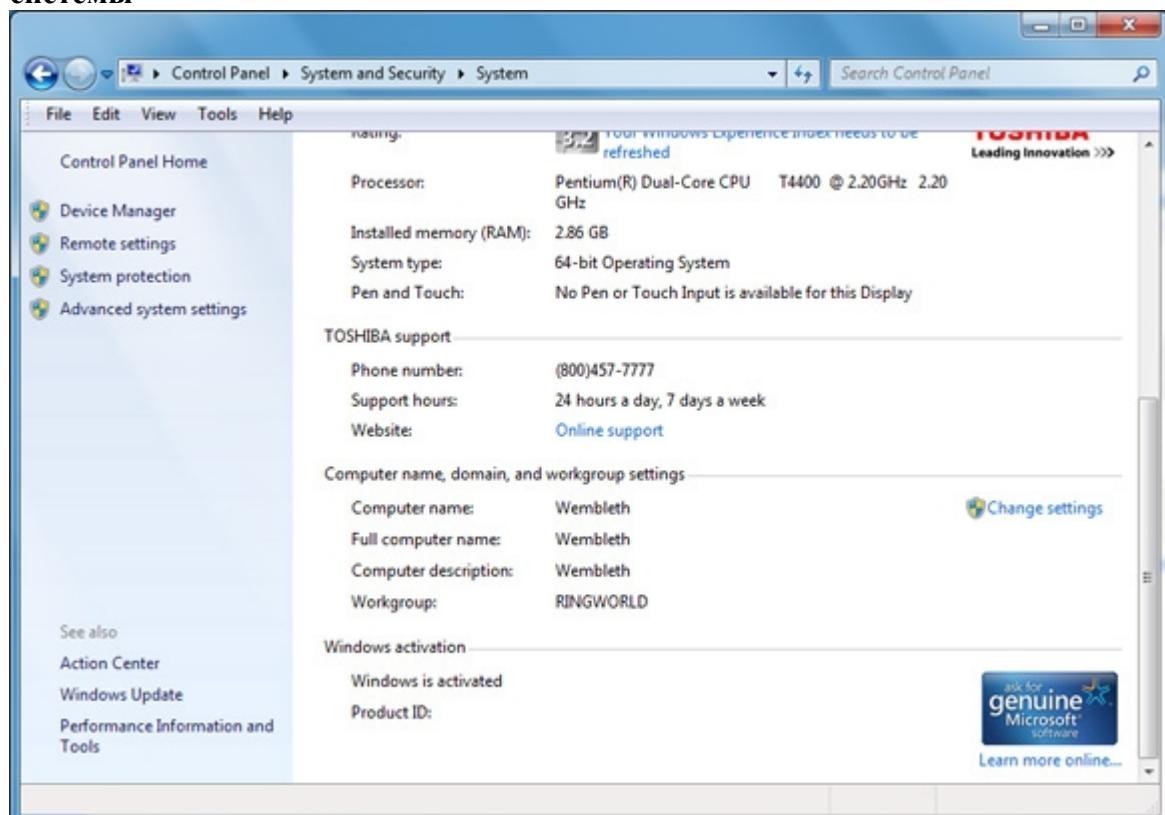
## Настройка параметров SMB/CIFS в Windows

Чтобы можно было получить доступ к общим ресурсам сервера Samba, на нем необходимо настроить ряд определенных параметров (например, указать имя рабочей группы или домена). Точно так же, необходимо выполнить ряд настроек и на Windows-клиенте с помощью следующих действий:

1. В панели управления щелкните **System and Security**, а затем **System**.

Вы должны увидеть окно с информацией о компьютере (имя и рабочая группа/домен), изображенное на рисунке 1.

**Рисунок 1. Панель управления Windows, позволяющая изменять свойства системы**

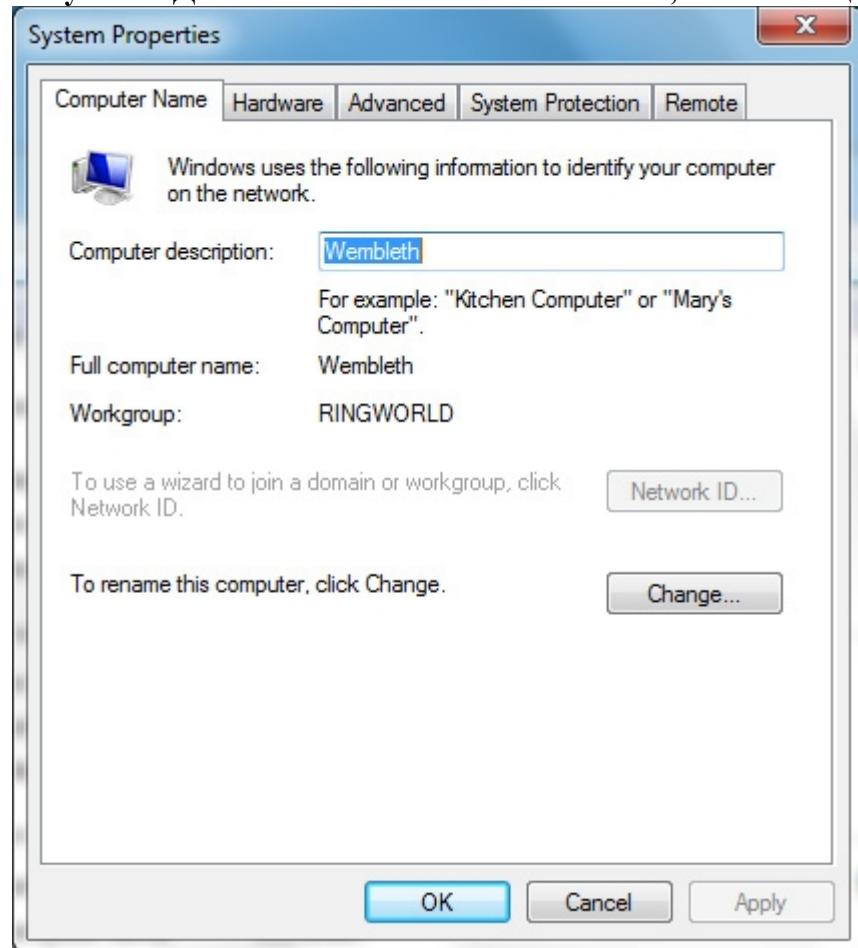


2. В области **Computer name, domain, and workgroup settings** щелкните **Change**

**settings.**

В результате откроется диалоговое окно **System Properties**, изображенное на рисунке 2. В этом окне можно изменить NetBIOS-имя компьютера и принадлежность к домену или рабочей группе.

**Рисунок 2. Диалоговое окно свойств системы, позволяющее задавать имена**



3. Введите в поле **Computer description** понятное описание компьютера (можете воспользоваться подсказкой, расположенной чуть ниже этого поля).
4. После того, как вы указали NetBIOS-имя и рабочую группу или домен (отображаются в поле **Full computer name**), нажмите кнопку **OK**. Если вы хотите изменить эту информацию, то нажмите кнопку **Change**.

После нажатия кнопки **Change** откроется диалоговое окно **Computer Name/Domain Changes**, изображенное на рисунке 3.

**Рисунок 3. В диалоговом окне Computer Name/Domain Changes можно изменить имя компьютера и его принадлежность к домену или рабочей группе**



5. В поле **Computer name** введите NetBIOS-имя компьютера.

Это поле особенно важно, если вы планируете предоставлять на этом компьютере общий доступ к файлам или принтерам; если же компьютер будет работать только как клиент, то это поле не так важно. Параметры, указанные в этом диалоговом окне, влияют только на NetBIOS-имена.

6. При желании можно изменить DNS-имя компьютера, нажав кнопку **More**, расположенную в области **Full computer name**.

Примечание: все изменения являются локальными и влияют лишь на то, как компьютер объявляет о себе другим компьютерам, но не влияют на имя, используемое другими компьютерами при обращении к нему. Если вы хотите изменить DNS-имя компьютера, то необходимо выполнить соответствующие настройки на DNS-сервере вашей сети.

7. Выберите требуемое значение **Domain** или **Workgroup** и укажите в соответствующем поле имя вашего локального домена или рабочей группы.

Примечание: опция **Domain** может быть недоступной в некоторых версиях Windows, например, в Windows 7 Home Premium. Если вы хотите присоединить такой компьютер к домену, то необходимо обновить установленную на нем версию операционной системы Windows.

8. После внесения всех изменений нажмите **OK**.

Операционная система сообщит о необходимости перезагрузки. Если вы присоединяете компьютер к домену, то вам также потребуется ввести необходимые учетные данные. Более подробно эта тема рассматривается в статье "[Изучаем Linux, 302 \(смешанные среды\): Управление доменом](#)".

После перезагрузки компьютера Windows будет использовать новые NetBIOS-имя и рабочую группу/домен компьютера, и вы сможете выполнить все действия, описанные далее в этой

статье. Если что-то не будет работать, то проверьте конфигурацию вашего компьютера – возможно, вы сделали опечатку при вводе имени рабочей группы или домена, хотя это может быть следствием и более серьезной проблемы, например, обрывом сетевого кабеля.

## Доступ к файловым ресурсам из Windows

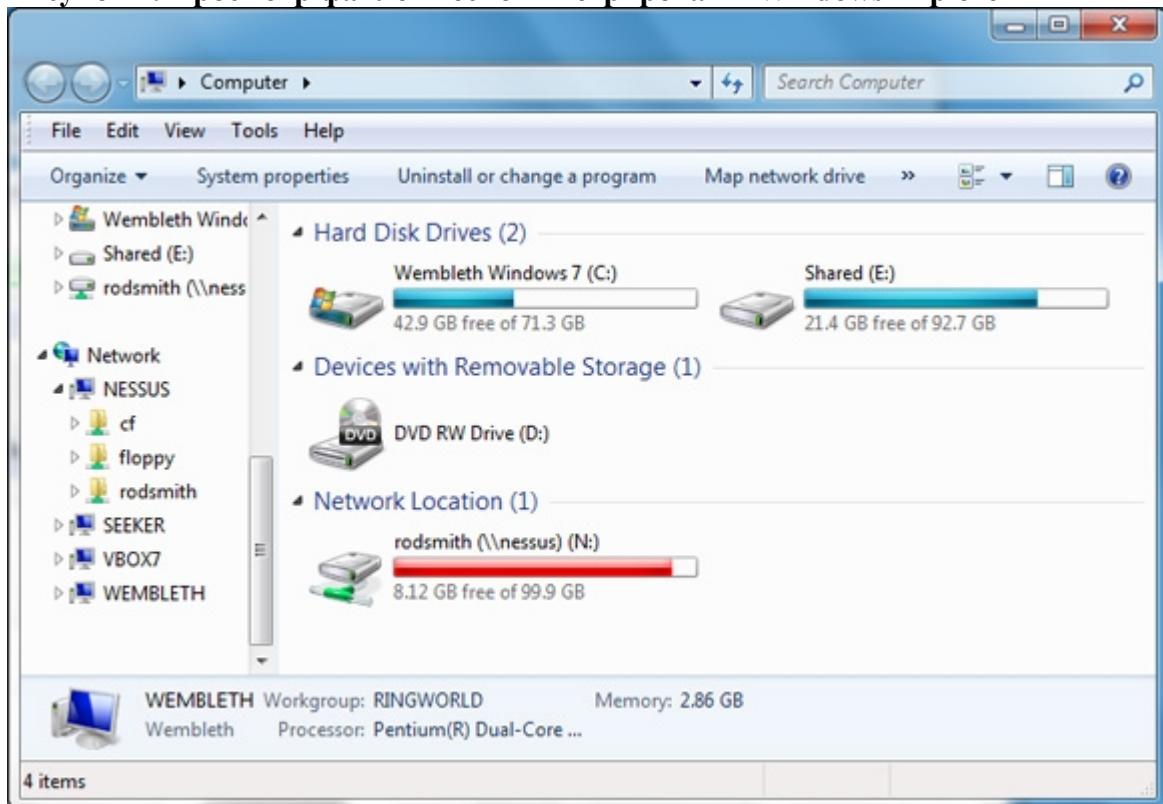
В идеале (и зачастую в действительности) для работы настроенного сервера Samba с Windows-клиентами не требуется предпринимать какие-либо дополнительные действия. Таким образом, мы будем предполагать, что все прекрасно работает. Исходя из этого предположения, я объясню, как находить общие ресурсы Samba и напрямую вводить их адреса Universal Resource Identifier (URI-адреса), работая в Windows. Если у вас возникнут какие-либо проблемы, то вам поможет раздел этой статьи [Поиск и устранение проблем с помощью Linux-клиента](#).

### Поиск общего ресурса Samba

Как рассказывалось в статье "[Изучаем Linux, 302 \(смешанные среды\): Файловые службы](#)", протокол SMB/CIFS реализует простой способ обнаружения в сети общих файлов и принтеров. Детали реализации этого способа зависят от пользовательского интерфейса клиента и, фактически, в Windows имеется несколько способов поиска общих ресурсов. Одним из них является встроенный файловый менеджер операционной системы:

1. Откройте проводник Windows (Windows Explorer), щелкнув правой кнопкой мыши на значке **Computer** и выбрав пункт **Explore**, как показано на рисунке 4.

**Рисунок 4. Просмотр файлов тесно интегрирован в Windows Explorer**



Значок Network в левой панели обеспечивает доступ к спискам просмотра сети. Щелкните на имени компьютера в левой части окна, чтобы увидеть, какие общие ресурсы он содержит, а затем щелкните на нужный ресурс, чтобы открыть его.

2. После того, как вы открыли общий ресурс и нашли нужный файл, вы можете дважды щелкнуть на нем, чтобы открыть с помощью программы по умолчанию, переместить его в корзину для удаления или выполнить любые другие действия, которые можно

выполнять для обычного локального файла, хранящегося на жестком диске.

Такие же возможности присутствуют в диалоговых окнах **Open** и **Save As**, используемых многими приложениями для поиска файлов.

## Ввод URI-адреса

Иногда просмотр сетевого окружения не работает корректно, поскольку в сети могут возникать различные проблемы, связанные с просмотром, или общий ресурс специально может быть сделан скрытым. В таких случаях, если вы знаете точное DNS- или NetBIOS-имя сервера, а также имя общего ресурса, то можете ввести его URI-адрес непосредственно в адресную строку Windows Explorer. Для этого выполните следующие действия:

1. Щелкните на адресной строке.

Необходимо щелкнуть справа от существующих элементов, поскольку нам не нужно выбирать ни один из них. Поле изменит вид, отображая элементы в форме URI-адреса.

2. Введите полный путь к требуемому общему ресурсу в виде `\\ИМЯ_КОМПЬЮТЕРА \ИМЯ_РЕСУРСА`, например, `\NESSUS\RODSMITH` для общего ресурса RODSMITH, расположенного на компьютере NESSUS.

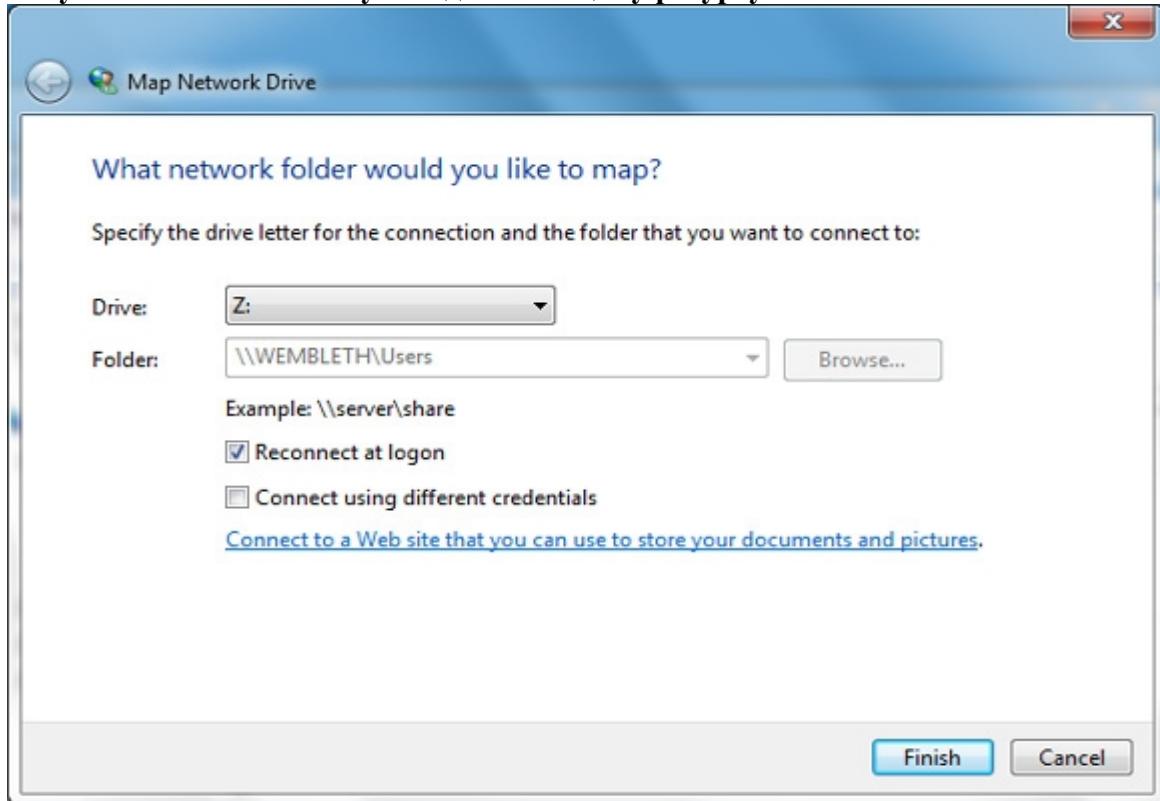
Теперь содержимое указанного общего ресурса будет отображаться в окне Windows Explorer. Вы сможете управлять файлами и папками точно так же, как и при их поиске через просмотр сетевого окружения.

## Подключение сетевых ресурсов

Иногда чтобы не искать каждый раз общий ресурс в сети или не вводить его URI-адрес, удобнее присвоить ему букву диска. На рисунке 4 изображен общий ресурс `\NESSUS\RODSMITH`, которому уже присвоена буква диска N.

Для присвоения буквы диска общему ресурсу найдите его в сетевом окружении, щелкните на нем правой кнопкой мыши и выберите пункт **Map Network Drive**, в результате чего откроется одноименное диалоговое окно, изображенное на рисунке 5. В этом окне вы можете выбрать букву диска для общего ресурса, а также указать различные дополнительные параметры, облегчающие работу.

**Рисунок 5. Назначение буквы диска общему ресурсу в Windows**



### Печать на принтеры Samba из Windows

Если принтер Samba правильно настроен, то Windows-клиенты могут отправлять на него задания без дополнительных настроек. Основной процесс включает в себя два этапа: настройка принтера в Windows и назначение ему соответствующих ресурсов печати Samba. Основная сложность состоит, пожалуй, в выборе драйвера печати, поскольку одни ресурсы Samba могут требовать использования родных драйверов Windows, другие – использования стороннего драйвера PostScript, а третьи позволяют использовать оба типа драйверов.

#### Настройка принтера

Для настройки нового принтера в Windows выполните следующие действия:

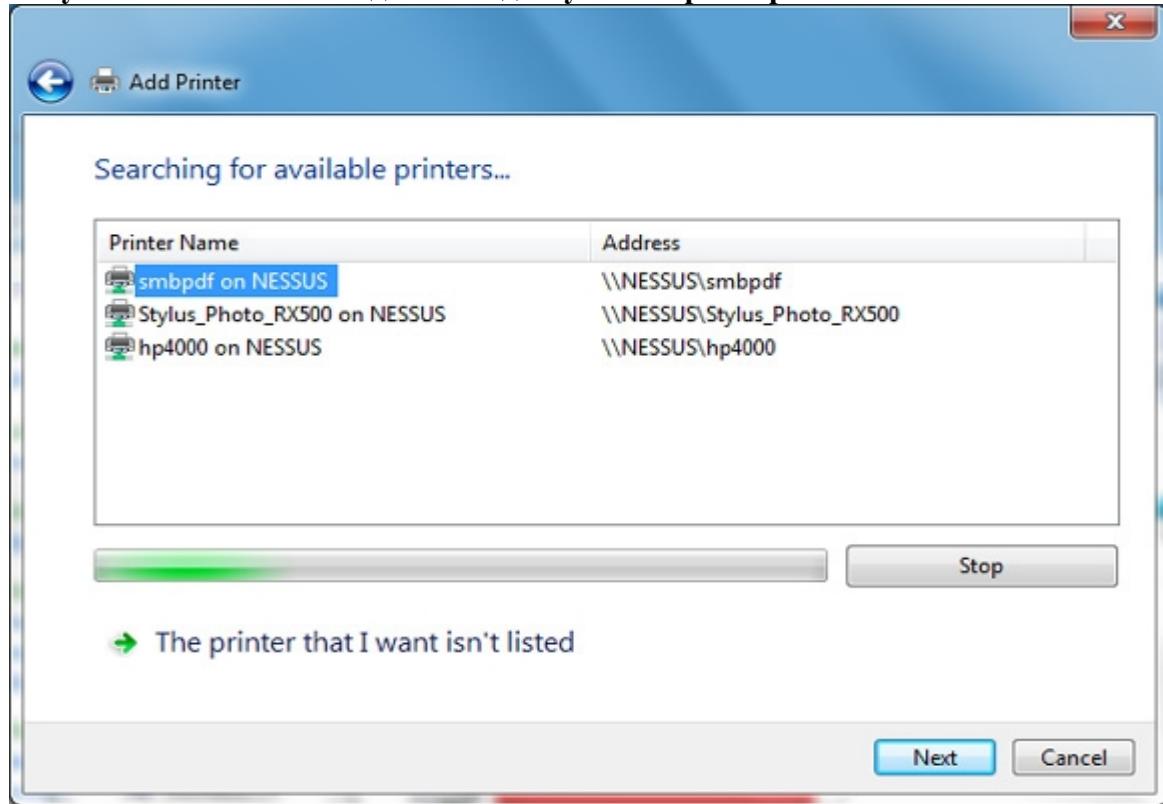
1. Откройте элемент Hardware and Sound в панели управления и щелкните **Add a Printer**.

Откроется диалоговое окно с запросом типа добавляемого принтера.

2. Выберите **Add a Network, Wireless or Bluetooth Printer**.

Windows выполнит поиск доступных принтеров и отобразит список, как показано на рисунке 6.

Рисунок 6. Поиск и вывод списка доступных принтеров в Windows



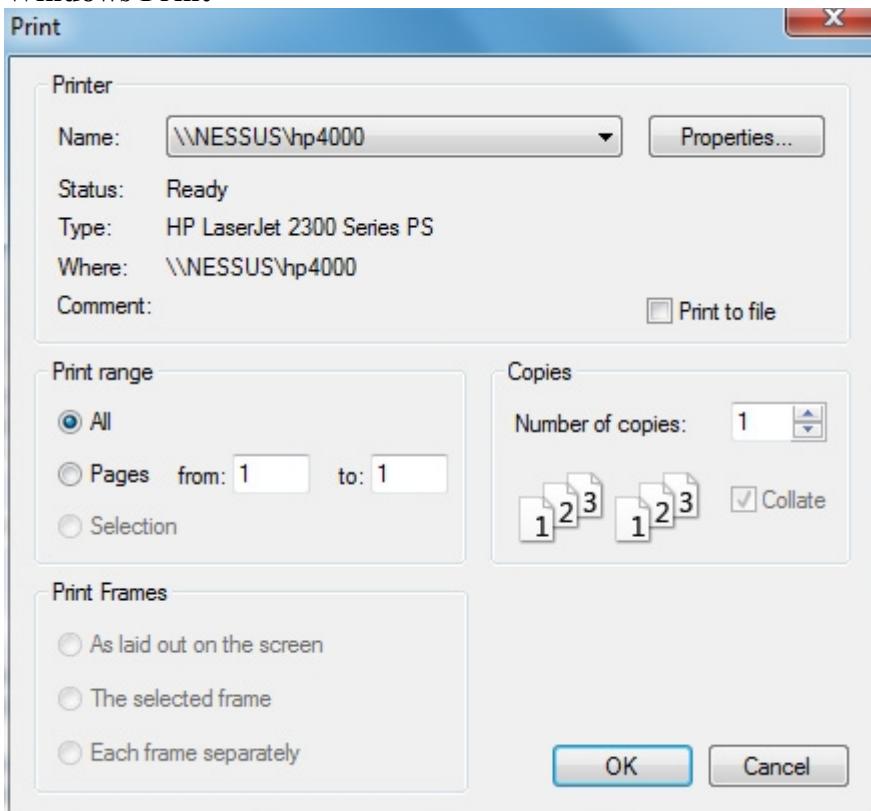
3. Выберите требуемый принтер и нажмите **Next**.

Если нужный принтер отсутствует в списке, то может потребоваться изменить его конфигурацию в Samba. Если же принтер был намеренно скрыт от просмотра в сетевом окружении, то щелкните **The printer that I want isn't listed**, чтобы ввести его URI-адрес.

4. Если сервер, на котором содержится общий ресурс печати, был настроен таким образом, чтобы передавать драйверы принтера на клиентские компьютеры, то операционная система Windows должна автоматически загрузить и установить их. В противном случае вы увидите диалоговое окно с информацией о том, что подходящий драйвер не найден, и вам придется выбрать его из списка стандартных драйверов Windows или указать драйвер производителя принтера.
5. После установки драйвера распечатайте тестовую страницу.

После успешной установки принтера можно выбирать его в диалоговом окне **Print** любого приложения, как показано на рисунке 7.

**Рисунок 7. Настроенный принтер можно выбрать в списке Name диалогового окна Windows Print**



### Выбор драйвера печати

Операционная система Linux использует в качестве стандартного устройства вывода на печать язык PostScript, и если принтеры не поддерживают PostScript, то вывод на печать преобразуется в родной язык принтера; таким образом, в общем случае можно считать, что все принтеры Samba являются PostScript-принтерами. Поэтому в Windows необходимо установить подходящий универсальный драйвер PostScript, либо выбрав его из списка в Windows, либо настроив Samba на передачу драйверов клиентам. Помимо стандартных драйверов принтеров Windows с принтерами Samba обычно хорошо работает драйвер Microsoft Publisher Color Printer из раздела **Generic**.

Тем не менее, вместо использования драйвера PostScript можно установить родной драйвер принтера, поставляемый производителем. Для этого нужно получить необходимый драйвер, а также может потребоваться настроить Samba или систему печати Common UNIX® Printing System (CUPS) так, чтобы она соответствующим образом обрабатывала входящий поток данных, а не пыталась интерпретировать его и преобразовывать в родной язык принтера, как это обычно делает CUPS.

В общих чертах отличие языка PostScript от родных драйверов заключается в следующем:

- Если язык PostScript используется для не-PostScript модели принтера, то могут не работать некоторые уникальные функции принтера, например, выбор лотка и настройка разрешения печати. Обычно доступ к таким функциям предоставляют родные драйверы принтеров.
- Использование драйвера PostScript может увеличить нагрузку на сеть при печати текстовых документов; обработка тех же документов с помощью родных драйверов принтера может оказаться менее ресурсоемкой. Этот эффект проявляется в основном при использовании недорогих струйных принтеров и не так заметен при использовании лазерных принтеров среднего и старшего класса.
- Использование драйвера PostScript увеличивает нагрузку на центральный процессор

компьютера, являющегося сервером печати. Этот эффект имеет значение в ситуациях когда если функции сервера печати выполняет устаревший компьютер или он управляет большим числом принтеров.

- Для использования родного драйвера может потребоваться изменить конфигурацию Samba или CUPS – иногда эффект от такой настройки не оправдывает затраченных усилий.

При желании вы можете протестировать несколько драйверов и определить, какой из них подходит для вашего принтера лучше всего. В некоторых ситуациях можно создать в Windows несколько принтеров для одного общего ресурса печати Samba и выбрать для них различные драйверы, и тогда пользователи смогут выбирать те драйверы, которые лучше всего подходят для определенных задач.

### Использование команды net операционной системы Windows

Хотя операционные системы Windows содержат больше графических элементов интерфейса для работы и администрирования по сравнению с Linux, в них также имеются текстовые утилиты, некоторые из которых имеют преимущества над своими графическими аналогами. Основным инструментом командной строки Windows для работы с SMB/CIFS является команда **net**, выполняющая множество задач, которые нельзя выполнить с помощью графического интерфейса. Преимущество команды **net** заключается в том, что ее можно использовать в сценариях, поэтому ее можно использовать в сетевых сценариях запуска для выполнения общих задач на всех компьютерах или для выполнения сложных действий путем запуска одной команды.

Для использования команды **net** нужно набрать эту команду и имя подкоманды, которая может иметь дополнительные параметры. В таблице 1 перечислены наиболее важные подкоманды команды **net**.

**Таблица 1. Подкоманды команды net операционной системы Windows**

Подкоманда	Действие
CONFIG	Выводит или обновляет различные сетевые параметры, например, NetBIOS-имя и рабочую группу/домен компьютера.
FILE	Выводит на файловом сервере Windows информацию об открытых клиентами общих файлах.
HELP	Выводит справочную информацию по работе с командой <b>net</b> .
SESSION	Выводит на файловом сервере Windows информацию об активных клиентских подключениях; запуск этой подкоманды с ключом /DELETE разрывает соединение.
STATISTICS	Выводит статистику о полученных и отправленных байтах, ошибках, связанных с работой клиента ( <b>NET STATISTICS WORKSTATION</b> ) или сервера ( <b>NET STATISTICS SERVER</b> ), и т. д.
TIME	Устанавливает на компьютере часы в соответствии с часами, установленными на другом компьютере.
USE	Назначает общему ресурсу букву диска Windows или выводит информацию о существующих сопоставлениях.
USER	Добавляет, удаляет или изменяет учетную запись пользователя, хранящуюся на контроллере домена.
VIEW	Выводит список компьютеров в сети или список доступных общих ресурсов указанного компьютера.

Некоторые подкоманды можно использовать для выполнения тех же действий, что и при использовании файлового менеджера. Обратите особое внимание на подкоманды **VIEW** и **USE** для просмотра файловых ресурсов. Пример их использования приведен в листинге 1.

### Листинг 1. Просмотр файловых ресурсов с помощью команды VIEW

```
> NET VIEW
```

Server Name	Remark
-------------	--------

```
-----  
-  
\NESSUS Nessus  
\SEEKER seeker server (Samba, Ubuntu)  
\VBOX7 VBox7  
\WEMBLETH Wembleth  
The command completed successfully.
```

```
> NET VIEW \NESSUS
```

```
Shared resources at \\NESSUS
```

```
Nessus
```

Share name	Type	Used as	Comment
------------	------	---------	---------

```
-----  
-  
cf Disk Epson RX500 CF port  
floppy Disk Floppy Drive  
hp4000 Print HP4000 via Ethernet  
rodsmith Disk Home Directories  
smbpdf Print PDF Generator  
Stylus_Photo_RX500 Print EPSON Stylus Photo RX500  
The command completed successfully.
```

```
> NET USE \\NESSUS\cf I:
```

```
The command completed successfully.
```

В этом примере представлен процесс просмотра сетевого окружения и назначения буквы диска I: файловому ресурсу \\NESSUS\cf в Windows. После этого можно получить доступ к этому ресурсу, просто обратившись к диску I:.

Команда **NET TIME** может служить быстрой заменой NTP-серверу. В следующем примере выполняется синхронизация часов локального компьютера с часами сервера NESSUS:

```
> NET TIME \\NESSUS /SET
```

Помните о том, что для выполнения некоторых подкоманд **net** требуются права администратора системы. Запуская эти подкоманды без прав администратора, вы получите сообщение о том, что для их выполнения недостаточно привилегий. При выполнении подкоманды **USER** также может потребоваться указать пароль учетной записи на удаленном компьютере, которую вы пытаетесь изменить.

### Использование Windows в качестве сервера

Хотя сертификация LPIC-302 и наша серия статей предполагает изучение вопросов, связанных с настройкой сервера Samba в Linux, было бы ошибкой предположить, что роль сервера SMB/CIFS может выполнять только лишь связка Linux/Samba. В конце концов, протокол был изначально разработан для обеспечения доступа к общим ресурсам компьютеров под управлением операционных систем DOS, Windows и IBM® Operating

System/2® (OS/2), а Samba появилась только через несколько лет. Поэтому сейчас я объясню, как создавать общие файловые ресурсы и ресурсы печати в Windows и использовать их на клиентах под управлением Windows или Linux.

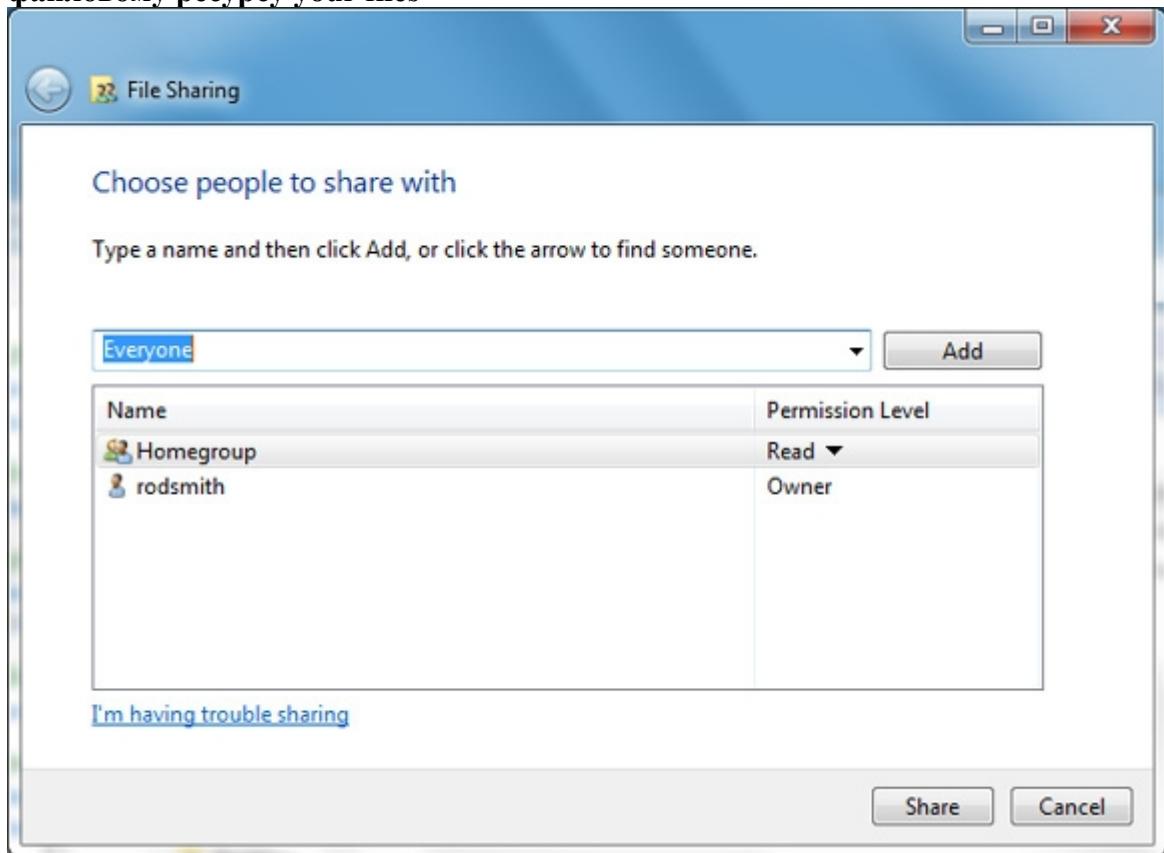
### Создание общих файловых ресурсов в Windows

На сервере Windows можно предоставить общий доступ ко всему разделу или к его отдельной поддиректории, например, к домашней директории пользователя или ее поддиректории. Для этого выполните следующие действия.

1. Найдите диск или директорию, доступ к которой вы хотите предоставить.
2. Щелкните на диске или директории правой кнопкой мыши и раскройте пункт **Share with**, содержащий несколько опций.
3. Выберите нужную опцию в списке **Share with**.

В Windows 7 имеется встроенная функция, известная как *Homegroup*, позволяющая открывать доступ к ресурсу только для заранее определенного списка пользователей. Если вы хотите создать новую группу или выбрать уже имеющуюся, то щелкните **Specific People**. В результате откроется диалоговое окно, изображенное на рисунке 8.

**Рисунок 8. Выбор пользователей, которым будет предоставлен доступ к общему файловому ресурсу your files**



4. Выберите пользователя или группу из списка **Add** или введите имя в текстовом поле.
5. Нажмите кнопку **Add**.
6. При желании можно изменить файловые разрешения для пользователей, настроив права доступа для выбранной записи в столбце **Permission Level**.

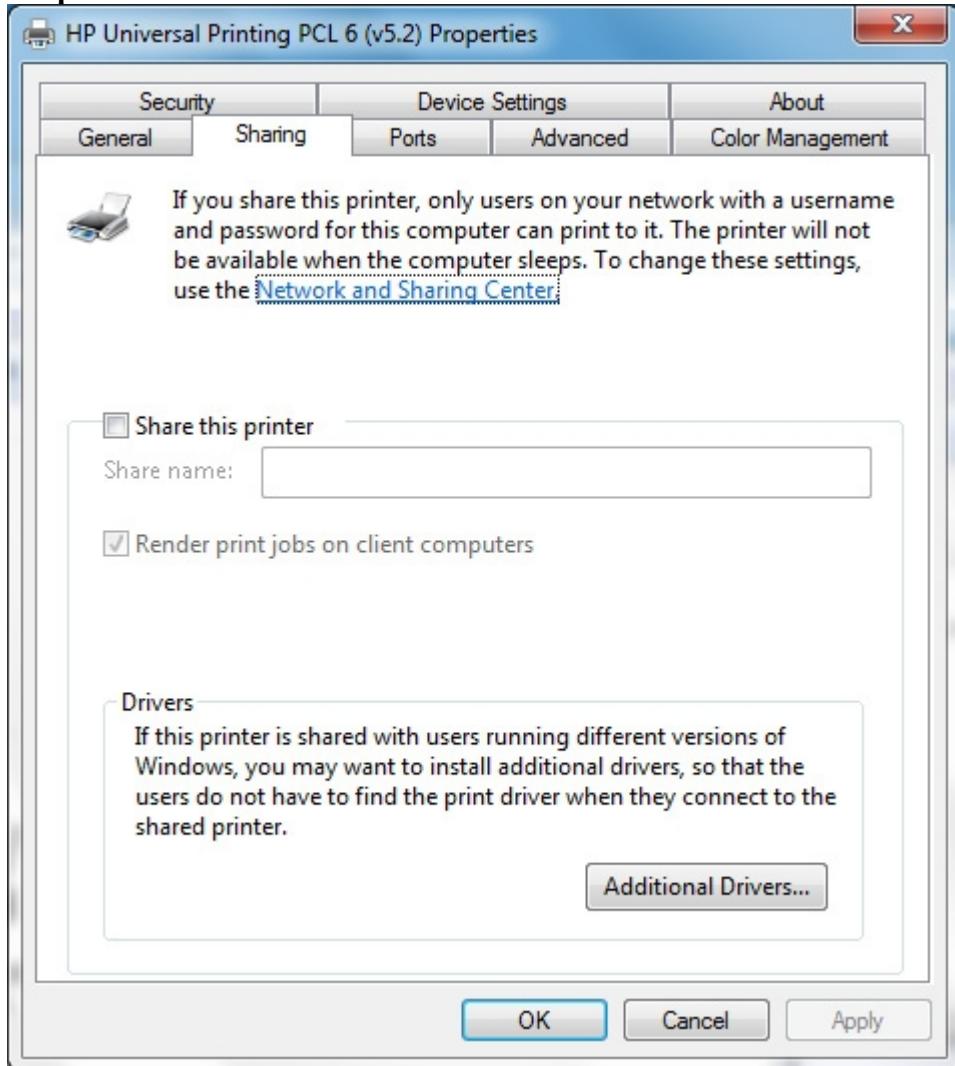
После этого можно проверить доступ к общему файловому ресурсу, попытавшись подключиться к нему с другого компьютера под управлением Windows или Linux.

## Создание общих ресурсов печати в Windows

Если необходимо распечатать документы на принтере, физически подключенном к компьютеру под управлением ОС Windows, то необходимо создать на этом компьютере общий ресурс печати. Для этого выполните следующие действия:

1. Установите и настройте локальный принтер.
2. Откройте раздел Hardware and Sound в панели управления и щелкните **Devices and Printers**.
3. Щелкните правой кнопкой мыши на принтере, который вы хотите сделать общим, и выберите пункт **Printer Properties**.
4. В диалоговом окне **Properties** перейдите на вкладку **Sharing**. В результате откроется окно, изображенное на рисунке 9.

**Рисунок 9. Предоставление общего доступа к принтеру из диалогового окна Properties**



5. Установите флажок **Share this printer** и введите имя принтера в поле **Share name**.

Для некоторых Windows-клиентов могут иметь важное значение флажок **Render print jobs on client computers** и кнопка **Additional Drivers**; если же вы намерены использовать принтер для печати заданий из Linux-клиентов, то не нужно беспокоиться об этих параметрах.

6. Нажмите кнопку **OK**, чтобы сохранить изменения.

Подсистема печати CUPS в Linux умеет создавать очереди печати для общих принтеров SMB/CIFS, поэтому в Linux можно без проблем использовать общие принтеры Windows. Не забывайте о том, что если общий принтер не является PostScript-принтером, то для него необходимо использовать драйверы Linux, т. е. задания печати должны обрабатываться на стороне Linux, после чего передаваться по сети в родном для принтера формате (в отличие от общих принтеров Samba, для которых задания на печать можно отправлять либо таким же образом, либо в формате PostScript).

### **Поиск и устранение проблем с помощью Linux-клиента**

Для отладки сетевых проблем, связанных с SMB/CIFS, в Linux существует множество инструментов, использующих как серверы Samba, так и серверы Windows. Некоторые из этих инструментов были описаны в других статьях этой серии, поэтому я расскажу о них кратко. Эти инструменты и приемы включают в себя монтирование общих ресурсов, а также использование команд `smbclient` и `smbget`. Также я упомяну о входящей в цели экзамена 314.4 команде `rdesktop`, хотя, фактически, она не связана с Samba или SMB/CIFS.

### **Тестовое монтирование общего ресурса**

Самый очевидный способ проверить общий ресурс – это попытаться использовать его при помощи обычных инструментов. Для Linux-клиента это означает смонтировать общий ресурс утилитами файловой системы `mount` и `cifs`, о которых рассказывается в статье "[Изучаем Linux, 302 \(смешанные среды\): интеграция с протоколом CIFS](#)" (ранее в целях экзамена LPIC 314.4 рассматривались команды `smbmount` и `smbumount`, предназначенные для монтирования и размонтирования файловых систем SMB/CIFS, но они больше не поддерживаются в Linux, начиная с версии ядра 2.6.37).

Самый простой способ проверить доступность общего ресурса – это попытаться выполнить команду `mount //СЕРВЕР/РЕСУРС /mnt`, где СЕРВЕР – имя сервера, а РЕСУРС – имя общего ресурса. Однако в соответствии с материалами статьи, содержащей цели экзамена 314.1, может потребоваться указать дополнительные параметры, например, имя пользователя, пароль и т. д.

Если вам не удается смонтировать общий ресурс, то можно поискать подсказки в log-файлах Samba или кольцевом буфере ядра (доступ к которому можно получить через команду `dmesg`). Может оказаться, что необходимо изменить настройки монтирования, а, возможно, проблема может быть на стороне сервера; также может потребоваться изменить файловые разрешения или какие-то другие параметры.

### **Проверка с помощью smbclient**

Программа `smbclient`, о которой рассказывалось в статье "[Изучаем Linux, 302 \(смешанные среды\): интеграция с протоколом CIFS](#)", может оказаться полезной при тестировании базовой функциональности, поскольку она является основным инструментом, напрямую взаимодействующим с сервером без необходимости монтирования файловой системы.

### **Проверка с помощью smbget**

Программа `Smbget`, являющаяся аналогом программы `wget`, получает файлы через протокол HTTP. Для использования `Smbget` необходимо знать точный URI-адрес файла, который необходимо получить. В простейшем случае необходимо набрать имя команды и URI-адрес нужного файла:

```
$ smbget smb://WEMBLETH/REPORTS/financial-report.pdf
```

В этом примере мы получаем файл `financial-report.pdf`, хранящийся на общем ресурсе REPORTS на сервере WEMBLETH. Можно использовать различные опции команды `Smbget`,

например, **-U** для указания имени пользователя или **-R** для рекурсивного получения содержимого всего дерева каталогов. Дополнительную информацию об этой команде можно найти на ее man-страницах.

## Проверка с помощью rdesktop

Программа **rdesktop** – это Linux-реализация протокола RDP (Remote Desktop Protocol), являющегося протоколом удаленного доступа Windows и основанного на тех же принципах, что и инструменты VNC (Virtual Network Computing). Протокол RDP включается в диалоговом окне **System Properties** (см. [рисунок 2](#)). Для его активации перейдите на вкладку **Remote** и в области **Remote Desktop** выберите параметр, соответствующий требуемому уровню безопасности. По завершении всех настроек нажмите кнопку **OK**.

**Примечание.** Не все версии Windows поддерживают протокол RDP (в частности, его не поддерживают версии Home).

Для использования программы **rdesktop** нужно набрать ее имя и указать DNS-имя или IP-адрес сервера. Можно также указать различные опции, например, **-U** для указания имени пользователя. Дополнительную информацию об этой команде можно найти на ее man-страницах.

## Ресурсы

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Working with Windows clients](#) (EN).
- В статье "[Изучаем Linux, 302 \(смешанные среды\): Службы печати](#)" (Родерик Смит, developerWorks, август 2011 г.) рассказывается о настройке Samba в качестве общего сервера печати.
- В статье "[Изучаем Linux, 302 \(смешанные среды\): Управление доменом](#)" (Родерик Смит, developerWorks, август 2011 г.) рассказывается о настройке Samba в качестве контроллера домена.
- В статье "[Изучаем Linux, 302 \(смешанные среды\): интеграция с протоколом CIFS](#)" (Родерик Смит, developerWorks, октябрь 2011 г.) рассказывается о клиентских инструментах Linux для работы с SMB/CIFS.
- В статье "[Изучаем Linux, 302 \(смешанные среды\): NetBIOS и WINS](#)" (Родерик Смит, developerWorks, ноябрь 2011 г.) рассказывается о настройке просмотра сетевого окружения и функций NetBIOS в Samba.
- В [онлайновой документации команды net](#) (EN) вы найдете подробную информацию об этой команде и ее подкомандах.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.

# Изучаем Linux, 302 (смешанные среды): Управление доступом к файловой системе и общим ресурсам Linux

*Знакомство с файловыми разрешениями файловой системы в Linux*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Эта статья поможет вам подготовиться к сдаче экзамена LPI-302 и содержит информацию о взаимодействии Samba с файловой системой Linux и управлении правами доступа к файлам.

[Больше статей из этой серии](#)

**Дата:** 25.09.2012

**Уровень сложности:** средний

## Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели экзаменов сертификации LPIC-3*.

В этой статье рассматриваются следующие темы:

- Эффективное использование системы контроля доступа к файлам и директориям.
- Взаимодействие Samba с файловыми разрешениями операционной системы Linux.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 315.1 темы 315. Цель имеет вес 3.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды.

## Еще раз о файловых разрешениях в Linux

Взаимодействие Samba с файловой системой Linux базируется на концепции файловых разрешений этой операционной системы.

## Базовые операции

Управлять доступом к файлам в Linux очень просто. Каждый файл имеет двух владельцев – пользователя и группу. Права доступа к файлу определяются отдельно для пользователя, группы и всех остальных.

Права доступа управляют тремя базовыми операциями с файлами – чтение, запись и выполнение. Право на чтение (Read) означает возможность просматривать содержимое файла или директории. Право на запись (Write) позволяет владельцу удалять или изменять файл, а

также создавать новые файлы в директории. Право на выполнение (Execute) требуется для запуска двоичного файла или сценария командной оболочки, а также для входа в директорию.

Особый интерес при работе с файловыми разрешениями представляют сценарии командной оболочки. Для обычного двоичного файла можно установить разрешение только на выполнение, но не устанавливать разрешение на чтение, и пользователь сможет запустить программу, не видя, что находится внутри исполняемого файла. Сценарии командной оболочки запускаются по-другому: для запуска такого сценария пользователю необходимо иметь разрешение на его чтение. Имея разрешение на выполнение, пользователь может запустить сценарий, например, с помощью такой команды: `./myscript.sh`.

Разрешение для файла в Linux представлено в виде набора восьмеричных (с основанием 8) цифр и называется *режимом* файла. Каждая цифра содержит права на чтение/запись/выполнение для одной группы людей. Первая цифра применяется к пользователю-владельцу файла, вторая – к группе-владельцу и третья – ко всем остальным пользователям. Иногда разрешения могут состоять из четырех цифр. В этом случае первая цифра содержит специальные свойства файла, а последние три цифры – разрешения для пользователя/группы/остальных (как в первом случае).

Чтобы понять закодированные разрешения, необходимо обратиться к двоичной системе счисления. Восьмеричное число может быть представлено тремя двоичными битами:

- **001**. Бит исполнения (Execute)
- **010**. Бит записи (Write)
- **100**. Бит чтения (Read)

Складывая биты вместе, мы получим комбинацию разрешений. Для файла с правами на чтение и выполнение будут установлены два соответствующих бита, образующие двоичное число 101, что соответствует восьмеричному значению 5. Если установлены все три бита, то мы получим двоичное число 111 или 7 в восьмеричной системе. И наоборот, число 6 в восьмеричной системе соответствует числу 110 в двоичной системе, т. е. разрешениям на чтение и запись, но не на выполнение.

Режим файла 644 для всех трех групп пользователей означает следующие разрешения: чтение/запись для пользователя-владельца и только чтение для группы-владельца и всех остальных. Разрешение для группы имеет приоритет по сравнению с разрешением для всех остальных, поэтому режим файла 604 не позволит группе-владельцу просматривать файл, но позволит делать это всем остальным. Режим 640 позволит читать и изменять файл его владельцу и просто читать файл группе; любой доступ к файлу будет запрещен для всех остальных.

### Работа с файловыми разрешениями из командной строки

Команда `chmod` изменяет режим файла. Например, команда `chmod 700 foo` изменит разрешения для файла `foo` на 700 независимо от того, какие разрешения были установлены ранее.

Можно также сбросить разрешения из командной строки. Вместо восьмеричных значений можно указывать относительные значение в форме `[ugo]a [+ - =] [rwx]`. Сначала указывается одна из букв `U`, `G`, `O` или `A`, что означает *user* (пользователь), *group* (группа), *other* (другие) или *all* (все), соответственно. После этого можно добавить (+), удалить (-) или установить (=) биты Read/Write/Execute (чтение/запись/выполнение).

Например, команда `chmod u+x foo` устанавливает для файла `foo` бит Execute, не затрагивая все остальные биты. Команда `chmod g-rw something` удаляет разрешения на запись и чтение для группы.

Можно также использовать команду `chmod` с параметром `--reference`. В результате выполнения команды `chmod --reference файл1 файл2` файл 2 будет иметь такие же разрешения, что и файл 1.

Для изменения владельца файла используется команда `chown`. Например, команда `chown sean foo` изменяет владельца файла `foo` на `sean`. Владельца файла может изменить только пользователь `root` или текущий владелец.

Команда `chgrp` изменяет группу-владельца файла. Обычный пользователь должен входить в состав новой группы.

### Использование масок

Поскольку по своей природе файловые разрешения являются двоичными, то для установки или снятия битов можно воспользоваться операциями двоичной логики. В данном случае применяются бинарные операции **OR** (логическое "или") и **AND** (логическое "и"). Результаты выполнения этих операций показаны в таблицах на рисунке 1.

**Рисунок 1. Таблицы результирующих значений для операций OR и AND**

		0	1
0	0	1	
1	1	1	

Or

		0	1
0	0	0	
1	0	1	

And

Результат операции **OR** будет Истина (**1**) в том случае, если хотя бы один из операндов равен 1. Результат будет Ложь (**0**) лишь в том случае, когда оба операнда равны 0. Операция **AND** работает наоборот: для получения Истины оба бита должны быть равны 1, в противном случае результат будет ложным. Важно заметить, что порядок расположения операндов не имеет значения: операция **A OR B** тождественна операции **B OR A**.

Если операция выполняется над несколькими двоичными битами, то каждый бит вычисляется отдельно. Таким образом, результатом операции **01 AND 11** будет являться **01**. Первая результирующая цифра – это 0, поскольку результатом операции **0 AND 1** будет 0. Вторая результирующая цифра – это 1, поскольку результатом операции **1 AND 1** будет 1. Чтобы в дальнейшем вам было проще работать с файловыми разрешениями, запомните, что для установки битов будет использоваться операция **OR**, а для их снятия – операция **AND**.

Возвращаясь к восьмеричной системе, получается, что если операция **OR** со значением 600 применяется к файлу с любым режимом, то пользователь-владелец файла получает разрешения на чтение и запись независимо от его текущих разрешений. Операция **AND** со значением 775бросит бит записи для всех остальных пользователей, поскольку число 5 в двоичном формате имеет значение 101, а бит записи – значение 010.

### Взаимодействие Samba с файловыми разрешениями

При каждом подключению к серверу Samba запускается отдельный процесс, хозяином которого является подключившийся пользователь. Следовательно, для процесса Samba действуют те же самые файловые разрешения, что для пользователя, подключившегося к директории на сервере. Отсюда следует, что когда пользователь создает файл или директорию через Samba, он становится ее владельцем. Когда пользователь изменяет файловые разрешения через проводник Windows, то они преобразуются в режим файла, как если бы использовалась команда `chmod`.

В Samba есть ряд параметров, управляющих назначением разрешений в различных ситуациях. Что касается параметров для работы с файловыми разрешениями, то одни

параметры предназначены для установки битов, а другие – для их сбрасывания. Все эти параметры можно использовать на уровне отдельного общего ресурса или на глобальном уровне, затрагивающем все общие ресурсы. Как и в случае с другими глобальными параметрами, их можно переопределить на уровне того или иного общего ресурса.

## Создание файлов и директорий

Вновь создаваемые файлы должны иметь набор разрешений. Точно так же, директория, созданная в проводнике Windows с помощью команды **New Folder**, должна иметь определенный начальный режим. Эти две ситуации обрабатываются различными параметрами Samba.

Сначала Samba получает запрос на создание файла с определенным файловым режимом. Затем для сброса битов выполняется операция **AND** со значением параметра **create mask**. По умолчанию маска имеет значение 0744, в результате чего удаляются разрешения на запись и выполнение для группы и всех остальных. После этого полученный результат объединяется операцией **OR** с параметром **force create mode** для установки нужных битов. По умолчанию параметр **force create mode** имеет значение 000, в результате чего файловые разрешения не затрагиваются.

При создании директории выполняются те же самые действия за исключением того, что вместо параметров **create mask** и **force create mode** используются параметры **directory mask** и **force directory mode**, соответственно. В листинге 1 приведены примеры конфигураций, изменяющих процесс создания файлов и директорий.

### Листинг 1. Использование параметров, изменяющих разрешения для создаваемых файлов и директорий

```
[global]
create mask = 770
force create mode = 600
directory mask = 777
force directory mode = 711

[public]
create mask = 777
force create mode = 666
```

В листинге 1 параметры распределены по двум разделам. Раздел **[global]** содержит параметры как для файлов, так и для директорий. При создании файлов биты разрешений для других пользователей сбрасываются в 0, поскольку последняя цифра параметра **create mask** – это 0, а все остальные остаются без изменений, поскольку им соответствуют семерки. После этого получившийся результат объединяется операцией **OR** со значением 600, в результате чего пользователи получают разрешения на чтение и запись своих собственных файлов.

Режим директорий объединяется операцией **AND** со значением 777, в результате чего все биты передаются без изменений оператору **OR** и объединяются со значением 711, позволяющим владельцам файлов читать, изменять и выполнять, а всем остальным – по меньшей мере, выполнять их. Для файлового ресурса *public* установлены менее строгие ограничения. Любой пользователь получает права на чтение и запись файлов в этой директории.

## Изменение разрешений для файлов и директорий

С помощью проводника Windows можно указывать пользователей, имеющих доступ к вашим

файлам. По умолчанию выводятся разрешения Linux, соответствующие группам Microsoft® Windows NT®. Если вы измените эти разрешения, то они будут заново сопоставлены с файловыми разрешениями Linux. Существует другой набор параметров, управляющих установкой и сбросом битов при изменении файловых разрешений, а не при создании файлов и директорий.

Для работы с файловыми разрешениями при их изменении используются следующие параметры:

- **security mask**, объединяемый операцией AND с файловыми разрешениями
- **force security mode**, объединяемый операцией OR с файловыми разрешениями
- **directory security mask**, объединяемый операцией AND с разрешениями для директории
- **force directory security mode**, объединяемый операцией OR с разрешениями для директории

#### **Сводная таблица параметров для работы с правами доступа**

В таблице 1 перечислены все параметры, относящиеся к установке или сбросу битов файловых разрешений. Помните о том, что сначала биты сбрасываются в соответствии с заданной маской, а уже затем устанавливаются.

**Таблица 1. Параметры Samba для работы с правами доступа к файлам и директориям**

<b>Действие</b>	<b>Установка битов (OR)</b>	<b>Сброс битов (AND)</b>
Создание файла	<b>force create mode</b>	<b>create mask</b>
Создание директории	<b>force directory mode</b>	<b>directory mask</b>
Изменение клиентом разрешений для файла	<b>force security mode</b>	<b>security mask</b>
Изменение клиентом разрешений для директории	<b>force directory security mode</b>	<b>directory security mask</b>
Принудительная установка владельцев и групп файлов		

Все параметры, о которых говорилось ранее, изменяют режимы файлов и, следовательно, разрешения на их чтение, запись и выполнение для их владельцев, групп и всех остальных пользователей. По умолчанию владельцами файлов становятся создавший их пользователь и его группа. Тем не менее, иногда необходимо назначить для файлов других владельца или группу (например, это может быть группа, созданная для определенного проекта, или какой-то типичный пользователь), что может оказаться полезным при создании файловых ресурсов, которые используются группами пользователей.

Для установки требуемого владельца и группы файла в Samba есть два параметра: **force user** и **force group**. Эти параметры можно использовать на глобальном уровне, но на практике они используются на уровне общего ресурса. Например, если задать на уровне файлового ресурса параметр **force group = projecta**, то все файлы будут принадлежать группе projecta.

Параметр **force group** можно использовать совместно со знаком плюс (+), после которого указывается имя группы, например, **force group = +admins**. Знак плюс не воспринимается здесь дословно, а говорит Samba о том, что необходимо изменить группу на **admins** только в том случае, если пользователь уже состоит в ней (например, это его дополнительная группа). Пользователи, которые не состоят в группе **admins**, будут продолжать использовать свою основную группу при создании новых файлов.

#### **Заключение**

В Samba имеется несколько различных параметров, влияющих на расчет файловых

разрешений. Эти параметры имеют вид восьмеричной маски, которая объединяется логической операцией **AND** с предлагаемыми разрешениями для сброса ненужных битов, а затем объединяется операцией **OR** для установки требуемых битов. Биты устанавливаются отдельно для файлов и директорий, а также для вновь создаваемых файлов и для изменяемых разрешений; таким образом, всего существует восемь различных параметров. Наконец, для файлов указанного общего ресурса можно принудительно назначить пользователя и группу.

Несмотря на то, что Samba позволяет гибко управлять файловыми разрешениями, следует аккуратно использовать эти возможности, поскольку можно получить совершенно не тот результат, который вы ожидали.

## Ресурсы

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Linux file system and share/service permissions \(EN\)](#).
- Man-страница [smb.conf](#) (EN) содержит дополнительные примеры и описание команд, встречающихся в этой статье.
- Параметр [umask](#) (EN) работает почти так же, как и параметр **force security mask**. Предложенная ссылка поможет вам разобраться с двоичной математикой.
- Руководство [File, Directory, and Share Access Controls](#) (EN) содержит дополнительную информацию о том, как Samba работает с системой контроля доступа при выполнении клиентом различных действий.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302 \(EN\)](#), а также [примеры заданий и вопросов \(EN\)](#).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.

# Изучаем Linux, 302 (смешанные среды): Безопасность Samba

*Защита Samba на уровне брандмауэра и на уровне демона*

Шон Уолберг, старший сетевой инженер, P.Eng

**Описание:** Эта статья поможет вам подготовиться к сдаче экзамена LPI-302. Она посвящена системе безопасности Samba и устранению проблем, связанных с безопасностью.

**Дата:** 23.10.2012

**Уровень сложности:** сложный

## Об этой серии

Эта серия статей поможет вам освоить администрирование операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Настройка доступа к серверу Samba на уровне брандмауэра.
- Поиск и устранение проблем, связанных с безопасностью Samba.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 315.2 темы 315. Цель имеет вес 2.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимы продвинутые знания о Linux и работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. Кроме того, вам потребуется доступ к Windows-окружению для проверки параметров безопасности, о которых будет идти речь в этой статье.

## Брандмауэры

В Samba имеется множество возможностей для ограничения доступа к общим ресурсам – это разрешение доступа только для определенных пользователей, установка паролей, проверка членства в группах и фильтрация на сетевом уровне. Параметры последней группы, например, `allow hosts` и `smb ports` работают на уровне IP-адресов и портов UDP/TCP, позволяя указывать, каким компьютерам разрешено подключаться к серверу Samba.

Контроль на сетевом уровне осуществляется тогда, когда можно определить, какие устройства будут подключаться к серверу – например, устройства локальной сети или подсети или группы серверов. Это первая линия обороны: если злоумышленник не может подключиться к устройству, значит, оно уже как-то защищено.

Управление сетевым доступом через демон Samba может показаться идеальным решением, однако это не самый лучший способ. Чтобы определить, является ли удаленное подключение доверенным, Samba должна сначала разрешить его, поскольку невозможно получить никакой информации о подключении, которое еще не установлено. Если защита строится на том, чтобы запретить определенному кругу лиц подключаться к Samba, то было бы благоразумнее сделать так, чтобы Samba вообще не видела этих подключений. Любые настройки внутри Samba касаются только Samba, и при таком способе защиты необходимо искать решения для защиты других демонов, например, для Web- и файловых серверов.

Как правило, в большинстве случаев за безопасность сети отвечает не системный администратор, а другие сотрудники ИТ-подразделений. Управление доступом на уровне хоста, а не на уровне приложения позволяет разделять проблемы и снижает количество ошибок, вызванных внесением изменений в файл `smb.conf`.

## Знакомство с `iptables`

В Linux имеется надежный брандмауэр уровня хоста под названием `iptables`. Этот брандмауэр может проверять входящие, исходящие или промежуточные пакеты, передаваемые через Linux-устройство. Название `iptables` может означать встроенную в ядро Linux систему фильтрации пакетов либо имя команды, используемой для управления сетевыми фильтрами. Система фильтрации пакетов ядра развивалась на протяжении многих лет, превратившись из простого механизма сравнения пакетов в надежный брандмауэр с поддержкой динамической загрузки подключаемых модулей. Таким образом, настройка `iptables` может оказаться достаточно сложной задачей, если не ограничиваться использованием типовых конфигураций.

Первая важная идея iptables заключается в идее самих таблиц. *Таблица* – это независимый список правил и действий. Если ядро Linux должно отфильтровать пакет, то оно обращается к таблице *filter*. Если выполняется трансляция сетевых адресов (NAT), то используется таблица *nat*. В зависимости от того, какие сетевые функции были загружены в ядро, могут использоваться различные таблицы. Пакеты могут передаваться между несколькими таблицами, например, при выполнении фильтрации пакета до выполнения трансляции сетевого адреса.

Внутри каждой таблицы содержится набор *цепочек*. В каждой таблице есть несколько предопределенных цепочек, к которым можно добавлять собственные цепочки. Предопределенные цепочки используются на различных этапах жизненного цикла пакетов. Например, в таблице *filter* есть три предопределенные цепочки:

- **INPUT.** Используется для определения того, что делать с пакетами, предназначенными для самого хоста.
- **OUTPUT.** Применяется к пакетам, созданным хостом.
- **FORWARD.** Работает только с пакетами, передающимися с одного сетевого интерфейса на другой, например, когда хост выступает в качестве маршрутизатора.

Цепочка содержит упорядоченный список правил (он может быть пустым), каждое из которых состоит из условия и выполняемого действия. Условие может быть практически любым, начиная от IP-адреса или порта и заканчивая операторами ограничения скорости, срабатывающими только при слишком частом повторении определенного события.

Выполняемым действием может быть другая цепочка или операция, например, инструкция на принятие илиброс пакета. Как условия, так и выполняемые действия можно создавать с помощью модулей ядра, поэтому возможности безграничны.

Ядро выбирает цепочку на основе того, что должно быть сделано, и просматривает каждое правило по порядку. При первом совпадении ядро переходит к выполнению действия. В большинстве случаев обработка правил прекращается, хотя некоторые действия (например, журналирование) считаются *незавершающими*, и ядро переходит к обработке следующего правила. Если совпадений не находится, то выполняется действие цепочки по умолчанию.

**Примечание.** Для выполнения действий, описанных в этой статье, используется только таблица *filter*.

### Защита Samba с помощью брандмауэра

Существует множество различных способов настройки политики брандмауэра для Samba, реализуемых в зависимости от топологии сети и от того, кто должен иметь доступ к серверу Samba. С самого начала нужно решить, будет обеспечен защитой весь хост целиком или же только Samba.

Если вы решили защищать весь хост, то не имеет значения, какие порты использует Samba. В следующем примере показана простая политика, позволяющая локальному серверу принимать трафик только от клиентов частной сети 10.0.0.0/8.

```
iptables -A INPUT -s 10.0.0.0/8 -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -P INPUT DROP
```

Первая команда добавляет правило в цепочку INPUT путем его добавления в текущий список правил. Это правило говорит о том, что для любого трафика, приходящего из сети-источника (-s) 10.0.0.0/8, будет выполняться действие ACCEPT, разрешающее получение пакетов.

Вторая команда разрешает получение пакетов, приходящих при уже установленном соединении – за это отвечает вызов matcher state (оператор проверки соответствия состояния

**-m state**). Этот matcher state следит за тем, какие подключения покидают хост. Ответные исходящие пакеты считаются *установленными* или *связанными* (**established** или **related**), поэтому остальные правила пропускают эти пакеты.

Последняя команда определяют политику по умолчанию цепочки INPUT, сбрасывающую пакеты. Если пакет не пришел из сети 10.0.0.0/8 или не является пакетом установленного хостом соединения, то он сбрасывается.

Можно настроить более тонкую фильтрацию на уровне портов. В предыдущем примере пакеты фильтруются только на основе адреса сети-источника, поэтому несоответствующий трафик блокируется для всех служб. Если же, например, на вашем хосте запущен Web-сервер, который должен быть доступен всем пользователям Интернета, то в этом случае предыдущая политика уже не подходит.

Вспомним статью "[Изучаем Linux, 302 \(смешанные среды\): конфигурация Samba](#)", в которой говорилось о том, что Samba использует четыре различных порта:

- **137 UDP.** Службы имен NetBIOS.
- **138 UDP.** Службы датаграмм NetBIOS.
- **139 TCP.** Службы сеансов NetBIOS.
- **445 TCP.** Прямая передача (CIFS over TCP).

В листинге 1 показана политика, разрешающая подключения к Samba только из сети 10.0.0.0/8 и подключения к Web-серверу без каких-либо ограничений.

### Листинг 1. Политика, работающая на уровне портов

```
iptables -A INPUT -p tcp -m state --state NEW --dport 80 -j ACCEPT
iptables -A INPUT -p tcp -m state --state NEW --dport 443 -j ACCEPT
iptables -A INPUT -p udp -s 10.0.0.0/8 --dport 137 -j ACCEPT
iptables -A INPUT -p udp -s 10.0.0.0/8 --dport 138 -j ACCEPT
iptables -A INPUT -p tcp -m state --state NEW -s 10.0.0.0/8 --dport 139 -j ACCEPT
iptables -A INPUT -p tcp -m state --state NEW -s 10.0.0.0/8 --dport 445 -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -P INPUT DROP
```

Политика в листинге 1 сложнее предыдущей политики, поскольку в ней определяются несколько различных приложений, для каждого из которых применяются свои правила. Первые два правила соответствуют любым входящим TCP-пакетам (-p **tcp**), являющимся частью нового подключения (-m **state --state NEW**) и отправляемым на порты 80 или 443 (-dport **80**, -dport **443**). Никаких ограничений на адрес источника нет, поэтому принимаются любые пакеты.

Следующие две строки соответствуют UDP-пакетам (-p **udp**), приходящим из внутренней сети (-s **10.0.0.0/8**) на порты 137 и 138 (-dport **137**, -dport **138**). UDP является протоколом без запоминания состояния (stateless), поэтому нет необходимости выяснять, новое это подключение или уже установленное.

Строки 5 и 6 объединяют команду сопоставления состояния с фильтром адресов источников, разрешая только новые подключения к портам 139 и 445, если пакеты приходят из внутренней сети.

Наконец, последние две строки работают так же, как и в предыдущей политики. Если пакет относится к уже установленному соединению, то он принимается. Весь остальной трафик отбрасывается.

## Поиск и устранение проблем, связанных с брандмауэром

Проблемы, связанные с брандмауэром, возникают достаточно часто, поскольку часто обнаруживаются непредвиденные требования или оказывается, что приложение работает не так, как ожидалось. Еще один частый источник проблем - это сами правила, особенно при работе с длинными списками номеров портов и IP-адресов. Тем не менее, не все проблемы связаны с брандмауэрами, поэтому нужно уметь выполнять основные операции диагностики и устранения проблем в сети.

### Просмотр политики

Для просмотра политики используется команда `unexpected`. Опция `-L` выводит содержимое политики, а опция `-V` позволяет выводить дополнительную информацию, например, счетчики пакетов. В листинге 2 показано содержимое политики из листинга 1.

### Листинг 2. Подробный просмотр содержимого политики

```
# iptables -L -v
Chain INPUT (policy DROP 47 packets, 5125 bytes)
pkts bytes target  prot opt in     out    source        destination
  0     0 ACCEPT   tcp   --  any    any    anywhere    anywhere      state NEW tcp dpt:http
  0     0 ACCEPT   tcp   --  any    any    anywhere    anywhere      state NEW tcp dpt:https
  0     0 ACCEPT   udp   --  any    any    10.0.0.0/8  anywhere      udp dpt:netbios-ns
  0     0 ACCEPT   udp   --  any    any    10.0.0.0/8  anywhere      udp dpt:netbios-dgm
  0     0 ACCEPT   tcp   --  any    any    10.0.0.0/8  anywhere      state NEW tcp dpt:139
  0     0 ACCEPT   tcp   --  any    any    10.0.0.0/8  anywhere      state NEW tcp dpt:445
 214 15216 ACCEPT   all   --  any    any    anywhere    anywhere      state RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in     out    source          destination

Chain OUTPUT (policy ACCEPT 292 packets, 35009 bytes)
pkts bytes target      prot opt in     out    source          destination
```

В первом и втором столбцах подробного вывода отображается число пакетов и байтов, попавших под действия правил. Из листинга 2 видно, что пакеты попадали под действие только последнего правила. Если внимательно посмотреть на первую строку вывода, то окажется, что выполняемое по умолчанию действие также содержит отличный от нуля счетчик пакетов. 47 пакетов было сброшено, поскольку они не удовлетворяли правилу; это означает, что либо кто-то пытался получить незаконный доступ к хосту, либо разрешенный трафик был заблокирован неверно настроенной политикой брандмауэра.

Просмотр политики брандмауэра нужен также для того, чтобы полностью выяснить, как работает правило. Поскольку обработка трафика останавливается после первого совпадения, следует начать с самой верхней строки политики и постепенно переходить к следующим правилам, чтобы определить, какое из них блокирует ваш трафик.

Одна из типичных ошибок – это ситуация, когда более общее правило предшествует более конкретному правилу. Во избежание проблем самые детализированные правила следует помещать в самое начало политики, чтобы исключения обрабатывались в первую очередь. Тем не менее, этот подход не всегда решает проблему, и вы можете обнаружить, что не все пользователи могут подключаться к вашему серверу.

В листинге 3 показана политика сервера, расположенного в сети Engineering. Пользователи этой сети не могут подключаться к службе.

### Листинг 3. Политика, содержащая перекрывающиеся правила

```
# iptables -L -v
Chain INPUT (policy DROP 21 packets, 2967 bytes)
target  prot opt in  out  source      destination
ACCEPT  tcp  --  any   any   anywhere   anywhere    state NEW tcp dpt:http
ACCEPT  tcp  --  any   any   anywhere   anywhere    state NEW tcp dpt:https
DROP    tcp  --  any   any   10.0.0.0/8 anywhere
ACCEPT  udp  --  any   any   10.2.3.0/24 anywhere   udp dpt:netbios-ns
ACCEPT  udp  --  any   any   10.2.3.0/24 anywhere   udp dpt:netbios-dgm
ACCEPT  tcp  --  any   any   10.2.3.0/24 anywhere   state NEW tcp dpt:netbios-ssn
ACCEPT  tcp  --  any   any   10.2.3.0/24 anywhere   state NEW tcp dpt:microsoft-ds
ACCEPT  all  --  any   any   anywhere   anywhere    state RELATED,ESTABLISHED
```

В этом примере сервер находится в сети Engineering с адресом 10.2.3.0/24. Доступ для остальных пользователей компании, находящихся в сети 10.0.0.0/8, должен быть заблокирован. Поскольку сеть 10.2.3.0/24 является частью сети 10.0.0.0/8, а правило, полностью блокирующее всю сеть 10.0.0.0/8 указано перед правилами, относящимися к SMB, то для пользователей сети Engineering будет срабатывать правило **DROP**, поскольку в iptables используется принцип первого совпадения, а не наилучшего.

Решением этой проблемы является перемещение правила блокирования корпоративной сети вниз чтобы оно стояло после более конкретных правил. Тогда прием пакетов сети Engineering сервером будет разрешен.

### Продвинутые инструменты диагностики и устранения проблем

Часто нельзя точно определить, является ли брандмауэр источником проблем сетевого соединения. Самый простой способ проверить это – отключить брандмауэр и посмотреть, установится ли соединение, или нет. Однако это не всегда возможно, поэтому если вы не можете отключить брандмауэр, то проверьте, приходят ли пакеты на сервер.

Сетевые пакеты, которые видят сервер (даже если брандмауэр блокирует их), может показать утилита **tcpdump**. Если утилита показывает попытки установления соединения сервером, значит, пакеты доходят до него. Это означает, что пакеты блокируются брандмауэром (при условии, что служба запущена). В листинге 4 показана работа утилиты **tcpdump**.

### Листинг 4. Трассировка пакетов заблокированного SMB-подключения

```
# tcpdump -i eth0 tcp port 445
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
20:24:18.392106 IP CLIENT.search > SERVER.microsoft-ds: S ...
20:24:21.358458 IP CLIENT.search > SERVER.microsoft-ds: S ...
20:24:27.393604 IP CLIENT.search > SERVER.microsoft-ds: S ...
```

Утилита **tcpdump** была запущена со следующими опциями:

- **-i eth0**. Прослушивать сетевой интерфейс eth0.
- **tcp port 445**. Отслеживать пакеты на TCP-порту 445.

В листинге 4 мы видим три пакета, пришедшие на сервер. Стрелка означает направление потока: эти три пакета пришли с клиента на порт сервера microsoft-ds (порт с номером 445). Символ **S** в конце строки означает попытку подключения, а отсутствие ответа означает, что сервер не отвечает.

Другим признаком ошибки соединения является увеличивающийся временной интервал между успешно принятыми пакетами. Метка времени слева показывает нам, что второй пакет был получен примерно через 3 секунды после получения первого пакета, а третий пакет – через 6 секунд после получения предыдущего. В большинстве сетевых протоколов реализована экспоненциальная задержка передачи, т. е. время между успешными попытками каждый раз удваивается.

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Samba security \(EN\)](#).
- Man-страница [smb.conf](#) (EN) содержит дополнительные примеры и информацию о командах из этой статьи.
- Правила брандмауэра действуют до следующей перезагрузки компьютера. Узнайте, как сохранять наборы правил в дистрибутивах [Debian](#) (EN) и [Red Hat](#) (EN).
- Не забывайте следить за [обновлениями безопасности Samba](#) (EN) и поддерживать вашу систему в актуальном состоянии.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.

### Получить продукты и технологии

- Графический менеджер политик брандмауэра [Firewall Builder](#) (EN) позволяет легко управлять правилами iptables.
- Посетите домашнюю страницу iptables, Web-сайт [Netfilter.org](#) (EN), где вы найдете дополнительную документацию и список модулей, расширяющих функциональность этого брандмауэра.

## Изучаем Linux, 302 (смешанные среды): Настройка производительности

*Измерение и повышение производительности Samba*

[Шон Уолберг](#), старший сетевой инженер, P.Eng

**Описание:** Эта статья поможет вам подготовиться к сдаче экзамена LPI-302. Она объясняет, как измерить и повысить производительность Samba.

**Дата:** 25.10.2012

**Уровень сложности:** средний

# Об этой серии

Эта серия статей поможет вам освоить задачи администрирования операционной системы Linux. Вы можете использовать материалы этих статей для подготовки к [экзаменам программы LPIC третьего уровня \(LPIC-3\)](#).

Чтобы посмотреть описания статей этой серии и получить ссылки на них, обратитесь к нашему [перечню материалов для подготовки к экзаменам LPIC-3](#). Этот перечень постоянно дополняется новыми статьями по мере их готовности и содержит *текущие (по состоянию на ноябрь 2010 года) цели* экзаменов сертификации LPIC-3.

В этой статье рассматриваются следующие темы:

- Измерение производительности Samba.
- Оптимизация оперативной памяти, используемой Samba.
- Повышение скорости передачи файлов в среде Server Message Block/Common Internet File System.

Эта статья поможет вам подготовиться к сдаче экзамена LPI 302 (специализация "Смешанные среды") и содержит материалы цели 315.3 темы 315. Цель имеет вес 1.

## Предварительные требования

Чтобы извлечь наибольшую пользу из наших статей, необходимо обладать продвинутыми знаниями о Linux и иметь работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды. Также необходимо хорошо понимать принципы работы сетей TCP/IP.

### Измерение производительности Samba

Прежде чем что-то улучшать, необходимо точно измерить это "что-то". Сначала мы измеряем что-либо, затем вносим изменения, измеряем снова и сравниваем результаты. В случае с Samba необходимо измерить следующие параметры:

- Время отклика и пропускную способность клиента при отсутствии нагрузки на сервер.
- Время отклика и пропускную способность клиента при определенной загрузке сервера.
- Характеристики сервера при его определенной загрузке.
- Предельно допустимую нагрузку сервера (количество клиентов или пропускная способность).

Измерение времени отклика позволяет понять, что может получить клиент в тестовой среде, поэтому для тестов нужно создать условия, приближенные к реальности. Например, производительность отличается при обращении к одному и тому же файлу, и копировании директории, содержащей файлы различного объема – это не одно и то же.

Измерение характеристик сервера при его определенной загрузке позволяет понять, сколько ресурсов имеется в запасе. Если при определенной загрузке сервер перестает справляться с выполнением своих задач, значит, его ресурсов уже не хватит для нормальной работы при увеличении нагрузки. Этот же подход используется для измерения загрузки рабочего сервера и экстраполяции его запаса мощностей.

Наконец, с помощью так называемых "стресс-тестов" (полная загрузка сервера для определения его предельных возможностей) также можно получить интересную информацию, хотя и не настолько полезную, как информация, которую выдают другие тесты. Если вы хотите выяснить, сможет ли ваш сервер справиться с определенной нагрузкой, то этот стресс-тестирование поможет вам получить такую информацию. Зачастую такие тесты более полезны при измерении низкоуровневых характеристик, например, максимальной

производительности дисковой подсистемы ввода/вывода сервера.

Samba – это сетевой сервер, поэтому важно как можно точнее создать модель используемого сетевого окружения. Если время обращения ваших клиентов к серверу составляет 50 миллисекунд, то для них эффект сетевой задержки будет более ярко выражен по сравнению с эффектом для локальных клиентов. Эта информация позволяет вам соответствующим образом настраивать приоритеты.

## Разработка теста

Можно купить довольно дорогое устройство, которое может имитировать клиентский трафик и выполнять точные измерения производительности. Если вы хотите выполнить эталонные тесты (benchmarks) или разрабатываете серверное аппаратное обеспечение, то такое устройство может оказаться хорошим решением. Однако если вам просто необходимо более оптимально настроить ваш сервер, а времени и средств как всегда в обрез, то, не стоит тратить деньги на дорогое устройство, которое еще надо будет освоить.

В качестве примера рассмотрим первый тест, оценивающий производительность случайного чтения путем загрузки большого числа файлов с помощью клиента, использующего командную строку Samba. Нас будет интересовать главный вопрос: как быстро эта директория может быть загружена.

Как и любая "правильная" утилита UNIX®, утилита **smbclient** может считывать список инструкций с устройства стандартного ввода. В следующем фрагменте кода показано несколько команд для загрузки содержимого директории:

```
prompt
recurse
mget smbtest
```

Команда **prompt** подавляет запросы на подтверждение загрузки, а команда **recurse** говорит о необходимости просмотра всех вложенных директорий при загрузке нескольких файлов. Наконец, команда **mget smbtest** инструктирует клиента начать загрузку директории **smbtest**. Поместите в эту директорию тестовые файлы объемом в несколько сот мегабайтов, и тест для проверки производительности готов.

Для запуска теста подключитесь к общему ресурсу с помощью **smbclient** и перенаправьте стандартный ввод в файл. В листинге 1 показано, как сделать это.

## Листинг 1. Запуск теста

```
$ time smbclient '\\192.168.1.1\test' password < instructions
Domain=[BOB] OS=[Unix] Server=[Samba 3.5.8-75.fc13]
getting file \smbtest\file2 of size 524288000 as file2 (5323.2 kb/s) (average 5323.2 kb/s)
getting file \smbtest\file1 of size 139460608 as file1 (5275.3 kb/s) (average 5313.0 kb/s)

real    2m2.289s
user    0m0.509s
sys     0m4.580s
```

Листинг 1 начинается с команды **time**, которая замеряет время выполнения команды, указанной в последующих аргументах. Основной командой является команда **smbclient**, а ее аргументами – имя общего ресурса и пароль пользователя. Можно также добавлять другие стандартные аргументы, например, **-U** для передачи имени пользователя. Наконец, конструкция **< instructions** перенаправляет стандартный ввод команды **Smbclient** в

файл под названием *instructions*, содержащий инструкции из первого фрагмента кода. В результате мы получаем процесс пакетного копирования нескольких файлов с измеряемым временем выполнения.

В результате выполнения этой команды был получен список переданных файлов, включающий среднее время передачи каждого из них. Вывод команды `time` помещен в конце этого списка и показывает, что копирование файлов объемом 664 МБ заняло 2 минуты и 2.289 секунды. Это мы и принимаем за эталонный результат теста. Если после каких-либо изменений время выполнения теста будет превышать 2 минуты и 2 секунды, то это значит, что эти изменения ухудшили производительность.

Если необходимо протестировать параметры Samba независимо от производительности локальных дисков и кэширования, то можно запустить тест несколько раз и взять за эталон результаты его последнего выполнения. Это даст гарантию того, что операционная система будет по максимуму кэшировать данные и минимизировать обращения к жесткому диску. В процессе тестирования изменений убедитесь в том, что количество свободной оперативной памяти на сервере не изменилось, в противном случае разница в объеме кэшированных данных может повлиять на результаты вашего эксперимента.

### Просмотр статуса Samba

Имея информацию о центральном процессоре, оперативной памяти, жестких дисках и сетевых интерфейсах, можно оценить состояние самого сервера, но нельзя понять, как работают сами приложения. Для просмотра текущих подключений и файловой активности Samba используется утилита `smbstatus`, которая также оказывается полезной для настройки производительности и диагностики неисправностей.

В листинге 2 приведен пример использования команды `smbstatus`.

### Листинг 2. Команда smbstatus

```
$ smbstatus
lp_load_ex: refreshing parameters
Initialising global parameters
params.c:pm_process() - Processing configuration file "/etc/samba/smb.conf"
Processing section "[global]"
Processing section "[homes]"
Processing section "[printers]"
Processing section "[extdrive]"

Samba version 3.5.8-75.fc13
PID      Username      Group          Machine
-----
17456    fred          fred          macbookpro-d0cd (:ffff:192.168.1.167)

Service      pid      machine      Connected at
-----
fred        17456    macbookpro-d0cd  Mon Jul 18 07:36:46 2011
extdrive    17456    macbookpro-d0cd  Mon Jul 18 07:36:46 2011

Locked files:
Pid  Uid  DenyMode  Access      R/W       Oblock     SharePath   Name      Time
-----
17456  505  DENY_NONE  0x100081  RDONLY    NONE       /home/fred   .
17456  505  DENY_NONE  0x100081  RDONLY    NONE       /home/fred   Documents
```

В листинге 2 показана текущее состояние сервера Samba. Из вывода команды видно, что в

текущий момент к Samba подключен один пользователь с именем *fred*, который смонтировал два общих ресурса (*fred* и *extdrive*). Также имеются две заблокированных директории.

## Настройка сети

Samba – это демон, основной задачей которого является отправка и получение пакетов по сети. Этим процессом, а также взаимодействием Samba с операционной системой управляют несколько параметров, которые могут значительно влиять на производительность.

## Основные параметры

Многие демоны обеспечивают одинаковый уровень обслуживания для всех клиентов без исключения, не различая их. Если служба перегружена, то каждый клиент получает одинаково плохой сервис. Эту ситуацию можно сравнить с телефонной сетью, в которой существует противоположная картина: если в сети возникает перегрузка, то люди не могут совершать новые звонки, однако уже существующие звонки продолжаются, как будто бы ничего не произошло. Если вы будете искусственно ограничивать значения определенных параметров Samba, то сможете предотвратить нехватку ресурсов.

Параметр **max connections** ограничивает количество подключений к определенному серверу Samba. Каждое подключение потребляет память и ресурсы центрального процессора, поэтому при слишком большом количестве подключений сервер может оказаться перегруженным. По умолчанию число подключений не ограничено. Если вы хотите ограничить число подключений к загруженному серверу, то можете задать жесткий лимит с помощью параметра **max connections**.

Параметр **max smbd processes** задает максимальное количество процессов, которые могут быть запущены. Этот параметр похож на параметр **max connections**, но управляет количеством процессов, создаваемых в результате подключений.

Параметр **log level** определяет количество информации, записываемой в системные журналы, и чем подробнее эта информация, тем больше ресурсов сервер тратит для ее записи на диск. Значения 1 и 2 этого параметра снижают объем записываемых на диск log-файлов и позволяют выделять больше ресурсов для обслуживания клиентов.

## Установка параметров сокетов

Когда приложение посылает операционной системе запрос на открытие сетевого подключения, оно также может запрашивать определенный способ обработки пакетов с помощью *параметров сокетов*. Параметры сокетов могут включать или отключать определенные настройки, влияющие на производительность, устанавливать для пакетов определенные биты качества обслуживания или же задавать опции сокетов на уровне ядра.

Команда **socket options** может управлять битами *type of service* (TOS), устанавливаемыми для пакетов. Биты TOS говорят маршрутизаторам о том, как следует обрабатывать трафик. Если маршрутизаторы настроены на распознавание этих битов, то трафик может обрабатываться в соответствии с требованиями приложений. Ключевое слово **IPTOS\_LOWDELAY** лучше всего подходит для сетей с низкими задержками (например, для локальных сетей), тогда как параметр **IPTOS\_THROUGHPUT** – наоборот, для глобальных сетей с большими задержками. Ваша сеть может быть настроена по-другому, поэтому возможны ситуации, когда использование этих опций может иметь обратный эффект.

Параметр **TCP\_NODELAY** отключает алгоритм Нагла (см. раздел Ресурсы), который больше подходит для диалоговых протоколов с большим количеством пересылаемых пакетов.

Если в вашей сети есть брандмауэры или другие устройства, сохраняющие состояние, то, возможно, вас заинтересует параметр **SO\_KEEPALIVE**, разрешающий использование элементов поддержки установленных TCP-соединений (TCP keepalives). Эти периодически рассылаемые пакеты поддерживают соединения открытыми и сохраняют состояние внутри

брандмауэров. В противном случае брандмауэр будет сбрасывать пакеты, и клиентам понадобится какое-то время на то, чтобы определить необходимость повторного подключения к серверу.

### Дополнительные факторы, влияющие на производительность

Настройка различных параметров в поисках оптимальных конфигураций может оказаться интересным занятием, но на производительность могут оказывать влияние некоторые вещи, совсем не связанные с конфигурацией Samba. Время операций увеличивается каждый раз, когда сервер выполняет какие-то действия помимо чтения данных с диска и их отправки клиентам или получения входящего трафика и его записи на диск.

### Ошибки Ethernet

Если пакет теряется во время передачи, то ядро должно заметить это и запросить повторную передачу пакета. Этот процесс может замедлить быстрый обмен данными, особенно если в сети между двумя устройствами существует большая задержка. Типичный источник потери пакетов - это несоответствие настроек коммутатора и сервера: либо протокол автосогласования не может предоставить правильные параметры, либо на одной стороне параметры заданы явно, а на другой включен режим автосогласования. Это приводит к ошибкам на обеих сторонах. Такие ошибки можно обнаружить при помощи команды **netstat**:

```
# netstat -d -i 2
Kernel Interface table
Iface      MTU Met     RX-OK RX-ERR RX-DRP RX-OVR     TX-OK TX-ERR TX-DRP TX-OVR Flg
eth1      1500  0 5258404     0     0     0 3024340     0     0     0     0 BMRU
eth1      1500  0 5258409     0     0     0 3024341     0     0     0     0 BMRU
eth1      1500  0 5258411     0     0     0 3024342     0     0     0     0 BMRU
```

Этот код показывает, что многоцелевая команда **netstat**, запущенная с параметрами **-d -i 2**, отслеживает ошибки на интерфейсе каждые две секунды. Каждые две секунды выводится статус всех доступных интерфейсов. В приведенном выше примере столбцы **RX-OK** и **TX-OK** содержат количество получаемых и отправляемых пакетов. Мы видим, что ошибки, сбросы и переполнения отсутствуют для обоих направлений трафика, т. е. пакеты не теряются.

Если вы обнаружите ошибку, то выясните с помощью команд **mii-tool** или **mii-diag**, какие скорость и режим дуплекса используются. В листинге 3 показано, как проверить сетевые параметры.

### Листинг 3. Проверка сетевых параметров

```
# mii-tool
eth1: negotiated 100baseTx-FD, link ok
# mii-diag eth1
Basic registers of MII PHY #24: 3000 782d 0040 6177 05e1 41e1 0003 0000.
The autonegotiated capability is 01e0.
The autonegotiated media type is 100baseTx-FD.
Basic mode control register 0x3000: Auto-negotiation enabled.
You have link beat, and everything is working OK.
Your link partner advertised 41e1: 100baseTx-FD 100baseTx 10baseT-FD 10baseT.
End of basic transceiver information.
```

Листинг 3 начинается с команды **mii-tool**, которая выводит общую информацию об

активных интерфейсах. Мы видим, что сетевой интерфейс работает на скорости 100 Мбит/с в режиме полного дуплекса и получил эти значения в процессе автосогласования. Команда `mii-diag` выводит более подробную информацию, которую уже можно отправить в службу поддержки для решения возникшей проблемы. Команду `mii-tool` можно использовать для решения проблем, связанных со скоростью и режимом дуплекса, хотя на практике лучше всего задавать для подключений режим автосогласования.

## Обслуживание TDB-файлов

В статье "[Изучаем Linux, 302 \(смешанные среды\): файлы базы данных Trivial Database](#)" рассказывалось о файлах базы данных Trivial Database (TDB), которые Samba использует для сохранения состояния. Если с этими базами данных что-то случится, то у вашего сервера возрастет нагрузка (например, он будет выполнять лишний поиск данных на диске), а, возможно, он не сможет кэшировать информацию, поступающую от удаленных служб. К счастью, в случае возникновения проблем вы можете проверять и восстанавливать целостность TDB-файлов. Временные TDB-файлы можно удалить, и они будут вновь созданы после перезагрузки Samba. Что касается остальных TDB-файлов, то не забывайте выполнять их резервное копирование и проверять резервные копии с помощью команды `tdbsql backup -v`.

## Диагностика клиента

Иногда причиной низкой производительности может быть клиентский компьютер. На нем могут возникать проблемы с дуплексом, компьютерными вирусами, а также другие различные проблемы. Использование в тестах исправного клиентского компьютера поможет исключить сервер из числа потенциальных источников снижения производительности.

## Заключение

Эта статья завершает серию руководств для подготовки к экзамену LPIC 302. Еще раз просмотрите ваши конспекты, и желаем удачи на экзамене!

## Ресурсы

### Научиться

- Оригинал статьи: [Learn Linux, 302 \(Mixed environments\): Performance tuning](#) (EN).
- Man-страница [smb.conf](#) (EN) содержит дополнительные примеры и описания команд, встречающихся в этой статье.
- [Настройка кластера Samba](#) (EN) позволяет обрабатывать больше нагрузки и устраняет риск сбоя из-за неисправности одного из серверов.
- Доктор Гюнтер (Dr. Gunther) из компании [Performance Dynamics](#) (EN) опубликовал книгу, посвященную анализу и настройке производительности. Если вы сдавали экзамен LPIC 301, то вам может быть известен метод анализа PDQ.
- [Алгоритм Нагла](#) (EN) может очень сильно повлиять на скорость сети.
- На Web-сайте [программы сертификации LPIC](#) (EN) вы найдете подробные цели, списки задач и примерные вопросы всех трех уровней сертификации на администратора Linux-систем профессионального института Linux. В частности, на этом сайте представлены [подробные цели экзамена LPI 302](#) (EN), а также [примеры заданий и вопросов](#) (EN).
- Просмотрите всю [серию статей для подготовки к экзаменам института LPI](#) (EN) на сайте developerWorks, основанных на предыдущих целях, определенных до апреля 2009 года, чтобы изучить основы администрирования Linux и подготовиться к экзаменам для получения сертификата администратора Linux.

## **Получить продукты и технологии**

- Если вы ищете [Open Source-инструменты для тестирования производительности](#) (EN), то взгляните на список инструментов на сайте opensourcetesting.org.
- [Wireshark](#) (EN) – лучший инструмент для анализа пакетов. Его можно использовать для анализа пакетов, отправляемых и получаемых вашим сервером.
- [Эталонный тест файловой системы IOzone](#) (EN) помогает выяснить скорость вашей системы хранения.
- [Оцените продукты IBM](#) (EN) любым удобным для вас способом: вы можете загрузить ознакомительные версии продуктов, поработать с ними в онлайновом режиме, использовать их в облачной среде или же потратить несколько часов на изучение [SOA Sandbox](#) (EN) и узнать, как можно эффективно применять сервис-ориентированную архитектуру при разработке программного обеспечения.