

Сегодня в программе

- 1 Инсталляция программ.
- 2 Управление учетными записями пользователя.
- 3 Журналирование.
- 4 Драйверы и ядро.
- 5 Печать.

Содержание

- 1 Инсталляция программ
- 2 Управление учетными записями пользователя
- 3 Журналирование
- 4 Драйверы и ядро
- 5 Печать

Инсталляция программ

В любой ОС нужно уметь ставить и обновлять программы - как внешние, например, Mozilla Firefox, так и части самой ОС, например ядро или интерпретатор bash. Программы могут быть написаны на интерпретируемых (JavaScript, Python, Ruby) либо компилируемых языках (C, C++, C#, Java, Rust).

Для интерпретируемых языков всегда скачивается исходный код, а для компилируемых – либо исходный код, либо бинарные файлы, собранные под архитектуру вашего процессора (i386, amd64, arm, mips, sparc, ..).

Инсталляция программ

В Debian и Ubuntu основной способ установки софта - в формате пакетов - файлов с расширением `.deb` (от слова Debian) из удаленных (remote) репозиториев. Как правило, пакеты содержат уже скомпилированный код, но есть и пакеты с исходниками. Пакеты помимо софта хранят метаинформацию: версию, автора, список зависимостей.

Инсталляция программ

В Debian и Ubuntu основной способ установки софта - в формате пакетов - файлов с расширением `.deb` (от слова Debian) из удаленных (remote) репозиториев. Как правило, пакеты содержат уже скомпилированный код, но есть и пакеты с исходниками. Пакеты помимо софта хранят метаинформацию: версию, автора, список зависимостей. Системы управления пакетами:

- RPM (Red Hat Package Manager): в системах Red Hat, CentOS, SUSE..
- `.deb` (Debian): в системах Debian, Ubuntu, Mint...

Инсталляция программ

В Debian и Ubuntu основной способ установки софта - в формате пакетов - файлов с расширением `.deb` (от слова Debian) из удаленных (remote) репозиториев. Как правило, пакеты содержат уже скомпилированный код, но есть и пакеты с исходниками. Пакеты помимо софта хранят метаинформацию: версию, автора, список зависимостей. Системы управления пакетами:

- RPM (Red Hat Package Manager): в системах Red Hat, CentOS, SUSE..
- `.deb` (Debian): в системах Debian, Ubuntu, Mint...

Двухуровневые средства управления конфигурацией:

- нижний уровень: инсталляция, деинсталляция и запрос пакетов: `rpm` для RPM и `dpkg` для `.deb`
- высший уровень: системы, которые знают, как нужно производить поиск пакетов в Интернете, анализировать зависимости между пакетами и модернизировать все пакеты в системе;
`yum` (Yellowdog Updater, Modified): работает с системой RPM
`apt` (Advanced Package Tool, APT): изначально для работы с пакетами `.deb`, сейчас может работать и с `rpm`.

Инсталляция программ

Команда `dpkg`: управление пакетами `.deb`

Полезные опции:

- `-install`
- `-remove`
- `-l` (перечисляет пакеты, установленные в системе)

```
sudo dpkg --install ./nvi_1.81.6-12_amd64.deb #установка  
исправлений к редактору nvi
```

```
dpkg -l #узнать, что все прошло нормально
```

```
dpkg -l nvi #узнать, что все прошло нормально для конкретного  
пакета
```

```
dpkg -S /bin/uname #показать, из какого пакета тот или иной  
файл
```

Инсталляция программ

Использование высокоуровневых систем управления пакетами в Linux

APT и `dpkg` ставят перед собой следующие задачи:

- упростить определение местонахождения и загрузку пакетов;
- автоматизировать процесс обновления или модернизации систем;
- способствовать управлению зависимостями между пакетами.

Инсталляция программ

АРТ: автоматизирует кучу рутинных процедур.

Включает в себя: apt-get, apt-cache (низкоуровневые команды), которые завернуты в apt.

Утилита apt автоматически загружает и устанавливает зависимые пакеты, благодаря чему можно легко устанавливать или обновлять блоки пакетов.

Инсталляция программ

APT: автоматизирует кучу рутинных процедур.

Включает в себя: apt-get, apt-cache (низкоуровневые команды), которые завернуты в apt.

Утилита apt автоматически загружает и устанавливает зависимые пакеты, благодаря чему можно легко устанавливать или обновлять блоки пакетов.

Основные команды:

- `install` # установить пакет
- `update` # обновляет индекс пакетов в системе Linux или списки пакетов
- `upgrade` # обновить пакеты до последней версии
- `remove` # удалить пакет, но оставить конфигурационные файлы
- `purge` # удалить пакет и удалить конфигурационные файлы
- `autoremove` # удалить неиспользуемые пакеты
- `moov` # промывать

Инсталляция программ

APT: автоматизирует кучу рутинных процедур.

Включает в себя: apt-get, apt-cache (низкоуровневые команды), которые завернуты в apt.

Утилита apt автоматически загружает и устанавливает зависимые пакеты, благодаря чему можно легко устанавливать или обновлять блоки пакетов.

Основные команды:

- `install` # установить пакет
- `update` # обновляет индекс пакетов в системе Linux или списки пакетов
- `upgrade` # обновить пакеты до последней версии
- `remove` # удалить пакет, но оставить конфигурационные файлы
- `purge` # удалить пакет и удалить конфигурационные файлы
- `autoremove` # удалить неиспользуемые пакеты
- `moov` # промывать

Пример: `sudo apt install sudo` # обновить пакет sudo

Инсталляция программ

APT: автоматизирует кучу рутинных процедур.

Включает в себя: apt-get, apt-cache (низкоуровневые команды), которые завернуты в apt.

Утилита apt автоматически загружает и устанавливает зависимые пакеты, благодаря чему можно легко устанавливать или обновлять блоки пакетов.

Основные команды:

- `install` # установить пакет
- `update` # обновляет индекс пакетов в системе Linux или списки пакетов
- `upgrade` # обновить пакеты до последней версии
- `remove` # удалить пакет, но оставить конфигурационные файлы
- `purge` # удалить пакет и удалить конфигурационные файлы
- `autoremove` # удалить неиспользуемые пакеты
- `moov` # промывать

Пример: `sudo apt install sudo` # обновить пакет `sudo` Индексный файл пакетов – это файл или база данных, которые содержат список программных пакетов, определенных в репозиториях, расположенных в файле `/etc/apt/sources.list`. Остальные списки пакетов находятся в каталоге `/etc/apt/sources.list.d`.

Инсталляция программ

Индексный файл пакетов – это файл или база данных, которые содержат список программных пакетов, определенных в репозиториях, расположенных в файле `/etc/apt/sources.list`.

Остальные списки пакетов находятся в каталоге `/etc/apt/sources.list.d`. В каждой строке файла `/etc/apt/sources.list`:

`deb http://site.example.com/debian distribution component1 component2 component3`

- Тип пакета: `deb` или `deb-src` для пакетов в стиле Debian либо `rpm` или `rpm-src` - для пакетов RPM;
- URL-адрес файла, компакт-диска, сервера HTTP или FTP, где находятся пакеты;
- Дистрибутив (название выпуска), если нужно работать с несколькими версиями пакета. Для Ubuntu, например: `jammy` (приложения для текущей версии Ubuntu), `jammy-updates` (рекомендуемые обновления), `jammy-backports` (обновления из более новой системы), `jammy-security` (важные обновления безопасности);
- Возможный список компонентов (категория пакетов в рамках дистрибутива). Для Ubuntu, например: `main` (здесь находятся пакеты, которые официально поддерживаются компанией Canonical), `restricted` (содержит поддерживаемое ПО с закрытым исходным кодом), `universe` (содержит ПО, которое поддерживается сообществом пользователей и разработчиков Ubuntu), `multiverse` (содержит ПО, которое каким-либо образом ограничено либо условиями лицензии, либо юрисдикцией).

Инсталляция программного обеспечения

Установка из исходников:

- `wget http://ftp.gnu.org/gnu/cbc-1.07.tar.gz` # скачать версию калькулятора

Инсталляция программного обеспечения

Установка из исходников:

- `wget http://ftp.gnu.org/gnu/bc/bc-1.07.tar.gz` # скачать версию калькулятора
- `tar -xvf bc-1.07.tar.gz` # распаковать в текущую папку

Инсталляция программного обеспечения

Установка из исходников:

- `wget http://ftp.gnu.org/gnu/bc/bc-1.07.tar.gz` # скачать версию калькулятора
- `tar -xvf bc-1.07.tar.gz` # распаковать в текущую папку
- заходим в распакованный каталог (`cd`), осматриваемся (`ls`). Начать изучение можно с README и INSTALL.

Инсталляция программного обеспечения

Установка из исходников:

- `wget http://ftp.gnu.org/gnu/bc/bc-1.07.tar.gz` # скачать версию калькулятора
- `tar -xvf bc-1.07.tar.gz` # распаковать в текущую папку
- заходим в распакованный каталог (`cd`), осматриваемся (`ls`). Начать изучение можно с `README` и `INSTALL`.
- `./configure` # подготавливает исходники к сборке на машине (смотрит, что у стоит, например компилятор, библиотеки каких версий, ...). Создает файл `Makefile` для следующего шага.

Инсталляция программного обеспечения

Установка из исходников:

- `wget http://ftp.gnu.org/gnu/bc-1.07.tar.gz` # скачать версию калькулятора
- `tar -xvf bc-1.07.tar.gz` # распаковать в текущую папку
- заходим в распакованный каталог (`cd`), осматриваемся (`ls`). Начать изучение можно с `README` и `INSTALL`.
- `./configure` # подготавливает исходники к сборке на машине (смотрит, что у стоит, например компилятор, библиотеки каких версий, ...). Создает файл `Makefile` для следующего шага.
- `make` # запуск утилиты сборки, она в нужной последовательности вызовет компилятор и другие команды (например, для сборки документации)

Инсталляция программ

Другие системы установки пакетов:

- `pip (pip3)` # для питоновских программ и модулей
- `npm` # для NodeJS приложений и модулей
- `nuget` # для C#

Управление учетными записями пользователя

Пользователь: 32-битное целое число без знака aka ID или UID.

Управление учетными записями пользователя

Пользователь: 32-битное целое число без знака aka ID или UID.
Файл `/etc/passwd`: содержит список пользователей, которые известны системе.

Управление учетными записями пользователя

Пользователь: 32-битное целое число без знака aka ID или UID.

Файл `/etc/passwd`: содержит список пользователей, которые известны системе. Пароли: хранятся в файле `/etc/shadow` (защищенный).

Управление учетными записями пользователя

Пользователь: 32-битное целое число без знака aka ID или UID.
Файл `/etc/passwd`: содержит список пользователей, которые известны системе. Пароли: хранятся в файле `/etc/shadow` (защищенный). В процессе регистрации пользователя система обращается к файлу `/etc/passwd` в поисках идентификатора пользователя и его домашнего каталога, а также другой информации. Каждая строка файла:

- регистрационное имя;
- "заполнитель пароля";
- идентификатор пользователя UID;
- идентификатор группы по умолчанию GID;
- поле GECOS (полное имя и т.д.);
- домашний каталог;
- регистрационная оболочка.

Управление учетными записям пользователя

Файл `/etc/shadow`: хранение паролей. Каждый файл содержит одну строку для каждого пользователя:

- регистрационное имя;
- зашифрованный пароль;
- дата последнего изменения пароля;
- минимальное число дней между изменениями паролей;
- максимальное число дней между изменениями паролей;
- количество дней до истечения срока действия пароля, когда выдается предупреждение;
- количество дней по истечении срока действия пароля, когда учетная запись аннулируется;
- дата истечения срока действия учетной записи;
- зарезервированное поле, в настоящее время пустое.

Обязательно для заполнения: только первые два.

Управление учетными записями пользователей

Файл `/etc/group`: содержит имена UNIX-групп и списки членов каждой группы.
Каждая строка содержит:

- имя группы;
- зашифрованный пароль или заполнитель;
- идентификатор группы;
- список членов, разделенный запятыми (пробелов быть не должно).

Установка пароля группы: `gpasswd`, зашифрованная форма: `/etc/gshadow`.

Управление учетными записями пользователей

Файл `/etc/group`: содержит имена UNIX-групп и списки членов каждой группы. Каждая строка содержит:

- имя группы;
- зашифрованный пароль или заполнитель;
- идентификатор группы;
- список членов, разделенный запятыми (пробелов быть не должно).

Установка пароля группы: `gpasswd`, зашифрованная форма: `/etc/gshadow`.

Традиция UNIX: новые пользователи добавляются в группу, название которой отражает их категорию, например, "students" или "finance". Однако, подобная традиция повышает вероятность доступа пользователей к файлам друг друга из-за неаккуратной установки прав.

Чтобы избежать: создавать для каждого пользователя уникальную группу с помощью утилит `useradd` и `adduser`.

Чтобы пользователи могли обмениваться файлами с помощью группового механизма, следует создать отдельные группы. Идея персональных групп: не препятствовать использованию групп как таковых, а просто создать более ограниченную группу *по умолчанию* для каждого пользователя, чтобы файла не были предоставлены для совместного пользователя непреднамеренно.

Управление группами: `groupadd`, `groupmod`, `groupdel`.

Подключение пользователей вручную: основные действия

Этапы:

- отредактировать файлы `passwd` и `shadow`, чтобы создать учетную запись пользователя (команда `vipw`);
- добавить запись нового пользователя в файл `/etc/group` (команда `vipg`);
- задать первоначальный пароль (команда `passwd`);
- создать для нового пользователя домашний каталог, назначив его владельца с помощью команды `chown` и задав режим доступа с помощью команды `chmod`;

Добавление пользователей с помощью сценариев: useradd, adduser

Параметры, хранящиеся в файле `/etc/default/useradd`, задают расположение домашних каталогов и оболочку по умолчанию для новых пользователей. Эти значения устанавливаются по умолчанию для новых пользователей с помощью команды `useradd`.

```
useradd -D # вывести текущие значения
```

```
sudo useradd -D -s /bin/bash # установить оболочку bash по умолчанию
```

```
sudo useradd vvp # создать пользователя vvp с параметрами по умолчанию
```

```
sudo useradd -c "Vasilii Vasilievich Pokrovskii" /home/os/pokrovskii -g vvp -G faculty  
-m -s /bin/tcsh vvp
```

Добавление пользователей с помощью сценариев: useradd, adduser

Параметры, хранящиеся в файле `/etc/default/useradd`, задают расположение домашних каталогов и оболочку по умолчанию для новых пользователей. Эти значения устанавливаются по умолчанию для новых пользователей с помощью команды `useradd`.

`useradd -D #` вывести текущие значения

`sudo useradd -D -s /bin/bash #` установить оболочку `bash` по умолчанию

`sudo useradd vvp #` создать пользователя `vvp` с параметрами по умолчанию

`sudo useradd -c "Vasilii Vasilievich Pokrovskii" /home/os/pokrovskii -g vvp -G faculty -m -s /bin/tcsh vvp` команда `adduser`: дополнение к семейству команд `useradd`, предоставляет несколько сценариев более высокого уровня.

Дополнительные команды настраиваются в файле `/etc/adduser.conf`, где указываются такие параметры:

- правила размещения домашних каталогов: по группам, по имени пользователя и т.д.
- настройки разрешения для новых домашних каталогов;
- диапазоны `UID` и `GID` для системных и общих пользователей;
- возможность создания отдельных групп для каждого пользователя;
- квоты диска;
- сопоставление имен пользователей и групп с помощью регулярных выражений.

Безопасное удаление учетных записей пользователей и файлов

После удаления учетной записи пользователя, необходимо убедиться, что в системе не осталость файлов с его идентификатором.

`sudo find filesystem -xdev -nouser #` проверка файловых систем по отдельности

специальная команда: `deluser` — сценарий на языке Perl, который вызывает обычную команду `userdel`, аннулируя все результаты работы команды `adduser`.

Блокирование регистрационных имен пользователей

Иногда возникает необходимость временно отключить учетную запись пользователя. Это можно сделать, поставим звезду (*) в файле `/etc/shadow`.

Команды:

```
usermod -L user # блокировать user
```

```
usermod -U user # деблокировать user
```

Альтернатива: замена интерпретатора команд пользователя на программу, выводящую сообщение об оболочке.

Журналирование

Управление журналированием:

- сбор журналов из различных источников;
- предоставление структурированного интерфейса для запросов, анализа, фильтрации и мониторинга сообщений;
- управление хранением и истечением срока действия сообщений, чтобы информация сохранялась до тех пор, пока она потенциально полезна или юридически необходима (но не на неопределенный срок).

Журналирование

Управление журналированием:

- сбор журналов из различных источников;
- предоставление структурированного интерфейса для запросов, анализа, фильтрации и мониторинга сообщений;
- управление хранением и истечением срока действия сообщений, чтобы информация сохранялась до тех пор, пока она потенциально полезна или юридически необходима (но не на неопределенный срок).

Демон syslog: сортирует сообщения, сохраняет их в файлы и пересылает их другому хосту по сети (не идеален, многие приложения реализуют свои журнальные службы, с интерфейсом и условиями хранения). Альтернатива: журнал от systemd.

Журналирование

Управление журналированием:

- сбор журналов из различных источников;
- предоставление структурированного интерфейса для запросов, анализа, фильтрации и мониторинга сообщений;
- управление хранением и истечением срока действия сообщений, чтобы информация сохранялась до тех пор, пока она потенциально полезна или юридически необходима (но не на неопределенный срок).

Демон `syslog`: сортирует сообщения, сохраняет их в файлы и пересылает их другому хосту по сети (не идеален, многие приложения реализуют свои журнальные службы, с интерфейсом и условиями хранения). Альтернатива: журнал от `systemd`. Просмотреть записи в журнале `systemd`: команда `journalctl`.
`journalctl -u ssh` # сообщения, касающиеся демона SSH
`journalctl -f` # выводить на экран новые сообщения по мере их поступления
Файл `syslog`: `/var/log/syslog`

Драйверы и ядро

Ядро — главная часть ОС UNIX, устанавливающая правила, распределяющая ресурсы и предоставляющая пользователям основные услуги.

Драйверы и ядро

Ядро — главная часть ОС UNIX, устанавливающая правила, распределяющая ресурсы и предоставляющая пользователям основные услуги. Функции абстрактного высокоуровневого интерфейса:

- управление и абстрагирование аппаратного обеспечения;
- процессы и потоки, а также способы взаимодействия между ними;
- управление памятью (виртуальная память и защита областей памяти);
- функции ввода-вывода (файловые системы, сетевые интерфейсы, последовательные интерфейсы и т.д.)
- организационные функции (начало и завершение работы, таймеры, многозадачность и т.д.)

Устройства и их драйверы

Драйвер устройства — уровень абстракции, управляющий взаимодействием системы с конкретным типом аппаратного обеспечения, так что остальная часть ядра не должна знать ее специфику.

Устройства и их драйверы

Драйвер устройства — уровень абстракции, управляющий взаимодействием системы с конкретным типом аппаратного обеспечения, так что остальная часть ядра не должна знать ее специфику. В основном драйвера являются частью ядра, а не пользовательским процессом. Тем не менее, получить доступ к драйверу можно как из ядра, так и из пространства пользователя. Для этого: используются файлы устройств, записанные в каталоге `/dev`.

С каждым файлом устройства в каталоге `/dev` связаны старший и младший номера устройств, посредством которых ядро преобразует обращения к файлу устройства в вызовы нужного драйвера.

Старший номер устройства обозначает драйвер, за которым закреплен данный файл (иначе говоря, он определяет тип устройства). Младший номер устройства указывает на то, к какому конкретно устройству данного типа следует обращаться. Иногда называют номером модуля.

Узнать номер устройства: `ls -l /dev/sda`

Устройства и их драйверы

Драйвер устройства — уровень абстракции, управляющий взаимодействием системы с конкретным типом аппаратного обеспечения, так что остальная часть ядра не должна знать ее специфику. В основном драйвера являются частью ядра, а не пользовательским процессом. Тем не менее, получить доступ к драйверу можно как из ядра, так и из пространства пользователя. Для этого: используются файлы устройств, записанные в каталоге `/dev`.

С каждым файлом устройства в каталоге `/dev` связаны старший и младший номера устройств, посредством которых ядро преобразует обращения к файлу устройства в вызовы нужного драйвера.

Старший номер устройства обозначает драйвер, за которым закреплен данный файл (иначе говоря, он определяет тип устройства). Младший номер устройства указывает на то, к какому конкретно устройству данного типа следует обращаться. Иногда называют номером модуля.

Узнать номер устройства: `ls -l /dev/sda` Файлы устройств бывают двух типов: блочные и символьные. Чтение из блочного устройства и запись в него осуществляется по одному блоку за раз, из символьного — чтение побайтово.

Устройства и их драйверы

Драйвер устройства — уровень абстракции, управляющий взаимодействием системы с конкретным типом аппаратного обеспечения, так что остальная часть ядра не должна знать ее специфику. В основном драйвера являются частью ядра, а не пользовательским процессом. Тем не менее, получить доступ к драйверу можно как из ядра, так и из пространства пользователя. Для этого: используются файлы устройств, записанные в каталоге `/dev`.

С каждым файлом устройства в каталоге `/dev` связаны старший и младший номера устройств, посредством которых ядро преобразует обращения к файлу устройства в вызовы нужного драйвера.

Старший номер устройства обозначает драйвер, за которым закреплен данный файл (иначе говоря, он определяет тип устройства). Младший номер устройства указывает на то, к какому конкретно устройству данного типа следует обращаться. Иногда называют номером модуля.

Узнать номер устройства: `ls -l /dev/sda` Файлы устройств бывают двух типов: блочные и символьные. Чтение из блочного устройства и запись в него осуществляется по одному блоку за раз, из символьного — чтение побайтово.

Псевдоустройства: устройства, которых нет, а драйвер к которому есть.

Примеры: псевдотерминал (PTY), `/dev/zero`, `/dev/null`, `/dev/urandom`.

Устройства и их драйверы

Драйвер устройства — уровень абстракции, управляющий взаимодействием системы с конкретным типом аппаратного обеспечения, так что остальная часть ядра не должна знать ее специфику. В основном драйверы являются частью ядра, а не пользовательским процессом. Тем не менее, получить доступ к драйверу можно как из ядра, так и из пространства пользователя. Для этого: используются файлы устройств, записанные в каталоге `/dev`.

С каждым файлом устройства в каталоге `/dev` связаны старший и младший номера устройств, посредством которых ядро преобразует обращения к файлу устройства в вызовы нужного драйвера.

Старший номер устройства обозначает драйвер, за которым закреплен данный файл (иначе говоря, он определяет тип устройства). Младший номер устройства указывает на то, к какому конкретно устройству данного типа следует обращаться. Иногда называют номером модуля.

Узнать номер устройства: `ls -l /dev/sda` Файлы устройств бывают двух типов: блочные и символьные. Чтение из блочного устройства и запись в него осуществляется по одному блоку за раз, из символьного — чтение побайтово.

Псевдоустройства: устройства, которых нет, а драйвер к которому есть.

Примеры: псевдотерминал (PTY), `/dev/zero`, `/dev/null`, `/dev/urandom`. Создание файлов устройств: `mknod имя_файла тип старший младший`

Конфигурирование ядра Linux

Три основных метода конфигурирования ядра Linux:

- модификация настраиваемых (динамических) параметров ядра;
- повторное создание ядра (компиляция исходных файлов ядра с возможными модификациями и дополнениями);
- динамическая загрузка новых драйверов и модулей в существующее ядро.

Конфигурирование параметров ядра linux

Для повышения гибкости: специальные информационные каналы, позволяющие админу динамически корректировать различные параметры, определяющие размеры внутренних таблиц ядра и его поведение в конкретных ситуациях. Каналы расположены в файловой системе /proc, которая представляет собой интерфейс между ядром и приложениями пользовательского уровня.

Конфигурирование параметров ядра linux

Для повышения гибкости: специальные информационные каналы, позволяющие админу динамически корректировать различные параметры, определяющие размеры внутренних таблиц ядра и его поведение в конкретных ситуациях. Каналы расположены в файловой системе /proc, которая представляет собой интерфейс между ядром и приложениями пользовательского уровня. В каталоге содержатся /proc/sys содержатся специальные файлы, позволяющие просматривать и динамически изменять значения параметров ядра. Изменить максимальное число файлов, которые одновременно могут быть открыты в системе:

```
cat /proc/sys/fs/file-max  
echo 100500 > /proc/sys/fs/file-max
```

Конфигурирование параметров ядра linux

Для повышения гибкости: специальные информационные каналы, позволяющие админу динамически корректировать различные параметры, определяющие размеры внутренних таблиц ядра и его поведение в конкретных ситуациях. Каналы расположены в файловой системе `/proc`, которая представляет собой интерфейс между ядром и приложениями пользовательского уровня. В каталоге `/proc/sys` содержатся специальные файлы, позволяющие просматривать и динамически изменять значения параметров ядра. Изменить максимальное число файлов, которые одновременно могут быть открыты в системе:

```
cat /proc/sys/fs/file-max
```

```
echo 100500 > /proc/sys/fs/file-max
```

При перезагрузке изменения теряются!

Чтобы не терялись: команда `sysctl` в сочетании с `/etc/sysctl.conf`

```
sysctl net.ipv4.ip_forward=0 # отключает IP-пересылку
```

Сборка ядра

Большинство дистрибутивов инсталлируют исходные файлы ядра в каталог `/usr/src/kernels`. Чтобы можно было создавать ядро системы, должен быть инсталлирован пакет его исходных кодов.

Сборка ядра

Большинство дистрибутивов инсталлируют исходные файлы ядра в каталог `/usr/src/kernels`. Чтобы можно было создавать ядро системы, должен быть инсталлирован пакет его исходных кодов. Весь процесс настройки ядра сосредоточен вокруг файла `.config`, находящегося в каталоге исходных кодов ядра.

Сборка ядра

Большинство дистрибутивов инсталлируют исходные файлы ядра в каталог `/usr/src/kernels`. Чтобы можно было создавать ядро системы, должен быть инсталлирован пакет его исходных кодов. Весь процесс настройки ядра сосредоточен вокруг файла `.config`, находящегося в каталоге исходных кодов ядра. Утилита `make`: `make xconfig` (если оболочка KDE), `make gconfig` (если оболочка GNOME).

Отображается окно конфигурации, в котором можно выбрать драйверы устройств, добавляемые к ядру.

Ещё вариант: `make menuconfig` (если нет графической оболочки).

Перенос конфигурации существующего ядра в новую версию ядра: `make oldconfig` для считывания ранее использовавшегося файла `.config` и определения только тех параметров, которые изменились или являются новыми.

Компиляция ядра

После формирования файла `.config` ещё ряд действий для получения ядра:

- перейти в каталог верхнего уровня (`cd`), содержащий исходные коды ядра;
- выполнить команду `make xconfig, make gconfig, make menuconfig`;
- выполнить `make clean`;
- выполнить `make modules_install`;
- выполнить `make install`

Добавление драйвера устройства в Linux

Драйверы распространяются в одной из трех форм:

- исправления конкретной версии ядра;
- загружаемого модуля;
- инсталляционного сценария или пакета, устанавливающего соответствующие исправления (наиболее распространенная форма).

Установка исправлений конкретной версии ядра:

```
cd путь_к_исходным_кодам_ядра ; patch -p1 <  
файл_исправления
```

Печать

CUPS (Common UNIX Printing System) — система печати в UNIX.
Компоненты печати:

- диспетчер печати (spool);
- утилиты пользовательского уровня, передающие команды диспетчеру печати;
- серверные службы, управляющие самими печатающими устройствами (обычно скрыты);
- сетевой протокол, позволяющий диспетчерам общаться и передавать задания.

Печать

Серверы CUPS — это веб-серверы, а клиенты CUPS — это веб-клиенты. Клиентами CUPS могут быть как команды, так и приложения с графическим пользовательским интерфейсом.

Сервер CUPS предоставляет полнофункциональный веб-интерфейс на порту 631. В основе всех взаимодействий между серверами CUPS и клиентами лежит протокол HTTP (протокол печати в Интернете IPP, Internet Printing Protocol).

Печать

Серверы CUPS — это веб-серверы, а клиенты CUPS — это веб-клиенты. Клиентами CUPS могут быть как команды, так и приложения с графическим пользовательским интерфейсом.

Сервер CUPS предоставляет полнофункциональный веб-интерфейс на порту 631. В основе всех взаимодействий между серверами CUPS и клиентами лежит протокол HTTP (протокол печати в Интернете IPP, Internet Printing Protocol).

Можно из командной строки:

`lpr foo.pdf` # передить копии файлов серверу CUPS `cupsd`, который сохраняет их в очереди на печать.

Печать

Серверы CUPS — это веб-серверы, а клиенты CUPS — это веб-клиенты. Клиентами CUPS могут быть как команды, так и приложения с графическим пользовательским интерфейсом.

Сервер CUPS предоставляет полнофункциональный веб-интерфейс на порту 631. В основе всех взаимодействий между серверами CUPS и клиентами лежит протокол HTTP (протокол печати в Интернете IPP, Internet Printing Protocol). Можно из командной строки:

```
lpr foo.pdf # передить копии файлов серверу CUPS cupsd, который сохраняет их в
очереди на печать. команда lprq — запрашивает с сервера CUPS информацию о
состоянии задания и форматирует ее для отображения;
lprm идентификатор_задания # удалить задание;
lpstat -t # хороший, годный отчет об общем состоянии сервера печати
```

100

```

lpw foo.pdf # передить копии файлов серверу CUPS cupsd, который сохраняет их в
очереди на печать. команда lpq — запрашивает с сервера CUPS информацию о
состоянии задания и форматирует ее для отображения;
lpw идентификатор_задания # удалить задание;
lpstat -t # хороший, годный отчет об общем состоянии сервера печати
файл конфигурации демона cupsd:
/etc/cups/cupsd.conf #обычно тут

```

Печать

Серверы CUPS — это веб-серверы, а клиенты CUPS — это веб-клиенты. Клиентами CUPS могут быть как команды, так и приложения с графическим пользовательским интерфейсом.

Сервер CUPS предоставляет полнофункциональный веб-интерфейс на порту 631. В основе всех взаимодействий между серверами CUPS и клиентами лежит протокол HTTP (протокол печати в Интернете IPP, Internet Printing Protocol).

Можно из командной строки:

`lpr foo.pdf` # передить копии файлов серверу CUPS `cupsd`, который сохраняет их в очереди на печать. команда `lprq` — запрашивает с сервера CUPS информацию о состоянии задания и форматирует ее для отображения;

`lprm` идентификатор_задания # удалить задание;

`lpstat -t` # хороший, годный отчет об общем состоянии сервера печати
файл конфигурации демона `cupsd`:

`/etc/cups/cupsd.conf` #обычно тут

перезапуск демона печати:

`systemctl restart org.cups.cupsd.service`

Задания

- Запустить из-под Linux программу для Windows (без виртуалки!!!)
- Напишите однострочную команду, которая у всех файлов, владельцем которых является root, сменит владельца на текущего пользователя;
- Создать пользователя, ограничить количество потребляемой памяти;
- Пересобрать ядро Linux.

Конец темы

Спасибо за внимание!