**Введение**

**Linux** — общее название UNIX-подобных операционных систем на основе одноимённого ядра и собранных для него библиотек и системных программ, разработанных в рамках проекта GNU. В данной операционной системе предусмотрено активное использование консоли, содержащей сотни команд. Используя команды в Linux, пользователь может оперативно совершать массу действий: открытие, перемещение и копирование файлов, просмотр различной информации и статистики, мониторинг и отладка, получение подробной информации о системе, видоизменение программной и визуальной части системы и т.п. Возможность видоизменять систему командами делает Linux очень универсальным инструментом. Командная строка, оболочка, консоль - эти термины используются для обозначения взаимодействия человека-оператора с операционной системой с применением стандартных устройств ввода-вывода, обеспечивающих ввод команд и получение результатов их выполнения. В операционных системах семейства Linux подобное взаимодействие обеспечивается специальным программным продуктом - оболочкой (shell) . Наиболее распространенной из них, является оболочка проекта GNU *bash*(Bourne Again SHell).

Цель курсовой работы: подробно разобрать команды и ключи операционной системы Linux, относящиеся к категории System configuration, а также рассмотреть область их применения.

Задача данной курсовой работы заключается в том, чтобы поближе познакомиться с оболочкой bash и используемыми в повседневной жизни IT-специалиста утилитами.

**ГЛАВА I.**

**§1.Что нужно знать о Linux пользователю**

**Вход в систему**

Linux — это многопользовательская система. На практике это означает, что для работы в системе нужно в ней *зарегистрироваться*, т. е. дать понять системе, кто именно находится за монитором и клавиатурой. Вместо формального «зарегистрироваться в системе» обычно используют выражение «войти в систему». Операционная система представляется чем-то вроде замкнутого помещения, внутри которого можно оказаться, только успешно проникнув через «дверь» — пройдя процедуру регистрации. Наиболее распространённый способ регистрации на сегодняшний день — использование **системных имён** (login name) и паролей (password). Это надёжное средство убедиться, что с системой работает тот, кто нужно, если пользователи хранят свои пароли в секрете и если пароль достаточно сложен и не слишком короток (иначе его легко угадать или подобрать).

Загрузка заканчивается интерфейсом входа в систему: выводится приглашение ввести системное имя пользователя (login:) и пароль. Если вы выбрали загрузку в графический режим, то можно не вводить системное имя вручную, а нажать на кнопку с нужным именем, однако пароль при этом всё равно нужно ввести самостоятельно.

Пользователи обычно создаются непосредственно в процессе установки системы, однако всегда можно добавить новых пользователей или удалить существующих при помощи стандартных средств управления пользователями.

**Графический и текстовый интерфейс**

В операционной системе Linux пользователю доступны два режима работы: графический и текстовый. В текстовом режиме недоступны возможности графических интерфейсов: рисование окон произвольной формы и размера, поддержка миллионов цветов и т.д. Все возможности текстового режима ограничены набором текстовых и псевдографических символов и несколькими десятками базовых цветов. Тем не менее в Linux в текстовом режиме можно выполнять практически любые действия в системе (кроме тех, которые требуют непосредственного *просмотра* изображений). Текстовый режим в Linux — это полнофункциональный способ управления системой благодаря интерфейсу командной строки.

Бывают ситуации, когда графический режим недоступен или неработоспособен (удалённый доступ по сети, проблемы с поддержкой видеокарты, сбои системы и др.). В таких случаях всегда остаётся возможность работать в текстовом режиме, поскольку его возможности поддерживаются непосредственно графическим оборудованием и не требуют специальных драйверов или настройки.

В процессе работы Linux активно несколько *виртуальных консолей*. Каждая виртуальная консоль доступна по одновременному нажатию *Alt* и функциональной клавиши с номером этой консоли.

Благодаря виртуальным консолям каждый компьютер, на котором работает Linux, предоставляет возможность зарегистрироваться и получить доступ к системе одновременно нескольким пользователям. Даже если в распоряжении всех пользователей есть только один монитор и одна системная клавиатура, эта возможность небесполезна: можно переключаться между виртуальными консолями так, как если бы вы переходили от одного монитора с клавиатурой к другому, подавая время от времени команды и следя за выполняющимися там программами. Более того, ничто не препятствует зарегистрироваться в системе несколько раз под одним и тем же *системным именем* — это один из способов организовать параллельную работу над несколькими задачами.

**Завершение работы**

Linux *нельзя* выключать, просто лишив компьютер электропитания. Множество информации, которая должна располагаться на диске, система держит в оперативной памяти для повышения быстродействия. Неожиданное выключение питания приводит к потере этой информации.

**Командная строка**

Командная строка — это способ организации интерфейса, в котором каждая строка, введённая пользователем — это команда системе, которую та должна выполнить. Термин «командная строка» происходит от того, что команды вводятся обычно в одну строку, которая завершается нажатием клавиши «ввод» (Enter). В Linux этот вид интерфейса всегда был основным, а потому хорошо развитым.

Первое слово в такой строке — это, как правило, имя исполняемого файла —*программы*, все остальные слова — *параметры*. Программа выполняет нужные пользователю действия, но может делать это по-разному в зависимости от полученных параметров. Параметры могут быть общими, например имя файла, который нужно обработать, или специфическими для этой программы модификаторами выполнения.

Чтобы получить командную строку, пользователь должен войти в систему и запустить программу, которая будет принимать его команды и передавать их на выполнение — командную оболочку (её ещё называют *интерпретатор командной строки*, просто *оболочка*, по-английски «shell»).

Командная строка начинается *приглашением* — это подсказка, свидетельствующая о том, что система готова принимать команды пользователя. В процессе выполнения команды система может вывести те или иные сообщения, а когда выполнение завершается — вновь выводится приглашение командной строки. Приглашение может быть оформлено по-разному, но чаще всего оно заканчивается символом «$». В примерах в документации этим символом условно обозначается командная строка: всё, что следует после него и до конца строки — это и есть команда, которую нужно ввести. Пока не нажата клавиша Enter, набранную команду можно редактировать.

**Понятие командной оболочки**

Современные операционные системы создают уровень абстракции, позволяющий использовать терминалы совместно с командными оболочками. Оболочки имеют расширенные возможности работы в текстовом режиме и идут в комплекте с любым дистрибутивом Linux.

*Командная оболочка* — это специальный программный интерпретатор. Он является посредником между пользователем и операционной системой. При вводе команды в окне терминала оболочка перехватывает её и осуществляет дополнительную обработку, что позволяет внедрить элементы программирования. Например, вы можете запустить приложение и записать результат его выполнения в файл. Операция не потребует дополнительных усилий, интерпретатор сделает всё самостоятельно, а запись команды уместится в одну короткую строку. Рассмотрим командную оболочку Bash.

История создания Bash берет свое начало с 1987 года. Командная оболочка была разработана Брайаном Фоксом на основе ранее существовавшей Bourne shell. Она обладает широким спектром возможностей. В последнее время появилось множество других интерпретаторов, значительно обгоняющих Bash по скорости работы. Тем не менее, многочисленное сообщество и установка практически на всех дистрибутивах Linux сохраняет её популярность.

***Встроенные возможности:***

* Использование конвейеров для передачи результата одной программы в другую.
* Встроенный язык программирования с условными операторами, циклами и т. д.
* Запуск набора команд из файла (скрипта).
* Авто дополнение имен файлов и директорий.
* Просмотр истории введенных команд.
* Перенаправление ввода-вывода.
* Использование оболочковых переменных

При открытии текстового терминала Linux командная оболочка будет загружена в него автоматически. От пользователя не требуется никаких дополнительных операций.

**Установка Linux на виртуальную машину VirtualBox**

VirtualBox (Oracle VM VirtualBox) - бесплатное программное средство виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, Mac OS X, Solaris/OpenSolaris, ReactOS, DOS и других.

Если упрощенно, то VirtualBox - это комплект прикладных программ, системных служб и драйверов, эмулирующих новое компьютерное оборудование в среде операционной системы, где работает VirtualBox. На виртуальном компьютере (виртуальной машине, сокращенно - ВМ), создаваемом в его среде можно установить практически любую операционную систему ( гостевую ОС ) и использовать ее параллельно с основной. Так, например, на реальном компьютере с Windows можно установить виртуальную машину (VM - Virtual Machine) с операционной системой семейства Linux и пользоваться обеими ОС одновременно. Кроме того, можно настроить взаимодействие между этими системами по локальной сети, обмен данными через сменные носители, общие папки и т. п. Также, текущее состояние виртуальной машины ( и состояние установленной на ней операционной системы) можно зафиксировать, и при необходимости, в любой момент времени - выполнить полный откат на это состояние.

***Установка Linux на виртуальную машину Oracle VM Virtualbox***

   Процесс установки Linux или какой-либо иной ОС состоит из двух частей:  
- создание виртуальной машины под выбранную для установки гостевую ОС   
- определение источника с установочным дистрибутивом и запуск процесса установки  
Процесс установки новой ОС на виртуальной машине, практически ничем не отличается от установки на реальной - выполняется загрузка VM с установочного диска и дальнейшее следование указаниям инсталлятора. Источник загрузки определяется настройками виртуальной машины (По умолчанию, порядок загрузки следующий - дискета, CD-ROM, жесткий диск, Сеть).

**ГЛАВА II. РЕАЛИЗАЦИЯ КОМАНД SYSTEM INFORMATION**

**Команда stat**

**Stat**

Команда stat выводит содержимое полей дескриптора файла или статус файловой системы .

|  |  |  |
| --- | --- | --- |
| Команда | Ключи | Описание Ключей |
| Stat | -с | применяется указанный формат вывода FORMAT (см.далее), вместо используемого по умолчанию |
| Stat | -f | выводит статус файловой системы (на которой расположен FILE) вместо статуса файла |
| Stat | -L | выводит информацию о статусе оригинального файла, с которым связана ссылка FILE |
| Stat | -t | выводит информацию в сокращённой (сжатой) форме |
| Stat | --help | выводит это описание программы и завершает её работу |
| Stat | --version | выводит информацию о версии программы и завершает её работу |

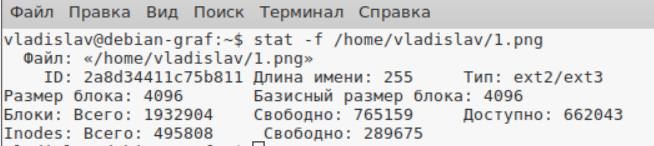
**Пример:**

***Задача:***

Узнать статус файловой системы, на которой находится файл.

***Команда:***

|  |
| --- |
| **Stat –f** |



**Команда free**

**Free**

Команда free предоставляет информацию об использованной и неиспользованной памяти, а так же о разделе подкачки (swap).

|  |  |  |
| --- | --- | --- |
| Команда | Ключи | Описание ключей |
| free | -b | вывод памяти в байтах |
| free | -m | вывод памяти в мегабайтах |
| free | -g | вывод памяти в гигабайтах |
| free | -s | указание времени обновления информации |
| free | -c | указание количества обновлений , используется совместно с ключем s |

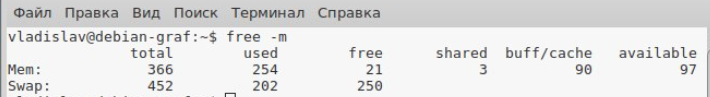
**Примеры:**

***Задача1:***

Узнать информацию об использованной и неиспользованной памяти , и вывести данные в Мб.

***Команда:***

|  |
| --- |
| **Free -m** |

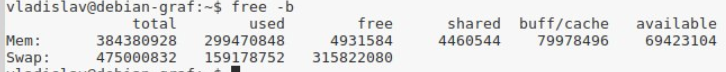


***Задача2:***

Узнать информацию об использованной и неиспользованной памяти , и вывести данные в Байтах.

***Команда:***

|  |
| --- |
| **Free -b** |

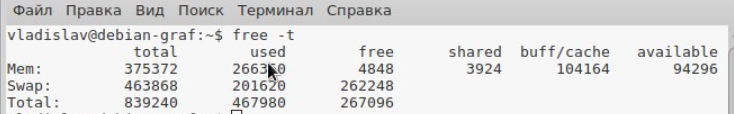


***Задача3:***

Узнать полную информацию об использованной и неиспользованной памяти , и вывести данные.

***Команда:***

|  |
| --- |
| **Free -t** |



**Команда uname**

**Uname**

Команда uname выдает информацию о текущей системе в стандартный выходной поток.

|  |  |  |
| --- | --- | --- |
| Команда | Ключи | Описание Ключей |
| uname | -a | Выдает основную информацию, доступную в настоящий момент в системе. |
| uname | -i | Выдает имя аппаратной реализации (платформы). |
| uname | -m | Выдает имя (класс) аппаратного обеспечения машины. Эту опцию использовать не рекомендуется; |
| uname | -n | Выдает имя узла (nodename) (имя узла - это имя, под которым система известна в сети). |
| uname | -p | Выдает ISA или тип процессора текущего хоста. |
| uname | -r | Выдает информацию о релизе операционной системы. |
| uname | -s | Выдает имя операционной системы. Эта опция используется по умолчанию |

**Пример:**

***Задача:***

Узнать класс аппаратного обеспечения машины.

***Команда:***

|  |
| --- |
| **Uname -m** |



**Команда htop**

**Htop**

Продвинутый монитор процессов, написанный для Linux. Он был задуман заменить стандартную программу top. Htop показывает динамический список системных процессов, список обычно выравнивается по использованию ЦПУ. В отличие от top, htop показывает все процессы в системе. Также показывает время непрерывной работы, использование процессоров и памяти.

**Столбцы:**

PID - идентификатор процесса.

USER - владелец процесса.

PRI - текуший приоритет (влияет на процессорное время, отводимое процессу, значение по умолчанию - 20; чем меньше приоритет, тем больше времени отводится процессу, следовательно он выполняется быстрее).

NI - величина изменения приоритета относительно значения PRI (клавиши F7, F8).

VIRT - общий объем виртуальной памяти, используемой процессом. Включает в себя: область кода (CODE), данные (DATA), разделяемые библиотеки (SHARED) и страницы, перемещенные в swap-область памяти. Если приложение потребовало от ядра выделить ему 100Мб памяти, а использует всего 5 Мб, данный столбец всё равно будет показывать цифру 100.

(CODE - объем памяти, содержащий исполняемый код процесса.

DATA - объем памяти, занятой данными, используемыми процессом в ходе выполнения.

SWAP - объем памяти, используемой процессом, но перемещенной в swap-область.)

RES - количество резидентной (не перемещаемой в swap) памяти в килобайтах. Если приложение потребовало от ядра выделить ему 100Мб памяти, а использует всего 5 Мб, то данный столбец покажет 5. Но здесь есть два ньюанса:

а) RES не показывает сколько данных было перемещено в swap,

б) часть RES-памяти может быть разделяемой.

SHR - количество разделяемой (shared) памяти программы в килобайтах, т.е. памяти, которая может быть использована другими приложениями.

S - состояние процесса:

S — так называемое состояние сна;

R — состояние выполнения;

D — состояние ожидания.

CPU% - использование процессора в процентном отношении.

MEM% - использование процессом памяти в процентном отношении.

TIME+ - время работы процесса.

Command - указывает на команду, которой был запущен процесс.

**Управление:**

F1 - справка;

F2 - настройки;

F3 - поиск процесса;

F4 - сортировка списка процессов (от большего к меньшему или от меньшего к большему);

F5 - устанавливает древовидное отображение (корни - родительские процессы, а листья - дочерние) и наоборот;

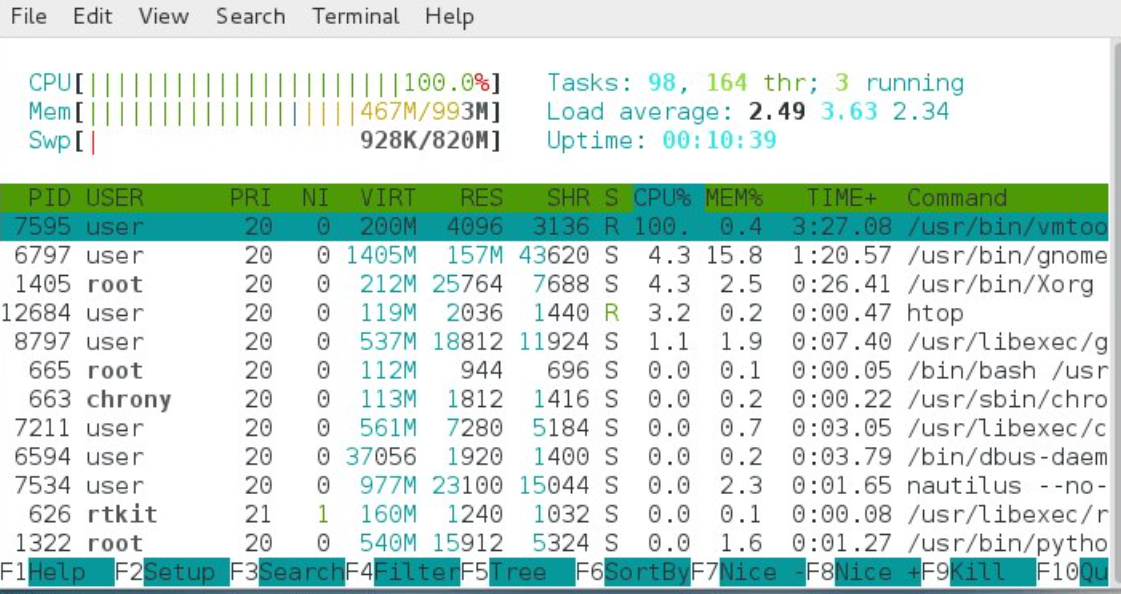
F6 - открывает панель с выбором параметра сортировки процессов;

F7 - увеличить приоритет выполнения текущего процесса;

F8 - уменьшить приоритет выполнения текущего процесса;

F9 - убить процесс;

F10 - выйти из программы.



**Команда ps**

**ps**

Команда ps выдает информацию о состоянии процесса

|  |  |  |
| --- | --- | --- |
| Команда | Ключи | Описание ключей |
| ps | -e | Вывести информацию обо всех процессах. |
| ps | -d | Вывести информацию обо всех процессах, кроме лидеров групп. |
| ps | -a | Вывести информацию обо всех наиболее часто запрашиваемых процессах, то есть обо всех процессах, кроме лидеров групп и процессов, не ассоциированных с терминалом. |
| ps | -f | Генерировать полный листинг. |
| ps | -l | Генерировать листинг в длинном формате |
| ps | -n | файл с системой. |

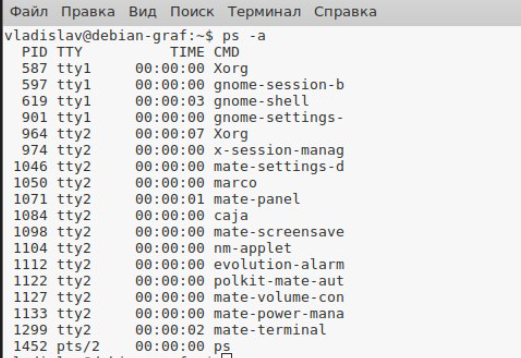
**Пример:**

***Задача:***

Вывести информацию обо всех наиболее часто запрашиваемых процессах.

***Команда:***

|  |
| --- |
| **Ps -a** |



**Команда pstree**

Команда pstree показывает дерево процессов

|  |  |  |
| --- | --- | --- |
| Команда | Ключи | Описание ключей |
| pstree | -a | показыввает команды с аргументами в командной строке |
| pstree | -c | Запрещает компактную форму вывода одинаковых поддеревьев |
| pstree | -G | Использовать для вывода дерева символы соответствующие VT100 |
| pstree | -h | Подсвечивает текущий процесс и его предков |
| pstree | - l | Показывает длинные строки. |
| pstree | -n | Сортирует процессы с одинаковым предком по  идентификатору процесса (PID) вместо сортировки  по имени. (Числовая сортировка.) |
| pstree | **-**p | Показывает идентификаторы процессов PIDs. |

**Примеры:**

***Задача1:***

Вывести дерево идентификаторов процессов PIDs.

***Команда:***

|  |
| --- |
| **Pstree -p** |



**Команда kill**

**Kill**

Команда kill завершает процессы или посылает им сигнал.  
**Ключи**

|  |
| --- |
| -l |
| Выдать все значения сигналов, поддерживаемые в данной реализации, если операнды не указаны. Если указан операнд статус\_выхода, соответствующий значению специального параметра ? командного интерпретатора и вызову wait для процесса, работа которого прекращена сигналом, выдать имя сигнала, которым была прекращена работа процесса. Если указан операнд статус\_выхода, представляющий собой целое число без знака - номер сигнала, - выдать имя соответствующего сигнала. В противном случае, результат не определен. |
| -s |
| Задает сигнал, который надо послать, используя одно из символьных имен, заданных в заголовочном файле <signal.h>. Значение сигнала распознается независимо от регистра символов. При этом префикс SIG указывать не надо. Кроме того, распознается символьное имя 0, представляющее сигнал со значением ноль. Указанный сигнал будет посылаться вместо стандартного, SIGTERM. |

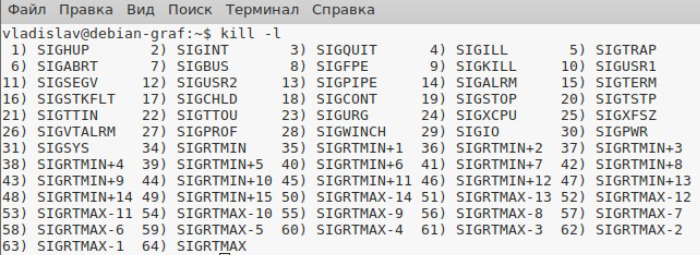
**Пример:**

***Задача:***

Выдать все значения сигналов .

***Команда:***

|  |
| --- |
| **Kill -l** |



**ЗАКЛЮЧЕНИЕ.**

Простой и скромный терминал в Linux является чрезвычайно мощным повседневным рабочим инструментом, а не средством, используемым лишь в крайнем случае. Терминал несложен в изучении, а знать принципы его работы очень полезно.

Проделав данную курсовую работу, мы научились работать с командами ОС Linux, относящимися к так называемому разделу System Information. Теперь мы без труда можем узнать всю необходимую информацию о нашей системе, процессах или файлах.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. [coder-booster.ru/learning/linux-beginners/command-she..](https://vk.com/away.php?to=http%3A%2F%2Fcoder-booster.ru%2Flearning%2Flinux-beginners%2Fcommand-shells-and-bash-interpreter&cc_key=)

[2.](https://vk.com/away.php?to=http%3A%2F%2Fcoder-booster.ru%2Flearning%2Flinux-beginners%2Fcommand-shells-and-bash-interpreter&cc_key=) [docs.altlinux.org/ru-RU/archive/3.0/html-single/compa..](https://vk.com/away.php?to=http%3A%2F%2Fdocs.altlinux.org%2Fru-RU%2Farchive%2F3.0%2Fhtml-single%2Fcompact%2Fcompact3.0%2Facc.kirill.prev%2Findex.html&cc_key=)

[3.](https://vk.com/away.php?to=http%3A%2F%2Fdocs.altlinux.org%2Fru-RU%2Farchive%2F3.0%2Fhtml-single%2Fcompact%2Fcompact3.0%2Facc.kirill.prev%2Findex.html&cc_key=) [help.ubuntu.ru/wiki/sudo](https://vk.com/away.php?to=http%3A%2F%2Fhelp.ubuntu.ru%2Fwiki%2Fsudo&cc_key=)

[4.](https://vk.com/away.php?to=http%3A%2F%2Fhelp.ubuntu.ru%2Fwiki%2Fsudo&cc_key=) [kuzevanov](https://vk.com/away.php?to=http%3A%2F%2Fkuzevanov.ru%2Fosnovnye-komandy-v-linux&cc_key=).ru/osnovnye-komandy-v-linux

[5.](https://vk.com/away.php?to=http%3A%2F%2Fkuzevanov.ru%2Fosnovnye-komandy-v-linux&cc_key=) [linuxgid](https://vk.com/away.php?to=http%3A%2F%2Flinuxgid.ru%2Fosnovnye-komandy-v-konsoli-linux%2F&cc_key=).ru/osnovnye-komandy-v-konsoli-linux/

[6.](https://vk.com/away.php?to=http%3A%2F%2Flinuxgid.ru%2Fosnovnye-komandy-v-konsoli-linux%2F&cc_key=) [pandia.ru/text/78/247/54989.php](https://vk.com/away.php?to=http%3A%2F%2Fpandia.ru%2Ftext%2F78%2F247%2F54989.php&cc_key=)

[7.](https://vk.com/away.php?to=http%3A%2F%2Fpandia.ru%2Ftext%2F78%2F247%2F54989.php&cc_key=) [rus-linux.net/MyLDP/consol/15-primerov-po-useradd.html](https://vk.com/away.php?to=http%3A%2F%2Frus-linux.net%2FMyLDP%2Fconsol%2F15-primerov-po-useradd.html&cc_key=)

[8.](https://vk.com/away.php?to=http%3A%2F%2Frus-linux.net%2FMyLDP%2Fconsol%2F15-primerov-po-useradd.html&cc_key=) [white55.ru/vboxmageia.html](https://vk.com/away.php?to=http%3A%2F%2Fwhite55.ru%2Fvboxmageia.html&cc_key=)

[9.](https://vk.com/away.php?to=http%3A%2F%2Fwhite55.ru%2Fvboxmageia.html&cc_key=) [white55.ru/sudo.html](https://vk.com/away.php?to=http%3A%2F%2Fwhite55.ru%2Fsudo.html&cc_key=)