**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.

**§ 2. 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 CONFIGURETION**

**§1. Useradd**

**Useradd**- Команда useradd используется в дистрибутиве Debian для упрощения создания учетных записей пользователей, а также автоматической проверки их соотвествия политике упомянутого дистрибутива. Она реализована в виде обертки для таких утилит, как useradd, passwd и chfn и использует дополнительный файл конфигурации /etc/adduser.conf для хранения параметров, относящихся к политике дистрибутива Debian.

Команда, введенная без ключей создает нового пользователя без дополнительных действий.

|  |  |  |
| --- | --- | --- |
| **Команда** | **Ключи** | **Описание ключей** |
| **Useradd** | -b | Этот ключ необходим для того, чтобы у пользователя был задан базовый каталог для домашнего |
| -d | Создает полный путь к домашнему каталогу |
| -e | Ключ нужен, чтобы установить дату устаревания |
| -m/M | Указатель на то, что нужно(не нужно) создать домашний каталог |
| -p | Создание пользователя с паролем |
| -c | Создание пользователя с комментарием |
| -u | Присвоение пользователю UID |
| -h | Доступные опции |

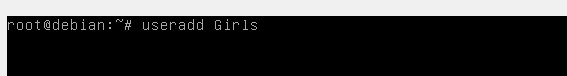
**Примеры:**

***Задача:***

Создать пользователя Girls без дополнительных опций

***Команда:***

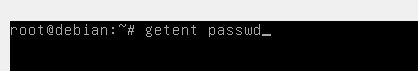
|  |
| --- |
| **useradd Girls** |



|  |
| --- |
| **getent passwd** |

***Проверка:***

С помощью данной команды мы проверяем, создался ли пользователь Girls.



***Результат:***

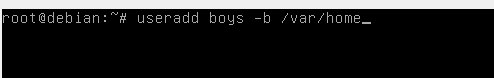


***Задача:***

Создать пользователя boys с заданным базовым каталогом, для домашнего.

***Команда:***

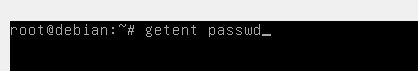
|  |
| --- |
| **useradd boys –b /var/home** |



|  |
| --- |
| **getent passwd** |

***Проверка:***

С помощью данной команды мы проверяем, создался ли пользователь boys.



***Результат:***



***Задача:***

Добавить нового пользователя people с созданием полного пути к домашнему каталогу.

***Команда:***

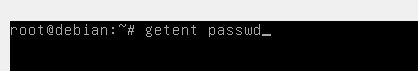
|  |
| --- |
| **useradd people –d /home/kursach** |



***Проверка:***

|  |
| --- |
| **getent passwd** |

C помощью данной команды проверяем, создался ли пользователь people с указанным путем к домашнему каталогу.



***Результат:***



***Задача:***

Создать пользователя cars с датой устаревания.

***Команда:***

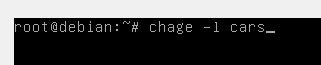
|  |
| --- |
| **useradd cars –e 2018-08-17** |



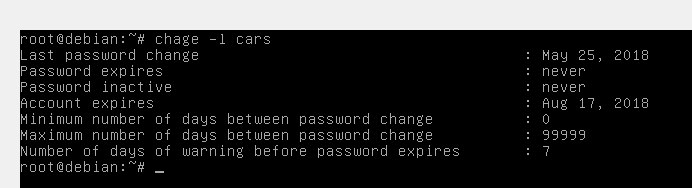
***Проверка:***

|  |
| --- |
| **chage –l cars** |

Данная команда показывает сведения о сроке действия учетной записи.



***Результат:***



***Задача:***

Добавить пользователя dogs и указать, что нужно создать домашний каталог.

***Команда:***

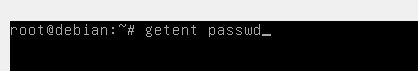
|  |
| --- |
| **useradd dogs -m** |



***Проверка:***

|  |
| --- |
| **getent passwd** |

По умолчанию команда "useradd" создает домашнюю директорию пользователя в /home и называет ее именем пользователя. Поэтому, например, для приведенного выше примера, мы увидим домашнюю директорию созданного нами пользователя "dogs" в "/home/dogs".



***Результат:***



***Задача:***

Добавить пользователя computer и указать, что не нужно создавать домашний каталог.

***Команда:***

|  |
| --- |
| **useradd comp –M computer** |



***Проверка:***

|  |
| --- |
| **ls –l /home/computer** |

В некоторых ситуациях мы не хотим, по соображениям безопасности, давать пользователям домашние директории. В таком случае, когда пользователь авторизуется в системе сразу после ее запуска, его домашней директорией будет root. Если такой пользователь использует команду su, то он авторизуется в домашней директории предыдущего пользователя.



***Результат:***



***Задача:***

Создать пользователя с паролем.

***Команда:***

|  |
| --- |
| **useradd chicken –p 1234** |



***Проверка:***

|  |
| --- |
| **cat /etc/passwd** |

С помощью этой команды также можно проверить, создался ли пользователь в системе.



***Результат:***



***Проверка:***

|  |
| --- |
| **su chicken** |

С помощью этой команды заходим от пользователя chicken



***Результат:***



***Задача:***

Создать пользователя Polya с комментарием.

***Команда:***

|  |
| --- |
| **useradd Polya –c “Polina”** |



***Проверка:***

|  |
| --- |
| **cat /etc/passwd** |

С помощью этой команды проверяем, создался ли пользователь с комментарием



***Результат:***



В Linux каждый пользователь имеет свой собственный UID (Unique Identification Number). По умолчанию при создании нового пользователя ему присваивается userid 500, 501, 502 и т.д. Но можно присвоить свой.

***Задача:***

Создать пользователя Yulia с уникальным UID номером.

***Команда:***

|  |
| --- |
| **useradd Yulia –u 8888** |



***Проверка:***

|  |
| --- |
| **cat /etc/passwd** |

С помощью этой команды проверяем, создался ли пользователь с UID.



***Результат:***



***Задача:***

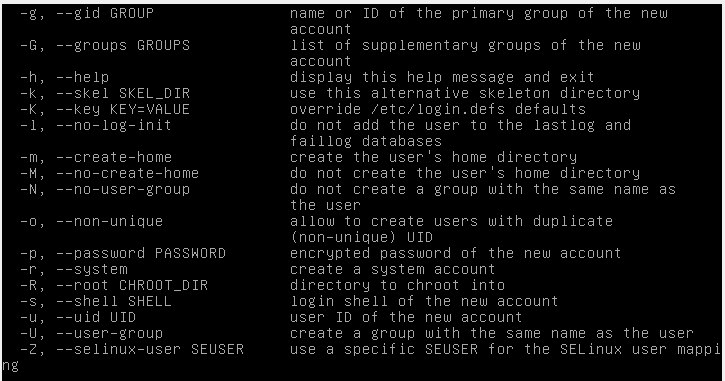
Проверить все возможные опции с помощью ключа –h.

***Команда:***

|  |
| --- |
| **useradd -h** |



***Результат:***



**§2. Groupadd**

***Groupadd*-** команда groupadd создает новую учетную запись группы, используя значения, указанные в командной строке, плюс значения по умолчанию из системы. Новая группа будет добавлена в системные файлы.

Команда, введенная без аргументов, создает новую группу в Linux.

|  |  |  |
| --- | --- | --- |
| **Команда** | **Ключ** | **Описание ключей** |
| **groupadd** | **-g** | С помощью ключа можно присвоить группе GID |
| **-K** | Создание группы пользователей с идентификатором из диапазона |

**Примеры:**

**1.**

***Задача:***

Создать группу students без дополнительных опций.

***Команда:***

|  |
| --- |
| **groupadd students** |



***Проверка:***

|  |
| --- |
| **cat /etc/group** |

С помощью данной команды проверяем, создалась ли группа.



***Результат:***



**2.**

***Задача:***

Создать группу bbso с уникальным GID.

***Команда:***

|  |
| --- |
| **groupadd –g 4444 bbso** |



***Проверка:***

|  |
| --- |
| **cat /etc/group** |

С помощью данной команды проверяем, создалась ли группа с индивидуальным GID (Group Identification Number).



***Результат:***



**3.**

***Задача:***

Создать группу flowers с идентификатором из заданного диапазона.

***Команда:***

|  |
| --- |
| **groupadd –K GID\_MIN=2031 –K GID\_MAX=2052 flowers** |



***Проверка:***

|  |
| --- |
| **cat /etc/group** |

С помощью команды проверяем, действительно ли группе flowers присвоился идентификатор от 2031 до 2052. Первый свободный.



***Результат:***



**§3.Usermod**

***Usermod(groupmod)-***для изменения учетной записи пользователя (account) используется команда usermod. Изменять можно любые атрибуты, но имя пользователя и код UID изменять нужно лишь в случае крайней необходимости, поскольку такое изменение может иметь общесистемные последствия.

Команда usermod, введенная без ключей, изменяет учётную запись пользователя.

|  |  |  |
| --- | --- | --- |
| **Команда** | **Ключи** | **Описание ключей** |
| **usermod** | **-l** | Смена имени пользователя |
| -a -G | Добавление пользователя в группу |
| -n | Смена названия группы |
| -p -l | Смена пароля при каждом входе |

**Примеры:**

**1.**

***Задача:***

Сменить имя пользователя girls на sweet.

***Команда:***

|  |
| --- |
| **usermod –l sweet girls** |



***Проверка:***

|  |
| --- |
| **vi /etc/group** |

Проверяем, действительно ли поменялось имя.



***Результат:***



**2.**

***Задача:***

Добавить пользователя Yulia в группу bbso.

***Команда:***

|  |
| --- |
| **usermod –a –G bbso Yulia** |



***Проверка:***

|  |
| --- |
| **cat /etc/group** |

Проверяем, действительно ли теперь Yulia находится в группе bbso.



***Результат:***



**3.**

***Задача:***

Сменить название группы flowers.

***Команда:***

|  |
| --- |
| **groupmod –n roses flowers** |



***Проверка:***

|  |
| --- |
| **cat /etc/group** |



***Изначальная группа:***



***Результат:***



**4.**

***Задача:***

Заставить пользователя менять пароль при каждом входе в систему.

***Команда:***

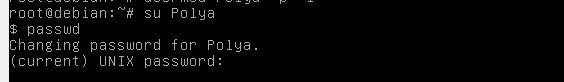
|  |
| --- |
| **usermod Polya –p -l** |



***Проверка и Результат:***

|  |
| --- |
| **Su Polya** |
| **passwd** |

Чтобы проверить, что при каждом входе компьютер будет запрашивать пароль, нужно войти от пользователя Polya и ввести команду passwd.



**§4.Sudo (su)**

***Sudo(su)***- предположим, что нужно выполнить нестандартную задачу, где графические инструменты нам не помогут. К тому же командная строка является очень гибким инструментом, здесь можно увидеть вывод команд и понять, что происходит не так как нужно. А самое главное, команды в терминале являются стандартными для всех дистрибутивов Linux.

Итак, существует два основных способа получить права суперпользователя (Root) это — sudo или su.

Без дополнительных опций команды sudo и su дают право работать от суперпользователя.

|  |  |  |
| --- | --- | --- |
| Команда | Ключ | Описание ключей |
| **Sudo** | -l | Cписок прав пользователя |
| -s | Оболочка SHELL |
| -h | Опции |
| **Su** | -l | Переход в домашнюю директорию |
| -h | Опции |

**Примеры:**

**1.**

***Задача:***

Зайти в систему под именем суперпользователя.

***Команда:***

|  |
| --- |
| **sudo** |



***Результат:***



**2.**

***Задача:***

Показать список прав пользователя или проверить заданную команду.

***Команда:***

|  |
| --- |
| **sudo -l** |

***Результат:***



**3.**

***Задача:***

Запустить оболочку, указанную в переменной окружения SHELL.

***Команда:***

|  |
| --- |
| **sudo -s** |



***Результат:***

exit-выход из сессии



**4.**

***Задача:***

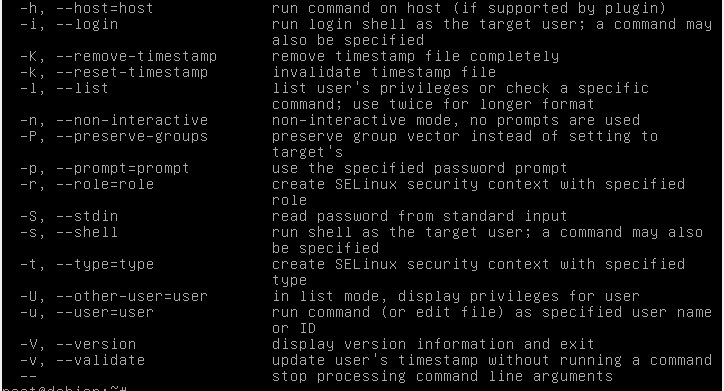
Посмотреть опции sudo.

***Команда:***

|  |
| --- |
| **sudo -h** |



***Результат:***



**5.**

***Задача:***

Зайти под другой учетной записью.

***Команда:***

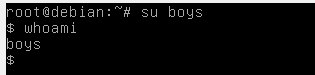
|  |
| --- |
| **su boys** |



***Проверка и Результат:***

|  |
| --- |
| **whoami** |

С помощью данной команды узнаем под каким пользователем мы работаем



**6.**

***Задача:***

Перейти в домашнюю директорию.

***Команда:***

|  |
| --- |
| **su -l** |

***Результат:***



Ничего не поменялось, т.к. пользователь уже находился в домашней директории.

**7.**

***Задача:***

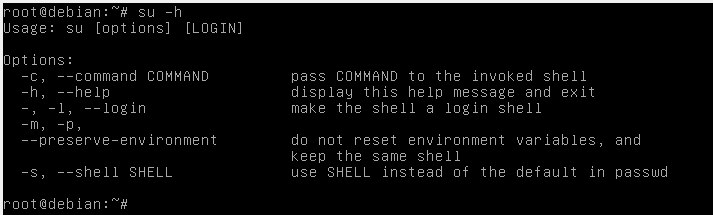
Посмотреть опции su.

***Команда:***

|  |
| --- |
| **su -h** |



***Результат:***



**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. К счастью, ОС Linux является многопользовательской и безопасной, и Вы можете создать учётную запись нового пользователя для улучшения своих практических навыков, а затем экспериментировать без риска что-либо испортить.

Проделав данную курсовую работу, мы научились работать с командами ОС Linux, относящимися к так называемому разделу System Configuration. Теперь мы без труда можем добавить нового пользователя, работать от имени супер пользователя, присвоить пользователю индивидуальный UID, создать группу пользователей, установить для пользователя дату устаревания работы учетной записи и т.п.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. coder-booster.ru/learning/linux-beginners/command-shells-and-bash-interpreter
2. docs.altlinux.org/ru-RU/archive/3.0/html-single/compact/compact3.0/acc.kirill.prev/index.html
3. help.ubuntu.ru/wiki/sudo
4. kuzevanov.ru/osnovnye-komandy-v-linux
5. linuxgid.ru/osnovnye-komandy-v-konsoli-linux/
6. pandia.ru/text/78/247/54989.php
7. rus-linux.net/MyLDP/consol/15-primerov-po-useradd.html
8. white55.ru/vboxmageia.html
9. white55.ru/sudo.html