Лабораторная работа №3

Тема: Командный интерфейс Unix-подобных ОС

Цель: ознакомиться с интерфейсом командной строки GNU/Linux операционных систем, изучить основные команды для настройки и работы с ОС.

Теоретические сведения

Чтобы получить доступ к командной строке необходимо запустить приложение **Терминал (terminal)** при работе с рабочим столом Gnome или **konsole** в KDE.

Когда вы регистрируетесь в системе Linux, выполняется проверка имени пользователя и пароля, определяется набор переменных окружения и запускается **интерпретатор команд**, который чаще называют **оболочкой**. В большинстве дистрибутивов Linux по умолчанию применяются оболочки **sh** или **bash**. Существуют и другие оболочки, однако они менее распространены, но при необходимости можно легко сменить оболочку по умолчанию, или запускать ее из командной строки по мере надобности.

Командная оболочка UNIX (англ. Unix shell, часто просто «шелл» или «sh») — командный интерпретатор, используемый в операционных системах семейства UNIX, в котором пользователь может либо давать команды операционной системе по отдельности, либо запускать скрипты, состоящие из списка команд. В первую очередь, под shell понимаются POSIX-совместимые оболочки, восходящие к Bourne shell (шелл Борна), появившемуся в Unix Version 7.

bash (от англ. Bourne again shell, каламбур «Born again» shell — «Возрождённый» shell) — усовершенствованная и модернизированная вариация командной оболочки Bourne shell. Одна из наиболее популярных современных разновидностей командной оболочки UNIX. Особенно популярна в среде Linux, где она часто используется в качестве предустановленной командной оболочки.

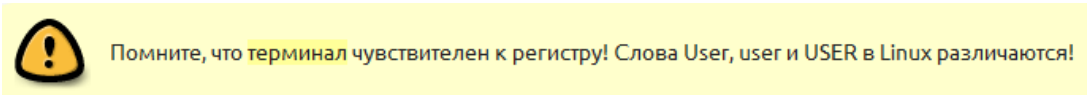
Bash — это акроним от Bourne-again-shell, «ещё-одна-командная-оболочка-Борна». Название — игра слов: Bourne-shell — одна из популярных разновидностей командной оболочки для UNIX (sh), автором которой является Стивен Борн (1978), усовершенствована в 1987 году Брайаном Фоксом.

Команда Linux - это строка символов из **имени команды** и **аргументов**, разделенных пробелами. Аргументы предоставляют команде дополнительные параметры, определяющие ее поведение.

Команды, являющиеся частью оболочки, называются встроенными. Естественно, они могут отличаться для различных оболочек. Но есть и команды, не зависящие от используемой оболочки, и представляющие собой отдельные программные модули, стандартные для выбранного вами дистрибутива Linux.

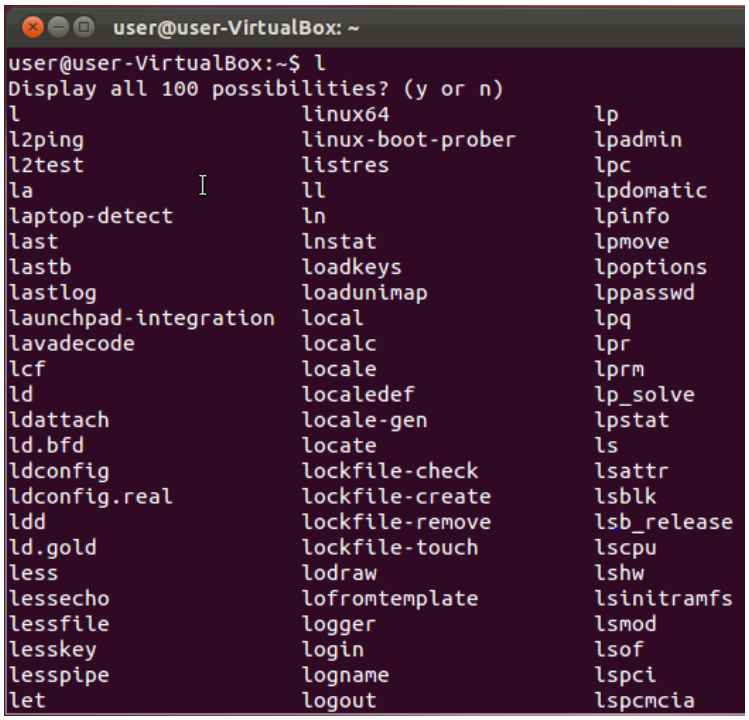
Запустим терминал:

где **user** – это имя пользователя, после символа @ указано название компьютера **user-VirtualBox.** Символ тильда (~) обозначает вашу домашнюю папку. Если вы **user**, то тильда (~) — это /home/**user/ (домашний каталог пользователя)**. Помните, однако, что эту же информацию Gnome Terminal всегда показывает в названии окна. Текущий каталог - это то, что между символами **:** и **$**.



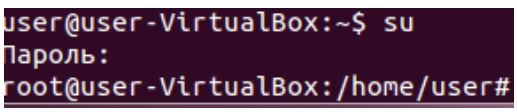
Набирать команды с клавиатуры посимвольно немного неудобно. Для копирования/вставки стандартные сочетания клавиш Ctrl+C и Ctrl+V в терминале не работают, вместо них используется старая добрая пара Ctrl+Insert с Shift+Insert или же сочетания с Shift: Ctrl+Shift+C для копирования и Ctrl+Shift+V для вставки, а также нажатием средней кнопки мыши или правой кнопки мыши и выбором строки «Вставить».

Автодополнение в терминале. Наберите в терминале символ **l**, а потом нажмите клавишу Tab дважды. Терминал автоматически покажет вам команды, которые начинаются с данного символа. Автодополнение в терминале работает практически везде, и не только для команд, но так же для их аргументов и имён файлов.



Для выполнения команд с неограниченными правами необходимо начинать запись команды с ключевого слова **sudo.** Это временно даёт права суперпользователя, которые необходимы для работы с файлами и каталогами, которые не принадлежат вашему аккаунту. Для использования **sudo** требуется ввести пароль. Только пользователи с такими административными привилегиями могут использовать эту команду.

Команда **su** – позволяет пользователю выполнять команды от имени другого пользователя, не завершая текущий сеанс, или получить роль. По умолчанию предполагается работа от имени суперпользователя root. В этом случае после ввода пароля для root строка приглашения будет иметь следующий вид:



Кратко перечислим основные команды (программы) ОС Linux.

1. **Вывод справки по команде (man)**

man (от англ. manual — руководство) — команда Unix, предназначенная для форматирования и вывода справочных страниц.

man <имя изучаемой команды>

man ls - выдаст подробную информацию по команде **ls.**

2. Список работающих процессов (top)

Команда top показывает список работающих в данный момент процессов и информацию о них, включая использование ими памяти и процессора. Список интерактивно формируется в реальном времени.

Чтобы выйти из программы top, нажмите клавишу [q].

3. Количество памяти (free)

free [-b | -k | -m] [-o] [-s delay] [-t] [-V]

Показывает общее количество свободной и используемой физической памяти и памяти отведенной для свопирования в системе, так же и совместно используемую память и буфера используемые ядром.

Опции :

-b показывает количество памяти в байтах; опция -k (по умолчанию) поазывает

количество пвмяти в килобайтах; Опция -m показывает количество памяти в мегабайтах.

-t показывает строки содержащие полное количество памяти.

-o запрещает показывать строки относящиеся к "массиву буфера" . Если не определено

отнять/добавить память буферов из/в используемую/свободнуб память (соответственно!).

-s разрешает безостановочно выводить информацию с промежутком в delay секунд.

-V показывает информацию о версии программы.

Команда **df** показывает объём занятого дискового пространства на всех смонтированных разделах.

Команда **du** отображает объём дискового пространства.

4. Отчет о работающих процессах (ps)

ps [опции]

Команда ps выводит в стандартный вывод информацию о текущем состоянии процессов.

Опции :

-b показывает количество памяти в байтах; опция -k (по умолчанию) поазывает

количество пвмяти в килобайтах; Опция -m показывает количество памяти в мегабайтах.

-t показывает строки содержащие полное количество памяти.

-o запрещает показывать строки относящиеся к "массиву буфера" . Если не определено

отнять/добавить память буферов из/в используемую/свободнуб память (соответственно!).

-s разрешает безостановочно выводить информацию с промежутком в delay секунд.

-V показывает информацию о версии программы.

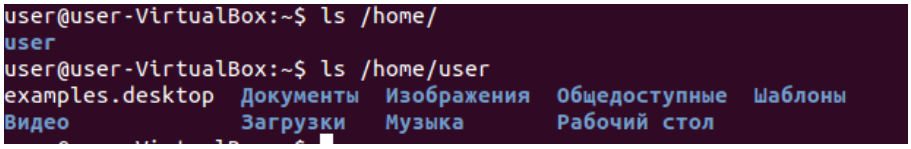
5. Выдача информации о файлах или каталогах (ls)

Синтаксис команды:

ls [флаги] [имя ...]

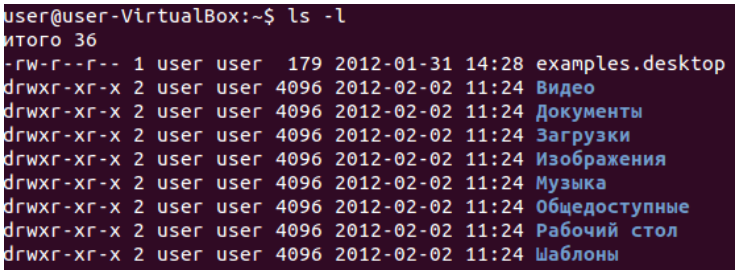
Команда ls для каждого имени каталога распечатывает список входящих в этот каталог файлов; для файлов - повторяется имя файла и выводится дополнительная информация в соответствии с указанными флагами. По умолчанию имена файлов выводятся в алфавитном порядке. Если имена не заданы, выдается содержимое текущего каталога. Если заданы несколько аргументов, то они сортируются по алфавиту, однако сначала всегда идут файлы, а потом каталоги с их содержимым.

Команда **ls [каталог]** – выводит список содержимого каталога.



В качестве аргументов командам передаются ключи или опции, состоящие из тире и одного или нескольких символов.

Например, **ls –l**



При использовании нескольких ключей, их можно объединить. Ниже приведенные варианты команд идентичны:

ls -l -d

ls –ld

Ключ **--help** позволяет получить справку по команде:

Например: **ls –help**

Для постраничного вывода в консоли необходимо использовать команду **less** вместе с основной цепочкой команд, например: **ls –help | less.** Для выхода из постраничного режима нажмите клавишу Q. Также можно перемещаться по терминалу клавишами **Home, End, Page Down, Page Up.** В этом случае для приостановки вывода на экран - **CTRL-S**, отмена приостановки - **CTRL-Q.**

В цепочки можно объединять несколько команд. Например, вам понадобилось вывести на экран в постраничном режиме все строки файла myfile, содержащие слово myname. Для поиска можно воспользоваться командой:

grep <строка поиска>

Итоговая цепочка из трех команд:

cat myfile | grep myname | less

6. Выдача имени текущего каталога (pwd)

Синтаксис команды:

pwd

Бывает, что при ее изучении, вы попадаете в какой-то каталог, про который уже не помните, как он называется и как вы в него попали. Узнать его полное имя позволяет команда pwd.

7. Смена текущего каталога (cd)

Синтаксис команды:

cd [каталог]

Команда cd применяется для того, чтобы сделать заданный каталог текущим. Если каталог не указан, используется значение переменной окружения $HOME (обычно это каталог, в который Вы попадаете сразу после входа в систему). Если каталог задан полным маршрутным именем, он становится текущим. По отношению к новому каталогу нужно иметь право на выполнение, которое в данном случае трактуется как разрешение на поиск.

cd / - выход в корневой каталог

cd ~ - домашний каталог

cd.. – на один каталог вверх.

```
user@user-VirtualBox:~$ cd /home/
user@user-VirtualBox:/home$
```

Вместо символа тильда сейчас /home. Это значит, что мы находимся в данном каталоге.

Необходимо сразу обратить внимание на несколько важных особенностей. Во-первых, при наборе путей так же работает автодополнение по Tab, это очень удобно. Во-вторых, использовать различные небуквенные символы и пробелы напрямую при наборе путей нельзя. Например, для того, чтобы перейти в каталог, содержащий в имени символ пробела, надо при наборе пути к такому каталогу перед пробелом поставить символ обратного слеша \. Установка обратного слеша перед некоторыми символами называется **экранированием.** Кроме того, можно просто заключить путь в двойные кавычки

8. Изменение режима доступа к файлам (chmod)

Синтаксис команды:

chmod режим файл

Права доступа к указанным файлам (среди которых могут быть каталоги) изменяются в соответствии с указанным режимом. Режим может быть задан в абсолютном или символьном виде.

Использование *символьного вида* основано на однобуквенных обозначениях, которые определяют класс доступа и права доступа для членов данного класса. Права доступа к файлу зависят от идентификатора пользователя и идентификатора группы, в которую он входит. Режим в целом описывается в терминах трех последовательностей, по три буквы в каждой:

Владелец Группа Прочие

(u) (g) (o)

rwx rwx rwx

Здесь владелец, члены группы и все прочие пользователи обладают правами чтения файла, записи в него и его выполнения. В примере показаны обозначения как для класса доступа, так и для прав доступа внутри класса.

Для задания **режима доступа** в символьном виде используется следующий синтаксис:

[кому] операция права

Часть [кому] есть комбинация букв u, g и o (владелец, члены группы и прочие пользователи соответственно). Если часть кому опущена или указано a, то это эквивалентно ugo.

Операция может быть: + (добавить право), - (лишить права), = (в пределах данного класса присвоить права абсолютно, то есть добавить указанные права и отнять неуказанные).

Права - любая осмысленная комбинация следующих букв:

r Право на чтение.

w Право на запись.

x Право на выполнение (поиск в каталоге).

s При выполнении переустанавливать действующий идентификатор пользователя или группы.

t После выполнения программы сохранять сегмент команд (бит навязчивости).

l Учет блокировки доступа.

Опустить часть права можно только если операция есть = (для лишения всех прав).

Если надо сделать более одного указания об изменении прав, то при использовании символьного вида в правах не должно быть пробелов, а указания должны разделяться запятыми. Например, команда chmod u+w,go+x f1 добавит для владельца право писать в

файл f1, а для членов группы и прочих пользователей - право выполнять файл. Права устанавливаются в указанном порядке. Право s можно добавлять только для пользователя и группы, право t - только для пользователя.

Чтобы установить права, позволяющие владельцу читать и писать в файл, а членам группы и прочим пользователям только читать, надо использовать следующую запись:

chmod u=rw,go=r f1

Позволить всем выполнять файл f2

chmod +x f2

Для понимания сути задания прав в Unix-системах, нужно знать представление чисел в восьмеричной и двоичной системах счисления.

пользователь	группа	остальные
7	5	5
111	101	101
rwx	r-x	r-x
u	g	o

u – user; g – group; o – other; a - все пользователи вообще (all), то есть a=ugo

400 — владелец имеет право на чтение;

200 — владелец имеет право на запись;

100 — владелец имеет право на выполнение;

40 — группа имеет право на чтение;

20 — группа имеет право на запись;

10 — группа имеет право на выполнение;

4 — остальные имеют право на чтение;

2 — остальные имеют право на запись;

1 — остальные имеют право на выполнение.

chmod 755 имяфайла

755 = 400+200+100+40+10+4+1 #такие права устанавливает запись 755

Каждый пользователь имеет право читать и запускать на выполнение; владелец может редактировать

chmod 444 имяфайла

444=400+40+4 # все имеют право только на чтение

Разберем числовую запись 755.

-**rwxr-xr-x** #эквивалентная символьная запись 755

Первый символ перед правами обозначает тип файла:

- обычный файл

d каталог
b файл блочного устройства
c файл символьного устройства
s доменное гнездо
p именованный канал
l символическая ссылка

Затем идут девять символов. Они делятся на три группы, по три символа.
Значения групп слева направо:
владелец
группа
все остальные пользователи системы
Значения символов в группах (символы тоже идут в определённом порядке):
r чтение (read)
w запись (write)
x выполнение (execute)
- отсутствие права (на месте которого находится символ)

Например:

chmod 755 имяфайла

```
user@user-VirtualBox:~$ chmod 755 script
user@user-VirtualBox:~$ ls -l script
-rwxr-xr-x 1 user user 89 2012-02-09 23:01 script
```

Каждая цифра - это сумма, сложенная из различных привилегий. Вот их значения:
0 - нет прав (---)
1 - только выполнение (--x)
2 - только запись (-w-)
3 - запись и выполнение (-wx)
4 - только чтение (r--)
Для установки прав на чтение и выполнение нужна цифра 5: 1 для выполнения и 4 для чтения. Для полного доступа нужно 4 для чтения, 2 для записи и 1 для исполнения: 4 + 2 + 1 = 7.
Если Вы устанавливаете на файл права 755, это значит, что владелец имеет полный доступ (7), группа имеет право читать и выполнять файл (5). Точно такие же права имеют все остальные пользователи.
3 = wx запись и выполнение
5 = rx чтение и выполнение
6 = rw чтение и запись
7 = rwx чтение, запись и выполнение (полные права)

Например, чтобы установить права, позволяющие владельцу читать и писать в файл, а членам группы и прочим пользователям только читать (2 способа):
chmod 644 filename
chmod u=rw, go=r filename

Позволить всем выполнять файл filename
chmod a+x filename

Установка прав 755 для всех файлов и поддиректорий в домашнем каталоге
chmod -R 755 ~/

9. Копирование файлов (cp)

cp файл1 [файл2 ...] целевой_файл

Команда cp копирует файл1 в целевой_файл. Файл1 не должен совпадать с целевым_файлом (будьте внимательны при использовании метасимволов shell'a). Если целевой_файл является каталогом, то файл1, файл2, ..., копируются в него под своими именами. Только в этом случае можно указывать несколько исходных файлов.

Если целевой_файл существует и не является каталогом, его старое содержимое теряется. Режим, владелец и группа целевого_файла при этом не меняются.

Если целевой_файл не существует или является каталогом, новые файлы создаются с теми же режимами, что и исходные (кроме бита навязчивости, если Вы не суперпользователь).

Время последней модификации целевого_файла (и последнего доступа, если он не существовал), а также время последнего доступа к исходным файлам устанавливается равным времени, когда выполняется копирование. Если целевой_файл был ссылкой на другой файл, все ссылки сохраняются, а содержимое файла изменяется.

10. Перемещение (переименование) файлов (mv)

Синтаксис команды:

mv [-f] файл1 [файл2 ...] целевой_файл

Команда mv перемещает (переименовывает) файл1 в целевой_файл. Файл1 не должен совпадать с целевым_файлом (будьте внимательны при использовании метасимволов shell'a).

Если целевой_файл является каталогом, то файл1, файл2, ..., перемещаются в него под своими именами. Только в этом случае можно указывать несколько исходных файлов.

Если целевой_файл существует и не является каталогом, его старое содержимое теряется.

Если при этом обнаруживается, что в целевой_файл не разрешена запись, то выводится режим этого файла [см. chmod] и запрашивается строка со стандартного ввода. Если эта строка начинается с символа y, то требуемые действия все же выполняются, при условии, что у пользователя достаточно прав для удаления целевого_файла. Если была указана опция –f или стандартный ввод назначен не на терминал, то требуемые действия выполняются без всяких запросов. Вместе с содержимым целевой_файл наследует режим файла1.

Если файл1 является каталогом, то он переименовывается в целевой_файл, только если у этих двух каталогов общий надкаталог; при этом все файлы, находившиеся в файле1, перемещаются под своими именами в целевой_файл. Если файл1 является файлом, а целевой_файл - ссылкой, причем не единственной, на другой файл, то все остальные ссылки сохраняются, а целевой_файл становится новым независимым файлом.

11. Удаление файлов (rm)

Синтаксис команды:

rm [-f] [-i] файл ...

rm -r [-f] [-i] каталог ... [файл ...]

Команда rm служит для удаления указанных имен файлов из каталога. Если заданное имя было последней ссылкой на файл, то файл уничтожается. Для удаления пользователь должен обладать правом записи в каталог; иметь право на чтение или запись файла не обязательно.

Следует заметить, что при удалении файла в Linux, он удаляется навсегда. Здесь нет возможностей вроде "мусорной корзины" в windows 95/98/NT или команды undelete в DOS.

Так что, если файл удален, то он **удален**!

Если нет права на запись в файл и стандартный ввод назначен на терминал, то выдается (в восьмеричном виде) режим доступа к файлу и запрашивается подтверждение; если оно начинается с буквы y, то файл удаляется, иначе - нет. Если стандартный ввод назначен не на терминал, команда rm ведет себя так же, как при наличии опции -f.Допускаются следующие три **опции**:

-f Команда не выдает сообщений, когда удаляемый файл не существует, не запрашивает подтверждения при удалении файлов, на запись в которые нет прав. Если нет права и на запись в каталог, файлы не удаляются. Сообщение об ошибке выдается лишь при попытке удалить каталог, на запись в который нет прав (см. опцию -r).

-r Происходит рекурсивное удаление всех каталогов и подкаталогов, перечисленных в списке аргументов. Сначала каталоги опустошаются, затем удаляются. Подтверждение при удалении файлов, на запись в которые нет прав, не запрашивается, если задана опция -f или стандартный ввод не назначен на терминал и не задана опция -i. При удалении непустых каталогов команда rm -r предпочтительнее команды rmdir, так как последняя способна удалить только пустой каталог. Но команда rm -r может доставить немало острых впечатлений при ошибочном указании каталога!

-i Перед удалением каждого файла запрашивается подтверждение. Опция -i устраняет действие опции -f; она действует даже тогда, когда стандартный ввод не назначен на терминал.

ПРИМЕРЫ

Опция -i часто используется совместно с -r. По команде:

rm -ir dirname

запрашивается подтверждение:

directory dirname: ?

При положительном ответе запрашиваются подтверждения на удаление всех содержащихся в каталоге файлов (для подкаталогов выполняются те же действия), а затем подтверждение на удаление самого каталога.

12. Удаление каталогов (rmdir)

Синтаксис команды:

rmdir [-p] [-s] каталог ...

Команда rmdir удаляет указанные каталоги, которые должны быть пустыми. Для удаления каталога вместе с содержимым следует воспользоваться командой rm с опцией -r. Текущий каталог [см. pwd] не должен принадлежать поддереву иерархии файлов с корнем – удаляемым каталогом.

Для удаления каталогов нужно иметь те же права доступа, что и в случае удаления обычных файлов [см. rm].

Командой rmdir обрабатываются следующие **опции**:

-p Позволяет удалить каталог и вышележащие каталоги, оказавшиеся пустыми. На стандартный вывод выдается сообщение об удалении всех указанных в маршруте каталогов или о сохранении части из них по каким-либо причинам.

-s Подавление сообщения, выдаваемого при действии опции -p.

13. Создание ссылки на файл (ln)

Синтаксис команды:

ln [-f] файл1 [файл2 ...] целевой_файл

Команда ln делает целевой_файл ссылкой на файл1. Файл1 не должен совпадать сцелевым_файлом (будьте внимательны при использовании метасимволов shell'a). Если целевой_файл является каталогом, то в нем создаются ссылки на файл1, файл2, ... с теми же именами. Только в этом случае можно указывать несколько исходных файлов.

Если целевой_файл существует и не является каталогом, его старое содержимое теряется. Если при этом обнаруживается, что в целевой_файл не разрешена запись, то

выводится режим доступа к этому файлу [см. chmod] и запрашивается строка со стандартного ввода.

Если эта строка начинается с символа y, то требуемые действия все же выполняются, при условии что у пользователя достаточно прав для удаления целевого_файла. Если была указана опция -f или стандартный ввод назначен не на терминал, то требуемые действия выполняются без всяких запросов. Целевой_файл наследует режим доступа к файлу1.

Команда ln не создает ссылок между разными файловыми системами, поскольку они (файловые системы) могут добавляться и удаляться.

14. Создание каталога (mkdir)

mkdir [-m режим_доступа] [-p] каталог ...

По команде mkdir создается дин или несколько каталогов с режимом доступа 0777 [возможно измененном с учетом umask и опции -m]. Стандартные файлы (. - для самого каталога и .. - для вышележащего) создаются автоматически; их нельзя создать по имени. Для создания каталога необходимо располагать правом записи в вышележащий каталог.

Идентификаторы владельца и группы новых каталогов устанавливаются соответственно равными реальным идентификаторам владельца и группы процесса.

Командой mkdir обрабатываются две **опции**:

-m режим_доступа - (явное задание режима_доступа для создаваемых каталогов [см.chmod]).

-p (при указании этой опции перед созданием нового каталога предварительно создаются все несуществующие вышележащие каталоги).

15. Вывод аргументов в стандартный поток вывода(echo)

echo [опции] [string ...]

-n не выводить завершающий символ новой строки.

-e раздрешить интерпретацию следующих backslashescaped последовательностей в строках:

\a alert (звонок)

\b backspace

\c запретить завершающий символ новой строки

\f перегон страницы

\n новая строка

\r перевод строки

\t горизонтальная табуляция

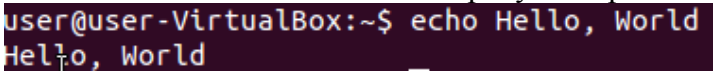
\v вертикальная табуляция \\ обратный слэш

Команда echo предназначена для выдачи на стандартный вывод строки символов, которая задана ей в качестве аргумента.

Передаваемая строка может быть перенеправлена в файл myfile с использованием оператора перенаправления вывода **>** . Например:

$echo "Hello, world!" > myfile

Командой **echo** можно вывести строку на экран:



здесь строка **Hello, World** передана в качестве аргумента.

16. Создание (просмотр) файла (cat)

cat имя_файла - используется для вывода содержимого файла (просмотра),

cat >> имя_файла (при завершении нажать ctrl+d) - для создания файла

17. Другие команды

Команда **uname** выводит системную информацию.

Команда **lsb_release -a** выводит информацию о версии Linux

Команда **ifconfig** выводит информацию об сетевых интерфейсах.

Команда **adduser** имя пользователя - добавляет нового пользователя.

Команда **passwd** пароль – назначает пароль

Команда **history** показывает историю введенных команд, а повторить введенную команду можно написав команду **!**26, где 26 – это порядковый номер команды в истории.

Полезно! Терминал запоминает введенные вами команды. Для перемещения по истории команд используйте клавиши управления курсором. Для поиска по вводимым ранее командам нажмите ctrl+r.

Команда **which** команда позволяет найти основной исполняемый файл команды.

Команда **reset** очищает терминал

Команда **ps** служит для работы с процессами

Команда **kill** - отправляет сигнал прекращения работы тому или иному процессу, используется для прекращения работы зависшего приложения

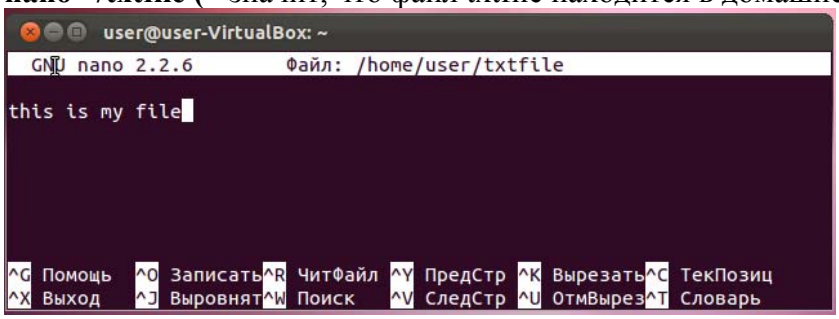
Команда **killall** - прекращение работы всех процессов, соответствующих заданным параметрам

Команды для редактирования файлов, используя текстовые редакторы:

Для редактирования файла можно использовать встроенный консольный текстовый редактор. Для этого необходимо набрать следующую команду:

nano путь к файлу

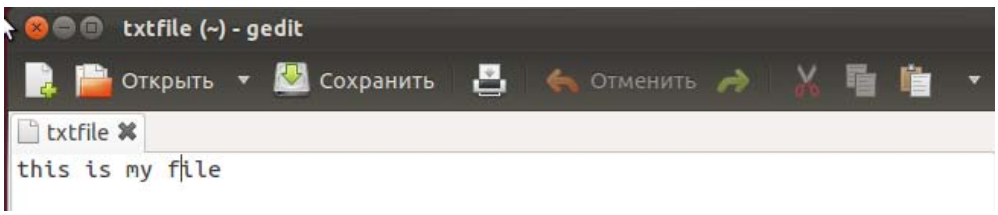
nano ~/txtfile (~ значит, что файл txtfile находится в домашней директории**)**



Команды стоит использовать с клавишей ctrl+

Для выхода, например, нужно нажать ctrl+x. Для получения справки по редактору **nano нажмите ctrl+g.**

Для редактирования файла можно использовать графический текстовый редактор **gedit.** Наберите в консоли**: gedit ~/txtfile**



О tar и gzip

Формат tar используется для создания архива, то есть для хранения нескольких файлов в одном файле. Важно знать, что сам по себе tar не сжимает файлы, а просто

создает архив, который по размеру схож с общим суммарным размером упакованных в него файлов.

Gzip (расшифровывается как GNU zip) является утилитой для сжатия файлов. Интересно то, что gzip не может создать архив - утилита только сжимает один файл до меньших размеров.

Тут стоит вопрос: как же создать сжатый архив в linux? Ответ прост - использовать и tar и gzip одновременно. Смысл в том, что с начала создается архив с файлами при помощи утилиты tar, а потом этот архив сжимается утилитой gzip. В результате всех этих манипуляций мы должны получить файл с расширением tar.gz

О bzip2

bzip2 - это альтернатива gzip, обладающая лучшими показателями сжатия и скорости. В результате мы получим архив с расширением tar.bz2

Общий вид вызова команды tar

tar -cf имя_будущего_архива файл(ы)_для_упаковки

Опции -c и -f означают, соответственно --create и --file, т.е создать файл.

Пример:

tar -cf arhiv txtfile

эта команда создаст архив из одного файла txtfile

tar -cf arhiv *

эта команда упакует в архив без сжатия все файлы и каталоги текущего каталога

Использование gzip для создания архива с сжатием:

tar -zcvf arhiv.tar.gz *

Использование bzip2 для создания архива с сжатием:

tar -cvjf arhiv.tar.bz2 *

Также для bzip2 подойдет и такая команда:

bzip2 имяфайла для упаковки

bzip2 -d имяфайла.bz2 – для распаковки

Распаковка gzip:

tar -zxvf путь_к_архиву

Распаковка bzip2:

tar -jxvf путь_к_архиву

Современные версии GNU tar имеют функцию автоматического распознавания типа архива, и поэтому можно написать:

tar -xvf имяархива.tar.bz2

Получение информации о файлах в архиве

tar -svtf путь_к_архиву для gzip

tar -jvtf путь_к_архиву для bzip2

Задание1. Работа с командами**:**

1.Ознакомиться с командами Linux. Выполнить команды top, free, ps с различными опциями.

2. Войти в свой домашний каталог. Для этого нужно сделать команду **cd ~**

Вы находитесь в своем рабочем каталоге. Здесь хранятся ваши пользовательские файлы и настройки программ, которые вы используете.

3. Создать следующую структуру каталогов и файлов:

1) в домашнем каталоге создать каталог **inform**

2) Перейти в каталог и inform создать в нем каталог **lab1**

3) Внутри каталога lab1 создать каталог **catalog1, файл file1**(например, используя команду echo), каталог **catalog2**. Перейти в каталог catalog2.

4) Внутри каталога catalog2 создать файлы **file3 и file4** , каталог **catalog3**

5) Внутри каталога catalog3 создать файл **file5**, жесткую ссылку **f1** на файл file1, жесткую ссылку **ct2** на каталог catalog2.

6) Создать в каталоге lab1 символичесткую ссылку **s_link** на файл file5

4. Запустить программу MC (Midnight Commander): mc

Здесь вы можете посмотреть структуру созданных вами каталогов и просмотреть содержимое файлов.

Задание 2. Работа с командами по вариантам:

1. В домашней директории создать папку и в ней 6 файлов, каждый из которых будет содержать название предмета, тему лабораторной работы, вариант, группу, номер подгруппы, вариант соответственно. Склеить все файлы в один, оставшиеся файлы удалить. Вывести подробности о каждом файле, включая права доступа, владельца, последние изменения, размер файла. Также необходимо вывести содержимое каталога до и после удаления. Установите следующие права на файлы: владелец может редактировать, остальные только читать
2. В домашней директории создайте следующие каталоги /БарГУ/Инженерный. Далее для каждой специальности инженерного факультета создайте отдельные каталоги (все каталоги должны быть созданы одновременно с помощью одной команды). В каталоге /АТП создайте файл с фамилиями студентов вашей подгруппы. Выведите информацию о файле, а также дерево созданных каталогов. Удалите каталог /БарГУ вместе с вложенными файлами и каталогами. Установите права доступа к файлу: 444
3. В домашней директории создайте файл с произвольным содержимым, выведите подробную информацию о нем, отредактируйте файл с помощью консольного текстового редактора, выведите на экран его содержимое. Создайте копию файла, а оригинал удалите. Установите права доступа к файлу: каждый пользователь может читать, владелец имеет право редактировать и запускать на выполнение.
4. Перенаправьте сообщение с экрана в файл с именем «Строка с терминала». Далее переместите файл в другой каталог и переименуйте его. Выведите подробную информацию о каталоге назначения в файл. Установите права доступа к файлу в символьном виде: владелец может редактировать, остальные только читать.
5. Создайте в домашней директории пустой файл. Откройте его в консольном текстовом редакторе, отредактируйте, сохраните и перенаправьте содержимое данного файла в другой файл. Оригинальный файл удалите. Установите права доступа к файлу в символьном виде: владелец может редактировать и выполнять, группа – читать и выполнять, остальные только читать.
6. Создайте файл и осуществите поиск строки в файле. Права доступа 666 – в символьном виде.
7. Создайте 2 каталога. Один будет содержать файлы *.txt, а другой файлы *.doc. Теперь из общего каталога переместите документы в соответствующие каталоги и выведите информацию о них. Права доступа 755 – в символьном виде.

8. Выполните в терминале любые 10 команд на свой выбор. Напишите команду для повторения 8 команды. Историю ввода команд сохраните в отдельном файле. И выведите на экран его содержимое. Права доступа 644 – в символьном виде.
9. Создайте файл и отредактируйте его в графическом текстовом редакторе. Также произведите редактирование с помощью консольного текстового редактора.
10. Создайте 2-новых пользователей и назначьте им пароли. Потом удалите созданных пользователей. Права доступа 660 – в символьном виде.
11. Выведите подробную информацию о процессах в файл, о системе, откройте несколько приложений через терминал, а потом завершите их процессы. Владелец файл имеет право чтения; никто другой не имеет права выполнять никакие действия.
12. Выведите информацию о сетевых интерфейсах в файл net-config, информацию об устройствах в файл hardware_info. Поменяйте содержимое файлов местами, а потом выведите содержимое файлов на экран. Владелец файла может читать, записывать и запускать на выполнение; никто другой не имеет права выполнять никакие действия.

Задание 3. Используя команды для работы с архивами создайте/распакуйте несколько архивов. Также научитесь работать с архивами, используя графический интерфейс.

Контрольные вопросы

1. Для чего предназначена программа terminal?
2. Что такое оболочка в Linux?
3. Что такое команда в Linux? Какие команды являются встроенными?
4. Расскажите про возможности терминала в Linux.
5. Расскажите про строку приглашения в терминале.
6. Для чего предназначено автодополнение в терминале?
7. Расскажите о командах sudo и su.
8. Для чего предназначена команда ls?
9. Расскажите об основных командах для работы с файлами.
10. Расскажите об основных командах для работы с архивами.
11. Расскажите об основных командах для работы с пакетами.
12. Какая команда используется для постраничного вывода?
13. Какая команда позволяет управлять доступом?
14. Расскажите о командах для работы с процессами в Linux.
15. Расскажите о командах для получения системной информации
16. Расскажите про способы работы с файлами в текстовых редакторах.
17. Для каких групп пользователей можно устанавливать права доступа?
18. В каком виде можно устанавливать права доступа?
19. Какие операции используются при задании прав доступа?