

Основы работы в ОС UNIX

Цель работы:

- Освоить основные принципы работы в командной строке UNIX;
- Освоить основные принципы создания пользователей, групп, делегирования полномочий на выполнение команд;
- Освоить принципы работы с правами доступа в UNIX подобных системах.
- Освоить принцип работы с командным текстовым редактором vi.

1 Теоретические основы

1.1 Аутентификация пользователей

В UNIX пользователи должны себя идентифицировать при входе в систему. Идентификация состоит из двух шагов: ввода имени (по имени система Вас идентифицирует) и ввода входного пароля.

В UNIX-системах имя и начальный пароль Вам присвоит системный администратор при Вашей регистрации в системе (при заведении в систему нового пользователя).

Системный администратор (root) - это особый пользователь, который организывает работу системы: он имеет полный доступ к любому файлу системы, регистрирует новых пользователей и имеет право выполнять специальные программы (например, создания файловой системы).

Чтобы зайти в режим командой строки как пользователь root для старых версий Linux нужно использовать команду `su -`, впринице сама команду `su` подразумевает ввод далее логина пользователя, то есть аналогом `su -` является команда `su root`. По своей сути команда `su -` просто удобное сокращение.

В новых версиях UNIX, не только в Linux, но например и в Mac OS X появилась возможность использования механизма делегирования полномочий пользователям разного уровня посредством программы `sudo`. С помощью нее пользователь временно может повысить свои права до уровня root с ведением протокола событий. То есть Программа `sudo` ведет лог файл, кто из пользователей и когда повышал свои полномочия. Мало кто знает, но с помощью `sudo` можно для некоторых учётных записей настроить не административные привилегии, а просто расширенные. Например позволить им запускать выполнять ряд административных утилит, но не разрешать выполнять их все, а так же пользоваться полным доступом к файловой системе, как это происходит у root.

1.2 Подробно о файловой системе

Как уже говорилось выше корневая файловая система обозначается как `/`. Именно на ней лежат все системные каталоги и файлы и именно к ней мы сможем монтировать наши локальные диски нашего компьютера, а так же флэш-диски, оптические диски, флоппи диски. Более подробно разумеется описано в книгах, ну хотя бы в книге «Linux для Чайников», тут же приведены только самые краткие сведения, которые позволят вам немного начать ориентироваться в Linux.

Основные каталоги корневой файловой системы, рисунок 1 :

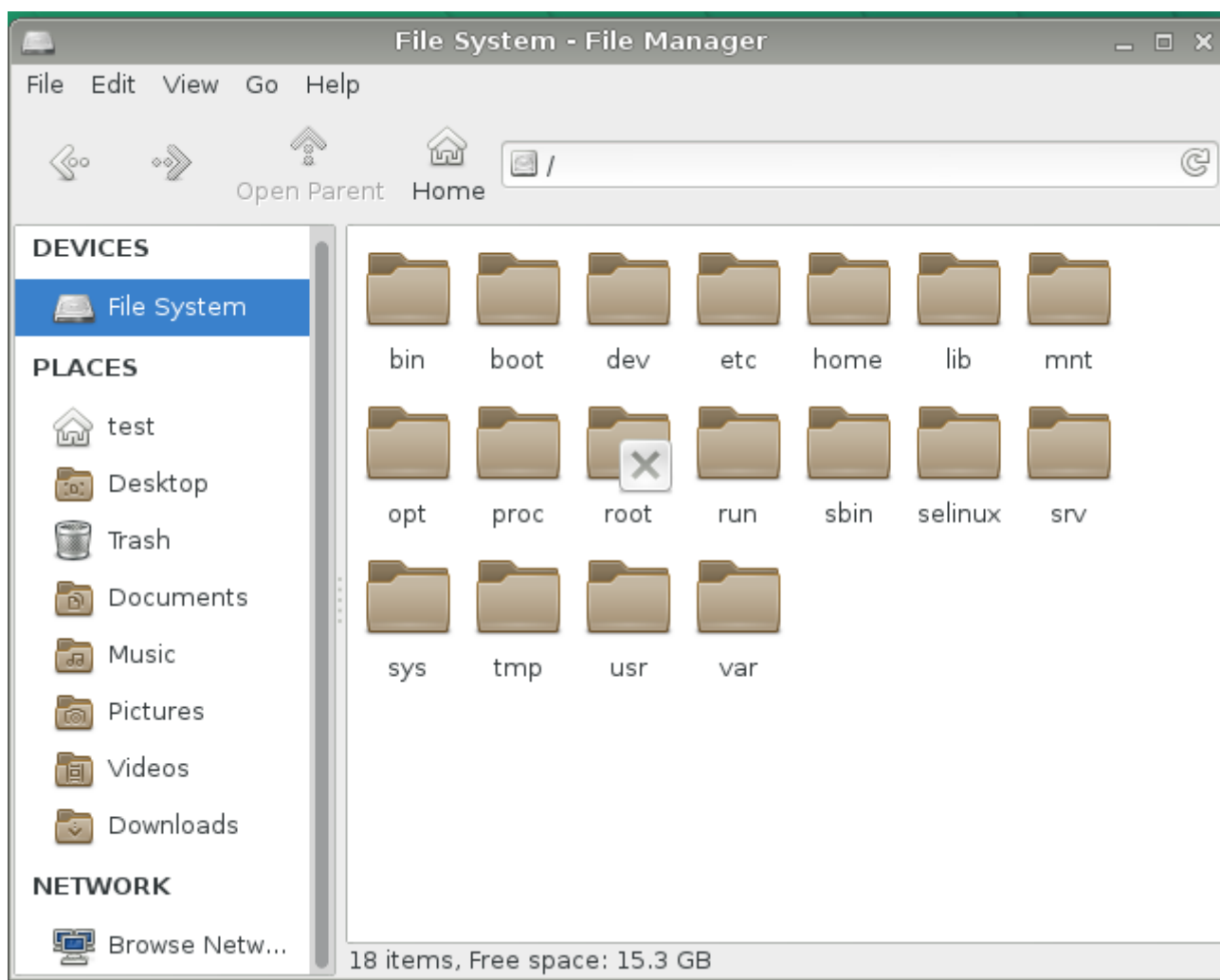


Рисунок 1.

- / - Корневой каталог
- /bin** - Содержит стандартные утилиты Linux
- /boot** - Содержит конфигурационные файлы загрузчика GRUB , образы ядра, файлы Initrd
- /dev** - Содержит файлы устройств. О том, что это такое, мы поговорим чуть позже
- /etc** - Содержит конфигурационные файлы операционной системы и всех сетевых служб. Данный каталог подобен реестру Windows, но в Windows общесистемные настройки хранятся в одном большом бинарном файле, а в Linux настройки хранятся в разных конфигурационных файлах, которые можно редактировать обычным текстовым редактором
- /home** - Содержит домашние каталоги пользователей. В домашних каталогах пользователей хранятся пользовательские файлы, а также пользовательские настройки различных программ
- /lib** - Здесь находятся различные библиотеки и модули ядра
- /misc** - В данном каталоге может быть все, что угодно
- /mnt** - Обычно в этом каталоге содержатся точки монтирования. О монтировании файловых систем мы тоже поговорим отдельно
- /proc** - Это не совсем обычный каталог, это каталог псевдофайловой системы procfs, которая используется для предоставления информации о процессах
- /root** - Каталог пользователя root (пользователь с максимальными полномочиями)
- /sbin** - Набор утилит для системного администрирования, запускать эти утилиты имеет право только root

/tmp - “Мусорка”, т.е. каталог, в котором хранятся временные файлы. Linux, в отличие от Windows, следит за чистотой и регулярно очищает этот каталог

/usr - Содержит пользовательские программы. По размеру это один из самых больших каталогов файловой системы. В этот каталог устанавливаются практически все программы. Также в этом каталоге находятся вспомогательные файлы, необходимые для работы установленных программ. Грубо, конечно, но данный каталог можно сравнить с каталогом Program Files в Windows

/var - Данные системы, которые постоянно изменяются, например, очередь печати, почтовые ящики и т.д.

Так же не стоит забывать, что большинство устанавливаемых прикладных программ в Linux устанавливают свои бинарные файлы, а так же настройки и файлы данных внутри каталогов /usr.

Например: бинарники в /usr/bin или /usr/sbin, файлы данных в /usr/share/папка_с_именем_программы и т. п.

1.3 Синтаксиса командной строки

command option(s) argument(s) ,где
command – это имя программы, которую вы хотите выполнить;
option – ключи, которые определяют поведение программы;
argument – данные, которые эта команда обрабатывает.

В командной строке, которая включает ключи и/или аргументы, все компоненты отделяется друг от друга по крайней мере одним пробелом. Если аргумент содержит пробел, его надо заключить в двойные кавычки. Например, если аргумент sample 1, то вы должны указать его в командной строке как "sample 1". Если кавычки отсутствуют, то shell будет интерпретировать sample и 1 как два отдельных аргумента.

Некоторые команды позволяют вам указывать несколько и/или аргументов в одной командной строке, рисунок 2:

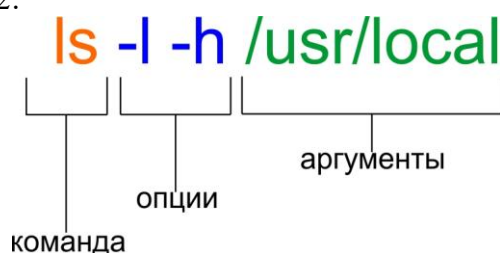


Рисунок 2.

В этом примере команда **ls** использует два. ключа **-l** и **-h** и аргумент в виде пути к каталогу, чье содержимое она должна отобразить.

1.4 Работа с архивами в Linux

Упаковщик **tar** - наиболее распространенный архиватор, используемый в Linux-системах. Сам по себе **tar** не является архиватором, как например Winrar, т.к. он не сжимает файлы, а упаковывает их для последующего сжатия. Разумеется, при упаковке файлов и папок архиватором **tar** сохраняются все атрибуты и права доступа. В то же время, архиваторы **gzip** или **bzip2**, применяющиеся в Linux для сжатия файлов, не умеют сжимать несколько файлов, а работают только с одним файлом или входным потоком. То есть с помощью архиватора **gzip** можно сжать только файл, если же нам понадобится создать архив из нескольких файлов или провести архивацию папки, нам нужно будет сначала упаковать их архиватором **tar** и только после этого провести сжатие программой **gzip**.

Поэтому не стоит удивляться тому, что архивы в *nix системах обычно имеют вид

имя_архива.tar.gz

Упаковщик tar запускается с обязательным указанием одного из основных действий, самые распространенные из которых - создание и распаковка архивов, далее задаются прочие параметры, зависящие от конкретной операции.

Создание архива

Для создания архива нужно указать tar соответствующее действие, что делается с помощью ключа -с. Во всех примерах приведенных ниже будет использоваться так же ключ f, он нужен для того чтобы предотвратить вывод на экран служебных сообщений при упаковке файлов. Далее укажем сначала имя будущего архива, а затем те файлы, которые хотим упаковать.

tar -cf txt.tar *.txt

Здесь команда упакует все файлы находящиеся в текущем каталоге с расширением .txt в архив txt.tar. Разумеется данный архив создан без сжатия. Для сжатия полученного архива можно использовать два метода:

Метод 1:

Можно сразу на этапе упаковки файлов указать tar, каким архиватором следует сжать полученный архив. Для двух самых популярных архиваторов gzip и bzip2 ключи будут -z и -j соответственно.

Пример для сжатия с помощью gzip:

tar -cvzf files.tar.gz ~/files

Пример для сжатия с помощью bzip2:

tar -cvjf files.tar.bz2 ~/files

Метод 2:

Заключается в том, что сначала надо упаковать файлы архиватором tar, а затем сжать архиватором gzip или bzip2:

Пример:

tar -cvf archive.tar *.txt

gzip archive.tar

или

bzip2 archive.tar

Ключ -v включает вывод списка упакованных файлов в процессе работы. К сожалению, более продвинутой индикации хода процесса (например, указания степени готовности в процентах) в tar не предусмотрено.

Распаковка архива

Действие «распаковка» задается с помощью ключа -x. И тут снова потребуется ключ -f для указания имени файла архива. Также добавим ключ -v для визуального отображения хода процесса.

tar -xvf /path/to/archive.tar.bz2

распакует содержимое архива в текущую папку. Альтернативное место для распаковки можно указать с помощью ключа -C (папка указанная после этого ключа уже должна существовать):

tar -xvf archive.tar.lzma -C /path/to/folder

Если имеется файл не просто упакованный архиватором tar, но и сжатый например gzip, то надо сначала разархивировать его с помощью утилиты gunzip, а затем уже распаковать с помощью tar.

Пример:

Имеется файл – archive.tar.gz

Распаковка состоит из двух этапов:

Этап 1:

gunzip archive.tar.gz

Этап 2:

tar -xvf archive.tar.gz

Просмотр содержимого архива

Для просмотра содержимого архива используйте следующую команду:

tar -tf archive.tar.gz

Она выведет простой список файлов и каталогов в архиве. Если же добавить ключ -v, будет выведен подробный список с указанием размера, прав доступа и прочих параметров (так же, как по ls -l)

2 Практическая часть

Задание 2.1 Работа с базовыми командами

1) Введите команду (не забывайте после набора нажимать клавишу ENTER, символ # будет обозначать приглашение командной строки от имени root).

date

Система выведет на экран информацию о дне недели, месяце и числе, времени и годе. На следующей строке будет выведено новое приглашение.

2) Введите команду

who

На экране появится информация, сообщающая, кто работает в данный момент в системе. В первом столбце указывается имя пользователя, во втором – системное имя используемого устройства связи, затем – дата и время входа в систему.

4) Проверьте ваши идентификационные данные. Для чего введите команду:

id

Система выведет параметры Вашей учетной записи. Более подробно о параметрах учетной записи поговорим далее.

5) Вывод строки на экран. Введите команду

echo Lanch is at 12-00

На экран выведется содержимое строки после команды **echo**.

6) Команда вызова справки man:

man ls

На экран выведется справка по команде **ls**

7) Очистка экрана, команда

Clear

Задание 2.2 Файлы и директории

1) Вывод текущего каталога. Введите команду

pwd

Команда `pwd` выводит имя текущего каталога вашей оболочки.

2) Просмотр содержимого каталога (каталога). Введите команду

ls

Система выведет список имен файлов и подкаталогов текущего каталога.

Опции команды **ls**:

ls -l

на экран будет выведен список файлов с дополнительной информацией о типе файла (обычный, каталог или специальный), правах на его использование, владельце, размере, дате последнего изменения (или создания) и др. Можно использовать несколько опций одновременно.

ls -al

добавлена опция **-a**. Система добавит к списку скрытые файлы.

Как видно опции можно добавлять слитно, то есть вывод команды **ls -la** будет аналогичен выводу команды **ls -l -a**.

3) Перемещение по файловой системе

Смена текущего каталога (команда `cd`).

Полное имя задает направление, которое начинается в каталоге `root` и показывает путь по уникальной последовательности каталогов к конкретному каталогу или файлу. Вы можете использовать полное имя для обращения к любому файлу или каталогу в системе UNIX.

Одна точка (`.`) означает текущий каталог, две точки (`..`) - каталог, непосредственно включающий в себя текущий каталог. Он называется родительским для каталога текущего каталога (parent).

Введите команду

cd /

Система сменит текущий каталог на указанный в качестве параметра (в данном случае корневой каталог системы). Для того, чтобы проверить это введите команду **ls**. Изменился ли список файлов?

Перейти на каталог выше можно командой:

cd ..

Перейти в домашний каталог можно командой:

cd ~/

Чтобы узнать точное местоположение используется команда **pwd**

4) Просмотр содержимого домашнего каталога пользователя `test` с загрузками `/home/test/Downloads`.

ls -l /home/test/Downloads или **ls -l /home/test/Загрузки**

на экране выведена информация о директории `etc` в расширенном формате

ls -l

На экране появится нечто подобное следующему, рисунок 4:

```
Terminal - mc [root@linux-70mv]:/home/test/Downloads
File Edit View Terminal Tabs Help
linux-70mv:/home/test/Downloads # ls -l
total 31208
-rw-r--r-- 1 test users 2908451 Mar 10 2017 cacti-0.8.8c.tar.gz
-rw-r--r-- 1 test users 7259630 Mar 8 2017 cacti-1.0.4.tar.gz
-rw-r--r-- 1 test users 10395934 Mar 10 2017 phpMyAdmin-4.6.6-all-languages.tar.gz
-rw-r--r-- 1 test users 11349727 Mar 10 2017 phpMyAdmin-4.6.6-all-languages.zip
-rw-r--r-- 1 test users 2648 Mar 24 2017 statuswrl (1).cgi
-rw-r--r-- 1 test users 2648 Mar 23 2017 statuswrl.cgi
drwxr-xr-x 2 root root 4096 Sep 29 09:39 test_folder
linux-70mv:/home/test/Downloads #
```

Рисунок 4.

Выведенные строки дают представление о каталогах и файлах в каталоге startship. Первый символ в каждой строке (d, -, l, b или c) говорит о типе файла:

- d каталог
- обыкновенный файл
- l символическая связь(канал)
- b специальный блочный файл
- c специальный символьный файл

Таким образом, на экране вы можете видеть, что каталог содержит три каталога и два обычных файла.

Следующие несколько символов, которые являются либо буквами, либо дефисами, идентифицируют право на чтение и использование файла или каталога.

Далее следует цифра - счетчик связей. Для каталога этот счетчик показывает число каталогов, расположенных под ним, плюс два (для самого каталога и каталога-родителя).

Следующим является регистрационное имя владельца файла (в данном случае startship), и за ним - групповое имя файла или каталога (project).

Следующее число показывает длину файла или каталога в байтах. Месяц, день и время последней модификации файла - в предпоследней колонке. В последней колонке представлено имя каталога или файла.

5) Создание директории

Зайдите в домашнюю директорию пользователя (подразумевается, что вы работаете под пользователем test, то есть путь до его домашнего каталога будет /home/test), все дальнейшие действия будут производиться в ней.

mkdir dir_name

Создастся директория с именем dir_name

Создание нескольких директорий:

mkdir -p dir1/dir2/dir3

6) Удаление директории

rmdir dir_name

Удаление нескольких директорий:

mkdir -p dir1/dir2/dir3

Задание 2.3 Операции с файлами

1) Создание каталога. Введите команду

mkdir draft

Создастся каталог с именем draft в текущем каталоге. Сделайте его текущим (команда cd)

Создайте несколько вложенных каталогов:

mkdir dir1/

mkdir dir1/dir2/

mkdir dir1/dir2/dir3

2) Удаление каталогов

rm -r dir1/dir2

Сколько каталогов было удалено?

3) Копирование папки

Для копирования директории необходимо указывать опцию -R

ср -R /home/test/folder1 /home/user2/folder1

4) Копирование файлов

Команда ср полностью копирует содержимое одного файла в другой. Синтаксис команды:

ср ключи имя-файла1 имя-файла2

ср ключи имя-файла(ов) каталог

Чтобы скопировать файл .profile в файл new.profile в каталоге draft, введите команду

ср /home/test/.profile ./new.profile

Чтобы проверить существование нового файла, введите команду:

ls

5) Создание пустого текстового файла

Для создания пустого текстового файлы можно использовать консольный текстовый редактор vi:

vi name_file

В данном случае откроется пустой текстовый файл с именем name_file для редактирования, чтобы он записался на диск надо будет его закрыть с сохранением на диск.

Второй способ легче — достаточно всего одной команды touch:

touch имя_файла

6) Переименование файла

Команда mv позволяет вам переименовать файл в том же каталоге или переместить его из одного каталога в другой. Когда перемещаете файл в другой каталог, то можете переименовать его или оставить прежнее имя.

Синтаксис команды:

mv имя-файла1 имя-файла2

Находясь в каталоге /home/test/draft создайте текстовый файл **table.txt** Переименуйте файл table.txt в new_table.txt:

mv table.txt new_table.txt

Проверьте, что файл new.table существует, командой **ls -l**.

Чтобы переместить файл в другой каталог без изменения имени, введите команду с указанием целевого каталога, например:

mv new_table.txt dir1/

7) Перемещение и переименование каталога

Находясь в папке /home/test/draft создайте каталог TEST_FOLDER

Переименуйте его в TEST_FOLDER_NEW:

mv TEST_FOLDER TEST_FOLDER_NEW

Переместите его на уровень выше, то есть в /home/test:

mv TEST_FOLDER_NEW /home/test/ TEST_FOLDER_NEW

Задание 2.4 Стандартные файлы ввода-вывода, просмотр содержимого файлов

1. Просмотр содержимого текстового файла.

Находясь в каталоге /home/test/draft/dir1 просмотрите содержимое файла:

new_table.txt

Введите команду

cat new_table.txt

Введите команду постраничного вывода файла

more new_table.txt

Система выведет на экран текст из файла new_table.txt. Если в файле больше строк, чем вмещает экран, вывод будет произведен постранично. Для просмотра следующей страницы необходимо нажать клавишу <пробел>.

Файл new_table.txt у нас пустой, так что использование команды more для него бессмысленно. Чтобы проверить как она работает используем просмотр текстового файла, содержащего много настроек. Таковым, к примеру, является файл /etc/group, содержащий список групп в Linux.

more group

2. Просмотр данных текстового файла с конца и с начала.

Дело в том, что в UNIX системах встречаются файлы, содержащие не сотни и тысячи строк, но порой и миллионы. Пример таких файлов является файлы логов системы или различных служб.

В OpenSuSE таковым является файл audit.log аудита работы различных сервисов, находящийся в /var/log/audit, рисунок 5:

Рисунок 5.

Обычно требуется просмотреть определенное количество строчек файла с начала или конца файла. Для чего использует программы-команды `head` и `tail`.

Просмотрите первые 20 строчек файла `/var/log/audit/audit.log`:

`head -n 20 /var/log/audit/audit.log`

Просмотрите последние 35 строчек файла `/var/log/audit.log`:

`tail -n 35 /var/log/audit/audit.log`

Обычно в таких файлах ищется какая-то информация, для чего используется программа-фильтр `grep`. Чтобы найти все действия, выполненные с помощью учетной записи суперпользователя `root`, в последних 1000 строках файла, используйте команду:

`tail -n 1000 /var/log/audit/audit.log |grep -i root`

На самом деле опция `-i` для работы программы фильтра `grep` не является обязательной, но она позволяет искать, игнорируя регистр, что бывает часто полезно.

3) Перенаправление ввода/вывода

Стандартные вход/выход можно изменять (перенаправлять). Синтаксис команды с перенаправлением ввода

`command < filename`

Синтаксис команды с перенаправлением вывода

`command > filename`

Перенаправление и добавление вывода к файлу

`command >> filename`

Давайте теперь выведем информацию о текущем времени и дате, которую нам предоставляет команда `date` в файл, заодно не только вывод результатов выполнения команды в файл, но и еще один способ создания текстовых файлов:

Перейдите в каталог `/home/test` и введите команду:

date > date_time.txt

Командой **ls -l** проверьте созданся ли файл **date_time.txt**

Разумеется, он должен был появиться, если вы все сделали правильно. Проверьте перенаправился ли вывод команды **date** в файл:

cat date_time.txt

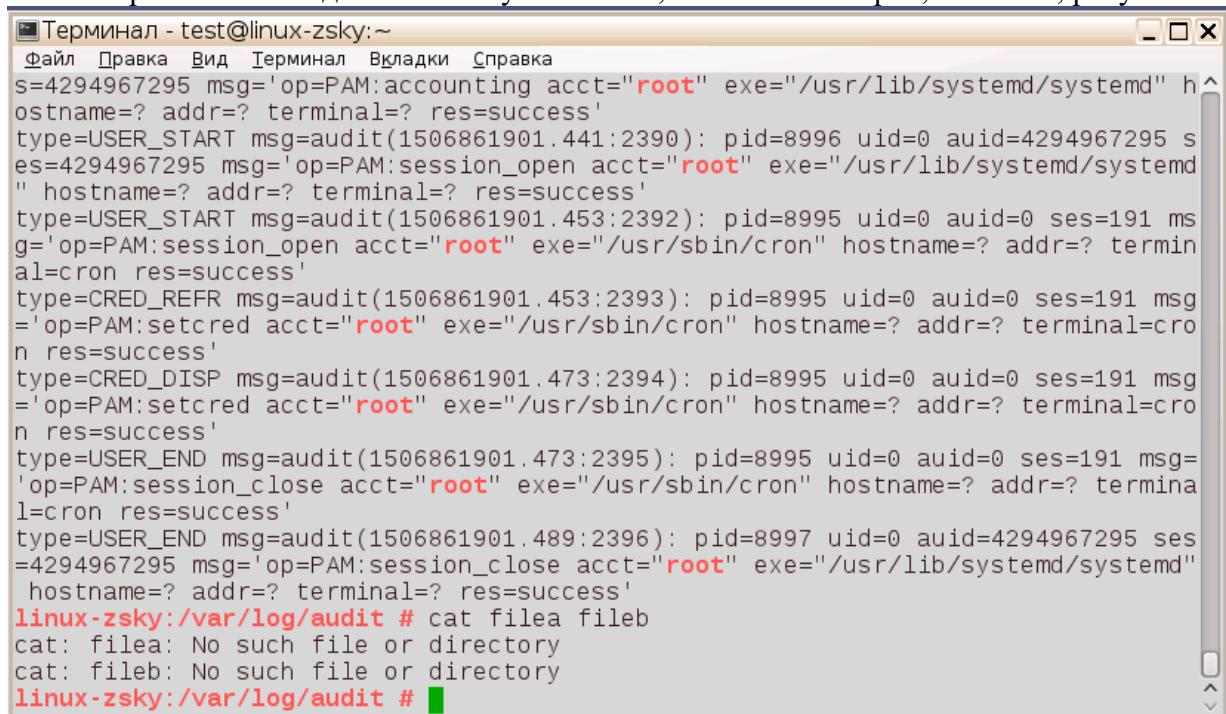
А как же быть если, нам нужно дозаписать новый вывод команды в файл? Дело в том что стандартный вывод с помощью оператора **>**, перезапишет файл заново, этого нам не надо. Для дозаписи в файл существует оператор **>>**:

date >> date_time.txt

Проверьте результат командой **cat**, вы должны увидеть в файле уже две записи о времени.

3) Иногда нужно перенаправлять в текстовый файл для последующего изучения не только стандартный вывод консольных программ, но и вывод ошибок их выполнения.

Как работает вывод ошибок по умолчанию, а именно на экран, а вот как, рисунок 6:

A screenshot of a terminal window titled "Терминал - test@linux-zsky: ~". The window shows a series of audit logs from the PAM system, including session opening and closing for the 'root' user. At the bottom, the user enters the command 'cat filea fileb' and receives two error messages: 'cat: filea: No such file or directory' and 'cat: fileb: No such file or directory'. The prompt then returns to 'linux-zsky:/var/log/audit #'.

```
Терминал - test@linux-zsky: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
s=4294967295 msg='op=PAM:accounting acct="root" exe="/usr/lib/systemd/systemd" h
ostname=? addr=? terminal=? res=success'
type=USER_START msg=audit(1506861901.441:2390): pid=8996 uid=0 auid=4294967295 s
es=4294967295 msg='op=PAM:session_open acct="root" exe="/usr/lib/systemd/systemd
" hostname=? addr=? terminal=? res=success'
type=USER_START msg=audit(1506861901.453:2392): pid=8995 uid=0 auid=0 ses=191 ms
g='op=PAM:session_open acct="root" exe="/usr/sbin/cron" hostname=? addr=? termin
al=cron res=success'
type=CRED_REFR msg=audit(1506861901.453:2393): pid=8995 uid=0 auid=0 ses=191 msg
='op=PAM:setcred acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cro
n res=success'
type=CRED_DISP msg=audit(1506861901.473:2394): pid=8995 uid=0 auid=0 ses=191 msg
='op=PAM:setcred acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cro
n res=success'
type=USER_END msg=audit(1506861901.473:2395): pid=8995 uid=0 auid=0 ses=191 msg=
'op=PAM:session_close acct="root" exe="/usr/sbin/cron" hostname=? addr=? termina
l=cron res=success'
type=USER_END msg=audit(1506861901.489:2396): pid=8997 uid=0 auid=4294967295 ses
=4294967295 msg='op=PAM:session_close acct="root" exe="/usr/lib/systemd/systemd"
hostname=? addr=? terminal=? res=success'
linux-zsky:/var/log/audit # cat filea fileb
cat: filea: No such file or directory
cat: fileb: No such file or directory
linux-zsky:/var/log/audit #
```

Рисунок 6.

Здесь мы попытались вывести на экран содержимое файлов **filea** и **fileb**:

cat filea fileb

Получим на экране сообщение об ошибке — **No such file or directory**.

Все как бы хорошо, но сообщения ошибок не будут перенаправляться в текстовый файл операторами **>** и **>>**. Для того чтобы ошибки перенаправить в файл для последующего анализа, нам нужно использовать операторы **2>** и **2>>**.

Общий синтаксис:

command 2> filename

или с добавлением в файл:

command 2>> filename

Рассмотрим пример:

cat filea fileb 2> errfile

cat errfile

4) Просмотр текстовых файлов с помощью специализированного ПО.

Разумеется, текстовые файлы можно смотреть не только с помощью утилит cat и more.

Их просмотра используется консольные текстовые редакторы **vi** / **vim**, **nano**. Ну и разумеется всеми любимым двух панельный файловый менеджер **mc**.

Задание 2.5 Текстовый редактор vi

Редактор vi - это полноэкранный текстовый редактор, входящий в состав ОС UNIX всех типов. Его приходится использовать на начальных стадиях настройки ОС, до установки более удобных инструментов редактирования файлов или в случаях отказов и сбоев в работе ОС, при необходимости ее восстановления.

1) Режимы редактора vi

РЕЖИМ КОМАНДЫ: в этом режиме сигнал с клавиатуры интерпретируется как команда редактирования или команда перехода в другие режимы.

РЕЖИМ ВСТАВКИ: переход в этот режим осуществляется набором команд вставки, присоединения, подстановки, изменения или замещения, имеющихся в vi. В режиме вставки символы, набранные на клавиатуре, вставляются в буфер редактирования. Для завершения режима вставки и возврата в командный режим следует нажать ESC.

РЕЖИМ ПЕРЕКЛЮЧЕНИЯ В «execution» (режим последней строки): В vi команды - это одиночные клавиши. В их командами являются строки текста, завершенные нажатием клавиши ENTER. Для перехода в режим переключения в их применяют символ ':'. Этот символ отображается в командной строке (последней строке экрана) как указатель на последующую команду редактора ex. Большинство команд обработки файла выполняются в режиме переключения в ex (например, команды чтения из файла и записи из буфера редактирования обратно в файл).

2) Вызов vi.

Наберите команду vi и укажите имя файла, который Вы хотите отредактировать:

vi <имя файла>

Столбец символов "~" говорит о том, что Вы находитесь в конце файла.

В командном режиме вы можете использовать клавиши со стрелками для перемещения по файлу.

3) Вставка текста. Переход из режима команд по команде «i» (вставка в текущей позиции), «a» (вставка в конец строки), «o» (вставка новой строки под текущей).

При вставке текста Вы можете напечатать столько строк, сколько пожелаете (нажимая ENTER после каждой строки), и можете корректировать ошибки используя клавишу возврата (BACKSPACE).

Для завершения работы в режиме вставки и возврата в командный режим нажмите ESC.

4) Удаление текста.

В командном режиме команда «x» удаляет символ перед курсором.

Вы можете удалять целые строки, набирая команду «dd» (т.е. нажимая «d» дважды).

Чтобы удалить слово, на котором находится курсор, используйте команду «dw».

5) Команды перемещения.

Команды:

w - перемещает курсор на начало следующего слова;

b - перемещает на начало предыдущего слова.

0 (это ноль) - передвигает курсор на начало текущей строки

\$ - перемещает курсор в конец строки.

Для перемещения вперед и назад сразу на размер экрана используют ctrl-F (курсор перемещается на экран вперед) и ctrl-B (на экран назад).

Для того чтобы переместить курсор в конец файла, следует напечатать «G». Также можно переместиться на любую строку. Напечатав команду «10G», Вы переместите курсор на десятую строку файла. Для того чтобы вернуться к началу (в первую строку), используйте «1G».

б) Сохранение файлов и выход из vi.

В режиме последней строки (после нажатия «:» в командном режиме):

w – записать изменения в файл на диск

q! - выйти из vi без внесения изменений в ранее существовавший файл.

wq - записать изменения в файл, а затем выйти из vi.

С помощью редактора vi будет необходимо создать текстовый файл со следующим содержанием:

#	device name	mount point	fs-type	options	dump-freq	pass-num
	LABEL= /	/	ext3	defaults	1	1
	/dev/hda6	swap	swap	defaults	0	0
	none	/dev/pts	devpts	gid=5,mode=620	0	0
	none	/proc	proc	defaults	0	0
	none	/dev/shm	tmpfs	defaults	0	0

Отчет по лабораторной работе номер должен состоять из скриншотов вывода командной строки Linux при выполнении всех 5 заданий.

Для последнего задания необходимо привести скриншот вывода, созданного и отредактированного в редакторе vi файла с помощью команды:

cat file_name

ВНИМАНИЕ! Все задания нужно выполнять внутри вашей домашней папки, то есть внутри /home/имя_вашей_учетной_записи, в объяснениях к работе подразумевается, что вы работаете под учетной записью test.

Во время отчета по работе вы должны продемонстрировать как вы работает в консольном редакторе vi, если вы сделать этого не можете работа считается списанной и не проработанной.

Задание 2.6 Работы со средствами архивации и упаковки файлов

Как уже писалось в пункте 1.5 теории к практической работе процесс архивации и упаковки файлов не есть одно и тоже. Так как в Linux нет умолчательных программ которые могут не только архивировать файлы, но и упаковывать папки.

Тут стоит сделать замечание, что все-таки такое ПО есть, а именно это всем известный архиватор winrar, которые отлично работает и в Linux как раз-таки в режиме командной строки. Но есть одно но, это ПО коммерческое и поэтому на большинстве Linux систем оно отсутствует по умолчанию.

- 1) Перейдите в директорию /home/test и создайте файл путем перенаправления вывода команды **date** с именем file_date_FIO.txt

Примечание под FIO имеется ввиду ваша фамилия, имя и отчество записанная

латиницей в формате `ivanovii` (для Иванов Иван Иванович).

- 2) Заархивируйте данный файл с помощью архиватора `gzip` с максимальной степенью сжатия.

Пример:

`gzip -9 file_date_ivanovii.txt`

*После процедуры сжатия вы получите архив с именем **`file_date_ivanovii.txt.gz`**.*

- 3) Создайте в домашней папке пользователя `test` папку с именем `documents_FIO`
- 4) Переместите полученный на шаге 2 архив в данную папку.
- 5) Перейти в папку `/home/test/documents_FIO` и распаковать находящийся там архив командой:

`gunzip file_date_FIO.txt.gz`

- 6) Проверить, что файл **`file_date_FIO.txt.gz`** распаковался и вывести на экран содержимое файла `file_date_FIO.txt`, командой `cat`.