Архитектура сетей и систем телекоммуникаций.

Практическая работа 2.

Основы работы в ОС UNIX

Цель работы:

* освоить основные принципы работы операционной системы UNIX;
* приобрести навыки использования интерфейса командной строки и простейших команд управления программами, файлами и устройствами UNIX;
* приобрести навыки управления устройствами под управлением промышленной операционной системы.

1. Теоретические основы

Промышленная операционная система (ОС семейства UNIX) - это набор программ, которые обеспечивают широкие возможности по управлению компьютером, обеспечивающим сетевые сервисы или управление внешними (периферийными) устройствами. ОС UNIX позволяет решать следующе задачи:

* обеспечение программно-аппаратной платформы для выполнения широкого спектра прикладных программ;
* реализация интерактивного пользовательского интерфейса для управления компьютером и периферией;
* обеспечение многопользовательского режим работы, который обеспечивает разделяемый доступ к ресурсам компьютера или к внешним ресурсам, которые находятся под его управлением, например, доступ к сетям, объединяемым маршрутизатором;
* многозадачный режим работы, который позволяет ЭВМ выполнять несколько задач одновременно, в том числе в параллельном режиме на многопроцессорных системах.

Имеется много реализаций операционной системы UNIX – Mac OS X, FreeBSD, Linux, Sun Solaris, HPUX, AIX, SCO и др. К их семейству можно отнести и операционные системы активных сетевых устройств разных производителей, например, CISCO IOS. Как бы не называлась nix-подобная система все они объединены общей архитектурной концепцией и организацией пользовательского интерфейса. Поэтому, изучив основы одной ОС, Вы сможете разобраться и в любой другой.

1.1 Состав ОС UNIX.

Система UNIX состоит из 4-х основных компонент:

***Ядро*** (kernеl)- это программа, которая образует ядро операционной системы; она координирует внутренние функции компьютера (такие как размещение системных ресурсов). Ядро работает невидимо для вас. Ядро контролирует доступ к компьютеру и его компонентам, управляет памятью компьютера, обслуживает файловую систему и распределяет ресурсы компьютера среди пользователей;

***Оболочка*** (shell) - это программа, которая осуществляет связь между вами и ядром, интерпретируя и выполняя ваши команды. Так как она читает ваш ввод и посылает вам сообщения, то описывается как интерактивная; shell считывает команды, которые вы вводите, и интерпретирует их как запросы на выполнение других программ, на доступ к файлу или обеспечение вывода. ОС UNIX содержит несколько пользовательских оболочек, каждая из которых включает в себя интерпретатор языка программирования, родственного Си по синтаксису. Одна из стандартных оболочек- csh- содержит интерпретатор Си.

***Команды*** (commands) - это имена программ, которые компьютер должен выполнить. Пакеты программ называются инструментальными средствами. Система UNIX обеспечивает инструментальными средствами для таких заданий как создание и изменение текста, написание программ, развитие инструментария программного обеспечения, обмен информацией с другими посредством компьютера;

***Файловая система*** (file system) - это набор всех файлов, возможных для вашего компьютера. Она помогает вам легко сохранять и отыскивать информацию. Файловая система UNIX имеет иерархическую структуру; чаще всего она описывается в виде дерева. *Вершина этого дерева - это каталог корень. Он обозначается с помощью символа /.*

Файловая система построена из обыкновенных файлов, символических связей и каталогов.

Обыкновенные файлы - это набор символов, хранящихся на диске. Они могут хранить тексты или программы или коды программ.

Специальные файлы UNIX- файлы устройств- соответствуют физическим устройствам (например, терминал, диск или последовательный порт компьютера) или виртуальным устройствам, например, программному генератору случайных чисел.

Символическая связь - это файл, который указывает на другой файл (или каталог).

Каталог является специальным файлом операционной системы и содержит имена входящих в него других каталогов (называемых подкаталогами) или файлов. Файлы вы можете группировать в каталоги по своему выбору.

Всякий раз, когда вы взаимодействуете с системой UNIX, то вы находитесь в определенном месте структуры файловой системы (текущем каталоге). К файлам текущего каталога пользователь может обратиться без указания полного имени. Полное имя файла включает в себя имена всех каталогов, к которым принадлежит файл, начиная от корневого. Например, /etc/rc2.d/S99start\_program.

1.2 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 системе нового пользователя).

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

**5.** Чтобы ваш запрос был понятен системе UNIX, вы должны ввести соответствующую команду в корректном формате и синтаксисе командной строки. Этот синтаксис определяет порядок, в котором вы вводите компоненты командной строки. И вы должны расположить все составные части командной строки в требуемом синтаксисом порядке, иначе shell не сможет интерпретировать ваш запрос.

1.3 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:

**/** 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

програмцм. 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.

1.4 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.

Например:

ls -l -i file1 file2 file3

| | |

команда ключи аргументы

В этом примере команда ls использует два ключа -l и -i и три аргумента file1, file2 и file3. Ключ -l определяет формат вывода (длинный, в одну колонку), ключ –i обеспечивает вывод системного кода файла inode. Система UNIX обычно позволяет вам группировать ключи, например -li, и выводить их в любом порядке. Этого нельзя делать с аргументами.

Для поиска исполнимых программ (ls в примере) оболочка использует полное имя файла, а если указано только имя команды, то сначала текущий каталог, а потом каталоги, перечисленные в системной переменной оболочки, которая называется PATH.

Системные (глобальные) переменные оболочки позволяют настраивать многие особенности ее поведения. Присвоение значения переменной – это команда оболочки следующего формата: <ИМЯ ПЕРМЕННОЙ>=<значение>. Например, PATH=/usr/bin

1.5 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](http://en.wikipedia.org/wiki/Gzip) или [bzip2](http://en.wikipedia.org/wiki/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 понадобиться 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 -c. 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 gbiz2:

tar -cvjf files.tar.bz2 ~/files

Метод 2:

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

Пример:

tar –cvf arhive.tar \*.txt

gzip archive.tar или bzip2 arhive.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:

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 – arhive.tar.gz

Распаковка состоит из двух этапов:

Этап 1: gunzip arhive.tar.gz

Этап 1: tar -xvf arhive.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. Ход работы

Задание 1. Получение доступа к системе

Если вы используете виртуальную машину, включите ее и используйте консоль. Если вы подключаетесь к серверу UNIX с использованием сети, необходимо осуществить терминальное подключение к нему. Для этого воспользуйтесь программой PuTTY.

На экране должно появится приглашение, содержащее слово login.

После того как на экране появится приглашение ввода имени пользователя, вы должны ввести имя пользователя root (административный пользователь) и нажать клавишу ENTER.

*Примечание: UNIX различает прописные и строчные буквы.*

Затем на экране появится слово password. Это значит, что необходимо ввести пароль. Вводимые символы не будут видны на экране.

В том случае, если Вы ввели пароль неправильно, на экране появиться сообщение login incorrect и новое приглашение.

Войдя в систему, Вы получите приглашение на ввод команд – на экране появится символ #, означающий, что система готова принимать Ваши команды.

Для выхода из системы введите команду exit или logout.

Задание 2. Простейшие команды

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.

5) Вывод строки на экран. Введите команду

echo Lanch is at 12-00

На экран выведется содержимое строки после имени команды echo.

6) Команда вызова справки man. Введите команду

man ls

На экран выведется справка по команде ls

7) Очистка экрана, команда

clear

Задание 3. 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.

3) 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?

4) Просмотр содержимого каталога.

ls -l /etc

на экране выведена информация о директории etc в расширенном формате

ls –l

На экране появится нечто подобное следующему:

drwx-xr-x 3 startship project 96 Oct 27 08:16 bin

drwx-xr-x 2 startship project 64 Nov 1 14:19 draft

drwx-xr-x 2 startship project 80 Nov 8 08:41 letters

-rwx----- 2 startship project 12301 Nov 2 10:15 list

-rw------ 1 startship project 40 Oct 27 10:00 mbox

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.

Введите команду

ls -al /usr/home/student/.profile

система выдаст информацию о файле .profile в каталоге /usr/home/student/.

5) Создание директории

mkdir dir\_name

Создастся директория с именем dir\_name

mkdir –p dir1/dir2/dir3

Создание нескольких директорий

6) Удаление директории

rmdir dir\_name

Удаление нескольких директорий

mkdir –p dir1/dir2/dir3

Задание 4. 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 файлов

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

cp ключи имя-файла1 имя-файла2

cp ключи имя-файла(ов) каталог

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

cp /usr/home/student/.profile ./new.profile

Чтобы проверить существование нового файла, введите команду:

ls

4) 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

Например, если вы находитесь в каталоге draft и хотите переименовать файл table на new.table, введите команду:

mv table new.table

Проверьте, что файл new.table существует.

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

mv new.table dir1/

5) Просмотр содержимого текстового файла.

Введите команду

cat dir1/new.table

Введите команду постраничного вывода файла

more dir1/new.table

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

6) Копирование папки

Для копирования директории необходимо указывать опцию –R

copy –R /home/test/folder1 /home/user2/folder1

7) 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

Задание 5. Стандартные файлы ввода-вывода

Три файла автоматически открываются для каждого процесса в системе. Это standard input, standard output и standard error.Сокращенно: stdin, stdout и stderr. Символьные обозначения для них >, < и 2>. По умолчанию стандартным входом является клавиатура, а для стандартных вывода и ошибок – дисплей (или окно терминала).

1) 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

3) Создание файла с помощью команды date

Введите команду

ls

Введите команду

date > date\_time.txt

Введите команду

ls

Вы обнаружите появление файла date\_time.txt. Проверьте результат выполнения команды – введите команду

cat date\_time.txt

3) Перенаправление ошибок.

Вывод ошибок по умолчанию:

Попытаемся вывести файл filea следующим образом:

cat filea fileb

Получим на экране сообщение об ошибке.

Сообщение об ошибке можно записать в файл. Общий синтаксис:

command 2> filename:

или с добавлением в файл:

command 2>> filename

Пример:

cat filea fileb 2> errfile

cat errfile

4) Комбинированное перенаправление. Примеры комбинированного перенаправления:

command < infile > outfile 2> errfile

command >> appendfile 2>> errfile < infile

Задание 6. 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 ex командами являются строки текста, завершенные нажатием клавиши ENTER. Для перехода в режим переключения в ex 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».

6) Сохранение файлов и выход из vi.

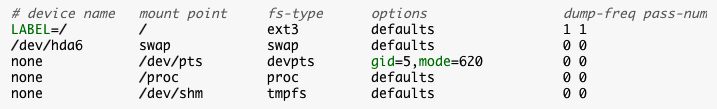
В режиме последней строки (после нажатия «:» в командном режиме):

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:



Отчет по лабораторной работе номер два должен состоять из скриншотов вывода командной строки Linux при выполнении всех шести 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**

**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 7. Работы со средствами архивации и упаковки файлов.

1. Перейдите в директорию /root и создайте файл путем перенаправления вывода команды ***date*** с именем file\_date\_FIO.txt
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.*

1. Создайте в домашней папке пользователя root папку с именем documents\_FIO
2. Переместите полученный на шаге 2 архив в данную папку.
3. Перейти в папку /root/documents\_FIO и распаковать находящийся там архив командой gunzip file\_date\_FIO.txt.gz.
4. Вывести на экран содержимое файла file\_date\_FIO.txt, командой cat.
5. Перейдите в домашний каталог пользователя root и упакуйте каталог documents\_FIO командой tar:

**tar –cvf /root/documents\_FIO.tar > /dev/null /root/documents\_FIO**

1. У вас должен появиться файл doceuments\_FIO.tar, сожмите его архиватором gzip с максимальной степенью сжатия.

После выполнения команды у вас должен получиться файл:

Docements\_FIO.tar.gz

5. Вопросы для подготовки к отчету:

1. Постарайтесь кратко пояснить для чего нужен каждый из каталогов корневой файловой системы UNIX.
2. Чем отличаются приглашения командной строки пользователя и супер администратора root.
3. С помощью каких команд можно перемещаться по файловой системе UNIX в командной строке, а так же определять свое текущее местоположение?
4. Вы находитесь в каталоге /etc, вам надо вернуться на каталог выше — в корневой каталог, какими двумя способами это можно сделать?
5. Как в командной строке UNIX проще всего создать пустой текстовый файл.
6. Назовите стандартные файлы ввода-вывода в UNIX.
7. Если мы не знаем, что делает команда, как мы можем в командной строке посмотреть по ней справку.
8. С помощью какой команды можно вывести в окне терминала содержимое текстового файла? И какие опции стоит применять для удобства чтения информации, если этот файл содержит большое количество строк.
9. Расскажите кратко как с помощью редактора vi можно редактировать простой текстовый файл.

6. Список литературы:

**1.** Лебланк, Ди-Анн, Хоуг, Мелани, Бломквист, Эван. Linux для "чайников", 4-е издание. : Пер. с англ. — М. : Издательский дом "Вильяме",

2003. — 336 с.

**2.** Колесниченко Д.Н. Linux-сервер своими руками. СПб: Наука и техника, 2002. -576 стр.

**3.** *Сивер Э., Спейнауэр С„ Фиггинс С., Хекман Д.* Linux. Справочник. - Пер. с англ. - СПб: Символ-Плюс, 2001. - 912 с., ил.

**4.** В. Белунцов Самоучитель пользователя Linux — Москва: “ДЕСС КОМ”, 2003. - 512 с., илл.

**5**. Шредер К. Linux. Сборник рецептов. — СПб.: Питер, 2006. — 432 с: ил.