Usage of Simple Commands

In this section we shall try to learn about the usage of some simple commands.

date Command

The date command tells us the current date and time.

Example 1.1

```
$ date
Thu   Nov  1 . 09:34:50 PST 1984
$ _
```

who Command

The output of the Who command gives us the details of the users who have logged in to the LINUX system currently.

Example 1.2

```
$ who
abc    tty0    Sep  26 11:17
xyz    tty4    Sep  26 11:30
lkj    tty9    Sep  26 11:48
$ _
```

We can also try the command,

Example 1.3

```
$ who am i
user1   ttya    Sep  26 12:20
$ _
```

which gives us details regarding the login time and the system's name for the connection being used.

man Command

If we get stuck on something, and cannot find an expert to help, we can print any manual page on our terminal with the command 'man' command-name.

Example 1.4

```
$ man who
```

head and tail Command

The head command is used to display the initial part of a text file. This can be thought of as a complement command to the tail command, which displays the last part of a text file. By default head command displays first 10 lines of a file and tail command the last 10 lines.

```
$ head  [ - count] [filename ...]
```

Let us assume that we have a file called wordlist and if we want to display the first four lines, the command to be given is :

Example 1.5

```
$ head -4 wordlist
pen
bat
ink
eraser
$ _
```

will display the first four lines of a file called wordlist.

The syntax for the tail command is

```
$ tail [+|- number] file
```

We can display lines of file beginning with an offset of + number from the beginning or - number lines from the end of the specified file.

Let us assume we have a file called mall and we want to display from the 10th line from the beginning of the file, then the syntax is,

Example 1.6

```
$ tail +10  mall
$ _
```

C. Points to Ponder

State True or False

1. The date –c command is used to display the current date.
2. The man command is used to get on-line help on any command.
3. The tail command is used to display the first portion of the text file.

Fill in the blanks

1. The _______ command is used as a complement to the tail command.
2. The _______ command is used to display the users details who have logged in.
3. The _______ command is used to see current date.

Usage of Directory Commands

Let us look at some directory commands.

pwd Command

The pwd command has no options. It displays the full pathname for the current directory we are working in.

The syntax is $ pwd

Example 1.7

```
$ pwd
/usr/temp
$_
```

ls Command

The ls command displays the list of files in the current working directory.

```
$ ls
```

Example 1.8

ls – list System director

ll – long list

```
$ ls
abc.c
xyz
lkj
$ _
```

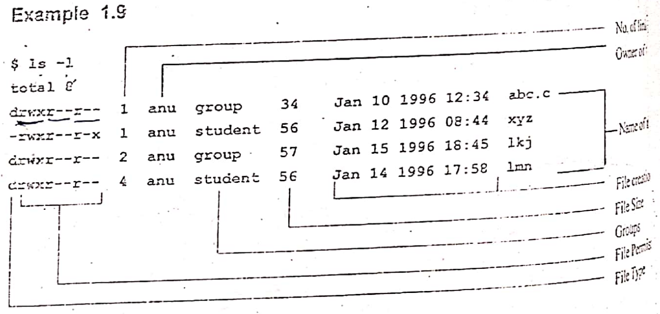
This gives us the list of files in the current directory .

The ls command comes with different options. Certain options are given in Table 1.1

ls -l	Lists files in the long format. The files are displayed along with their mode, number of links, owner of the file, file size, modification date and time and filename.
ls -t	Lists in order of last modification time (most recent first).
ls -a	Lists all entries, including the hidden files.
ls -d	Lists directory file instead of its contents.
ls -p	Puts a slash after each directory.
ls -u	Lists in order of last access time.

Table 1.1

Example 1.9



```
$ ls -l
total 8
drwxr--r--  1  anu  group    34   Jan 10 1996 12:34  abc.c
-rwxr--r-x  1  anu  student  56   Jan 12 1996 08:44  xyz
drwxr--r--  2  anu  group    57   Jan 15 1996 18:45  lkj
drwxr--r--  4  anu  student  56   Jan 14 1996 17:58  lmn
```

mkdir Command

This command is used to create a new directory.

```
$ mkdir  directoryname
```

Example 1.10

```
$ mkdir temp
$ _
```

To verify the creation of the above directory, use the ls command.

cd Command

The ~~rmdir~~ cd command is used to change from the working directory to any other directory specified. There are no options connected with this.

```
$cd
```

Example 1.11

```
$ cd jack
$ pwd
/jack
```

rmdir Command

This command is used to remove a directory specified in the command line. It requires the specified directory to be empty before removing it.

```
$  rmdir  directoryname
```

Example 1.12

Suppose we wish to remove the directory **temp** and let us assume that this directory did not have any files in it, then

```
$ rmdir temp
$ _
```

Usage of File Commands

We shall continue with getting to know some simple file commands.

cat Command

The **cat** command helps us to list the contents of a file we specify. If we do not specify the name of the file, it takes input from the standard input.

```
$ cat [option...] [file... ].
```

The cat command comes with different options. They are

cat Takes input from the standard input file. We can type in our text and terminate with Ctrl+d to take us back to $ prompt.

cat -s Suppresses warnings about non -existent files.

Assume that we want to display the contents of a file xyz. The command is :

Example 1.13

```
$ cat xyz
To be or not to be.  That is the question.
$ _
```

To create a file called mat,

```
$ cat > mat
Hi ! have a nice day
Ctrl+d
$ _
```

In order to append data to the same file, use cat command with the ">>" symbol.

cp Command

The cp command is used to create duplicate copies of ordinary files.

```
$ cp file target
```

Target is the file to which the contents of the file has to be copied. If the name of the target file already exists, the original contents are erased and the contents of the file are written on the target.

Example 1.14

Let us assume that we have a file called abc and its contents have to be copied to lkj, which is empty then the command is,

```
$ cp abc lkj
```

ln - link Command

The ln command is to establish an additional filename for the same ordinary file.

```
$ ln firstname secondname
```

Here firstname is the name of the file for which we would like to establish an additional name or link. The secondname is the additional name of the file firstname. The advantage of link command is that several users can have access to a common data file. Any modification made in the additional file or in the original file is reflected in both of them.

Linking allows a user to access a file by means of two or more names. The alternative names can be located in the same or different directory. Since Linux is a multi-user operating system, it allows the user to protect the data from other users.

Example 1.15

To create an additional name called rose for a file called red the command is,

```
$ ln red rose
$ _
```

mv Command

The mv command is used to rename and move ordinary and directory files. To do this we need both execute and write permissions.

```
$ mv file  target
```

$ touch file name ↵

Cat > raj ↵ To add the content

Ctrl+Z

Cat raj ↵ To see The content

Example 1.16

To move the contents of the file old to the file new, the syntax is :

```
$ mv old new
$ _
```

The file old no longer exists in the directory.

rm Command

The rm command is used to remove one or more files from a directory. This can be u[sed to] delete all files as well as the directory.

```
$ rm [option...] file
```

The options available are

rm -i	Asks the user if he wants to delete the file mentioned. When this is combined with -r option, rm asks whether to examine each file in that directory.
rm -r	Recursively delete the entire contents of the directory as well as the directory itself.

Table 1.2

Example 1.17

If we want to delete a file called mat which is under the directory temp.

```
$ rm   temp/mat
$ _
```

We can use -i and -r option to interact with the rm command and then delete a directory.

$ clear - For clear the screen

$ cal - To see calander

$ wall ↵ For send the message