

## EXPERIMENT NO:-1

**AIM:-**To study and implement the internal commands of Linux like ls, chdir, mkdir, chown, chmod, chgrp, ps, etc

### THEORY:-

#### WHAT IS LINUX?

Linux is an Operating System's Kernel. You might have heard of UNIX. Well, Linux is a UNIX clone. But it was actually created by Linus Torvalds from Scratch. Linux is free and open-source, that means that you can simply change anything in Linux and redistribute it in your own name! There are several Linux Distributions, commonly called "distros". A few of them are:

- Ubuntu Linux
- Red Hat Enterprise Linux
- Linux Mint
- Debian
- Fedora

#### 1. Command - ls

<b>Name</b>	list command
<b>Purpose</b>	functions in the Linux terminal to show all of the major directories filed under a given file system, will also show the user all of the folders stored in the specified folder. <b>Options:</b> <ul style="list-style-type: none"><li>a- used to list all the files including the hidden files.</li><li>c - list all the files columnwise.</li><li>d - list all the directories.</li><li>m - list the files separated by commas.</li><li>p - list files include ,/“ to all the directories.</li><li>r - list the files in reverse alphabetical order.</li><li>f - list the files based on the list modification date.</li><li>x - list in column wise sorted order.</li></ul>
<b>Syntax</b>	\$ ls – options <arguments>
<b>Example</b>	ls /

#### OUTPUT:

```
user@user-H81M-S:~$ ls /
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  swapfile  tmp  var
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
```

## 2. Command – chmod

<b>Name</b>	change mode command
<b>Purpose</b>	Permissions can be changed by owner of the file <b>Symbolic modes-</b> User(u) - the owner of the file Group(g) - users who are members of the file's group Others(o) - users who are not the owner of the file or members of a group All(a) - three of the above; is the same as ugo Read(r) - read a file or list a directory's contents Write(w) - write to a file or directory Execute( x) - execute a file or recurse a directory tree
<b>Syntax</b>	\$ chmod ug+x file
<b>Example</b>	\$ chmod 400 test.php

### OUTPUT:

```
(nirajy@niraj)-[~]
$ chmod 777 niraj

(nirajy@niraj)-[~]
$ ls -al
total 228
drwx----- 23 nirajy nirajy 4096 Jan 31 23:54 .
drwxr-xr-x  3 root  root  4096 Dec 27 22:38 ..
-rw-----  1 nirajy nirajy    0 Dec 27 23:29 .ICEauthority
-rw-----  1 nirajy nirajy   50 Jan 31 23:38 .Xauthority
-rw-r--r--  1 nirajy nirajy  220 Dec 27 22:38 .bash_logout
-rw-r--r--  1 nirajy nirajy 5551 Dec 27 22:38 .bashrc
-rw-r--r--  1 nirajy nirajy 3526 Dec 27 22:38 .bashrc.original
drwxr-xr-x  9 nirajy nirajy 4096 Dec 28 15:44 .cache
drwxr-xr-x 14 nirajy nirajy 4096 Jan 26 12:31 .config
-rw-r--r--  1 nirajy nirajy   35 Dec 27 23:29 .dmrc
-rw-r--r--  1 nirajy nirajy 11613 Dec 27 22:38 .face
```

## 3.Command - Sort

<b>Name</b>	sort command
<b>Purpose</b>	sort command can be used to get sorted content
<b>Syntax</b>	\$ sort file
<b>Example</b>	Let's say you have a file, <b>data.txt</b> , which contains the following ASCII text: apples oranges pears kiwis bananas To sort the lines in this file alphabetically, use the following command: \$ sort data.txt

## OUTPUT:

```
user@user-H81M-S:~$ sort new.txt
Goa
Gujarat
Haryana
Maharashtra
Panjab
user@user-H81M-S:~$
```

## 4.Command – chdir

Name	chdir command- change directory
Purpose	<b>chdir</b> is the system function for changing the current working directory.
Syntax	<b>\$ chdir</b> name of the directory
Example	chdir /

## OUTPUT:

+

```
File Actions Edit View Help
chdir: not a directory: file.txt

(root@niraj)-[~]
# chdir /

(root@niraj)-[/]
# ls -l
total 68
lrwxrwxrwx 1 root root 7 Dec 27 20:28 bin -> usr/bin
drwxr-xr-x 3 root root 4096 Dec 27 22:52 boot
drwxr-xr-x 17 root root 3380 Jan 31 23:37 dev
drwxr-xr-x 162 root root 12288 Jan 31 23:37 etc
drwxr-xr-x 3 root root 4096 Dec 27 22:38 home
lrwxrwxrwx 1 root root 33 Dec 27 20:31 initrd.img -> boot/initrd.img-6.5.0-kali3-amd64
lrwxrwxrwx 1 root root 33 Dec 27 20:31 initrd.img.old -> boot/initrd.img-6.5.0-kali3-amd64
lrwxrwxrwx 1 root root 7 Dec 27 20:28 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Dec 27 20:28 lib64 -> usr/lib64
drwx----- 2 root root 16384 Dec 27 20:28 lost+found
drwxr-xr-x 3 root root 4096 Dec 27 20:28 media
drwxr-xr-x 2 root root 4096 Dec 27 20:28 mnt
drwxr-xr-x 4 root root 4096 Dec 27 21:08 opt
dr-xr-xr-x 204 root root 0 Jan 31 23:35 proc
drwx----- 9 root root 4096 Feb 1 00:23 root
drwxr-xr-x 31 root root 800 Feb 1 00:26 run
lrwxrwxrwx 1 root root 8 Dec 27 20:28 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Dec 27 20:28 srv
dr-xr-xr-x 13 root root 0 Jan 31 23:35 sys
drwxrwxrwt 16 root root 4096 Feb 1 00:26 tmp
drwxr-xr-x 14 root root 4096 Dec 27 20:54 usr
drwxr-xr-x 12 root root 4096 Dec 27 22:57 var
lrwxrwxrwx 1 root root 30 Dec 27 20:31 vmlinuz -> boot/vmlinuz-6.5.0-kali3-amd64
lrwxrwxrwx 1 root root 30 Dec 27 20:31 vmlinuz.old -> boot/vmlinuz-6.5.0-kali3-amd64

(root@niraj)-[/]
#
```

## 5.Command – mkdir

<b>Name</b>	mkdir command –make directory
<b>Purpose</b>	Create the DIRECTORY(ies), if they do not already exist.
<b>Syntax</b>	<b>\$ mkdir</b> directory_name
<b>Example</b>	\$ mkdir images  \$ls

### OUTPUT:

```
user@user-H81M-S:~$ ls
'archive(1).zip'      'Exp 1.docx'
'archive(2)'          'EXP 1 SELAB'
'archive(2).zip'      Finding-missing-person-using-AI
archive.zip           Firefox_wallpaper.png
BullyElectionAlgorithm.class  jgroups-5.0.0.Final.jar
BullyElectionAlgorithm.java   ls1.png
Change                ls.png
chmod.png             MASM
Client.class          MSB.exe
Client.java           new.txt
user@user-H81M-S:~$ mkdir Art.txt
user@user-H81M-S:~$ ls
'archive(1).zip'      Client.java
'archive(2)'          'Exp 1.docx'
'archive(2).zip'      'EXP 1 SELAB'
archive.zip           Finding-missing-person-using-AI
Art.txt              Firefox_wallpaper.png
```

## 6.Command - chown

<b>Name</b>	chown
<b>Purpose</b>	To change owner, change the user and/or group ownership of each given File to a new Owner.
<b>Syntax</b>	chown [options] new_owner object(s)
<b>Example</b>	The following would transfer the ownership of a file named <i>file1</i> and a directory named <i>dir1</i> to a new owner named <i>alice</i> : chown root test.php

## OUTPUT:

```
(root@niraj)-[~]
# touch file.txt

(root@niraj)-[~]
# ls -l
total 4
drwxr-xr-x 3 root root 4096 Dec 31 01:07 SpamWa
-rw-r--r-- 1 root root  0 Feb  1 00:23 file.txt

(root@niraj)-[~]
# chown niraj file.txt

(root@niraj)-[~]
# ls -l
total 4
drwxr-xr-x 3 root  root 4096 Dec 31 01:07 SpamWa
-rw-r--r-- 1 niraj root  0 Feb  1 00:23 file.txt

(root@niraj)-[~]
#
```

## 7. Command - chgrp

Name	chgrp
Purpose	'chgrp' command changes the group ownership of each given File to Group (which can be either a group name or a numeric group id) or to match the same group as an existing reference file.
Syntax	chgrp [OPTION]... GROUP FILE...
Example	To Make oracleadmin the owner of the database directory \$ chgrp wheel .

## OUTPUT:

```
[bourbon@Shadowmere ~/test]$ chgrp wheel .
[bourbon@Shadowmere ~/test]$ ls -la
total 16
drwxr-xr-x  2 bourbon wheel  4096 Apr 30 14:52 .
drwx----- 51 bourbon bourbon 4096 Apr 30 14:52 ..
-rw-r--r--  1 bourbon bourbon   6 Apr 30 14:52 data.txt
-rw-----  1 root    bourbon 1016 Apr 15 20:33 test.php
```

## 8. Command - ps

Name	ps
Purpose	displays information about a selection of the active processes.
Syntax	ps aux
Example	\$ ps aux

## OUTPUT:

```
user@user-H81M-S:~$ ps
  PID TTY          TIME CMD
 3046 pts/0        00:00:00 bash
 4664 pts/0        00:00:00 ps
user@user-H81M-S:~$
```

## 9.Command – man

<b>Name</b>	man
<b>Purpose</b>	It is used to show the manual of the inputted command.
<b>Syntax</b>	\$ man <command_name>
<b>Example</b>	The inputting command will show the manual or all relevant information for the change directory command. \$ man cd

## OUTPUT:

```
SORT(1)                                User Commands
NAME
  sort - sort lines of text files

SYNOPSIS
  sort [OPTION]... [FILE]...
  sort [OPTION]... --files0-from=F

DESCRIPTION
  Write sorted concatenation of all FILE(s) to standard output.

  With no FILE, or when FILE is -, read standard input.

  Mandatory arguments to long options are mandatory for short options too.  Ordering options:

  -b, --ignore-leading-blanks
      ignore leading blanks

  -d, --dictionary-order
      consider only blanks and alphanumeric characters

  -f, --ignore-case
      fold lower case to upper case characters
```



## 10.Command – rm

<b>Name</b>	rm - remove file
<b>Purpose</b>	It is used to remove files from your Linux OS.
<b>Syntax</b>	rm filename.txt
<b>Example</b>	\$ rm tmp.txt

### OUTPUT:

```
user@user-H81M-S:~$ ls
add.txt                               Client.java
'archive(1).zip'                     'Exp 1.docx'
'archive(2)'                         'EXP 1 SELAB'
'archive(2).zip'                     Finding-missing-person-using-AI
archive.zip                           Firefox_wallpaper.png
Art.txt                              jgroups-5.0.0.Final.jar
BullyElectionAlgorithm.class          ls1.png
BullyElectionAlgorithm.java           ls.png
Change                                man.png
chmod.png                             MASM
Client.class                          mkdir.png
user@user-H81M-S:~$ rm add.txt
user@user-H81M-S:~$ ls
'archive(1).zip'                     'Exp 1.docx'
'archive(2)'                         'EXP 1 SELAB'
```

### CONCLUSION:-

Hence we have studied and implemented internal commands of Linux successfully.

### SIGN AND REMARK:

R1	R2	R3	R4	R5	Total	Signature
(3 Marks)	(3 Marks)	(3 Marks)	(3 Mark)	(3 Mark)	(15 Marks)	