<u>Unix Commands</u>

1) Ls command

ls command is most widely used command and it displays the contents of directory.

```
EG:
$ ls
j1     j2     j4     jay    jay1   k1     k2
```

ls -l will list all the file names, permissions, group, etc in long format.

```
$ ls -l
total 7
drwxr-xr-x  2 FOLDER1 Domain+Users     512 Jun 17 10:25 j1
drwxr-xr-x  2 FOLDER1 Domain+Users     512 Jun 17 10:27 j2
-rwxrwxrwx  1 FOLDER1 Domain+Users      62 Jun 22 10:31 j4
drwxr-xr-x  3 FOLDER1 Domain+Users     512 Jun 22 10:07 jay
-rw-r--r--  1 FOLDER1 Domain+Users      22 Jun 17 10:07 jay1
-rw-r--r--  1 FOLDER1 Domain+Users      41 Jun 17 10:49 k1
-rw-r--r--  1 FOLDER1 Domain+Users      41 Jun 17 10:54 k2
```

ls -a will list all the files including hidden files that start with .

```
$ ls -a
.           .sh_history   j2           jay           k1
..          j1            j4           jay1          k2
```

ls -lt will list all files names based on the time of creation, newer files bring first.

```
$ ls -it
1592694658 j4     3495130627 k2     936307168 j2      1889217990 jay1
3812471796 jay    1374746037 k1     923959549 j1
```

ls -Fxwill list files and directory names will be followed by slash.

```
$ ls -fx
x
```

```
x
x
x
x
x
x
$
```

ls -Rwill lists all the files and files in the all the directories, recursively

```
$ ls -R
j1     j2     j4     jay    jay1   k1     k2

./j1:
jay

./j2:
jay

./jay:
file1

./jay/file1:
```

ls -R | more will list all the files and files in all the directories, one page at a time

```
$ ls -R | more

j1
j2
j4
jay
jay1
k1
k2

./j1:
jay

./j2:
jay

./jay:
```

```
file1

./jay/file1:
```

2) Mkdir command

will create new directory

```
$ mkdir j3
$ ls
j1     j2     j3     j4     jay    jay1   k1     k2
```

3) Cd command

will change directory from current directory

```
$ cd jay
$ pwd
/home/FOLDER1/jay
```

4) Pwd command

pwd command will print your home directory on screen, pwd means print working directory.

```
$ pwd
/home/FOLDER1/jay
```

5) Cal command

cal command will print the calander on current month by default. If you want to print calander of august of 2011. That's seventh month of 2011.
cal 7 2011 will print following results.

```
$ cal 7 2011
     July 2011
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

6) Man command

is help command, and will explains you about online manual pages you can also use man in conjunction with any command to learn more about that command

```
$ man mkdir



mkdir(1)                                mkdir(1)


NAME
    mkdir - make directories

SYNOPSIS
    mkdir [ options ] directory ...

DESCRIPTION
    mkdir  creates  one  or  more directories. By default, the
    mode of created directories is a=rwx minus the bits set in
    the umask(1).

OPTIONS
    -m, --mode=mode
        Set  the  mode of created directories to mode. mode
        is symbolic or octal mode as in chmod(1).  Relative
        modes assume an initial mode of a=rwx.
……………………………………………
```

7) Clear command

clear command clears the screen and puts cursor at beginning of first line**.**

8) Rmdir command.

rmdir command will remove directory or directories if a directory is empty.

Options:

- rm -r directory_name will remove all files even if directory is not empty.
- rmdir sandeep is how you use it to remove sandeep directory.
- rmdir -p will remove directories and any parent directories that are empty.
- rmdir -s will suppress standard error messages caused by -p.
-

```
$ ls
j1    j2    j3    j4    jay    jay1   k1    k2
$ rmdir j3
$ ls
j1    j2    j4    jay    jay1   k1    k2
```

9) Banner command

banner prints characters in a sort of ascii art poster, for example to print wait in big

letters. I will type

```
$ banner jay

     #    ##     #   #
     #   #  #     # #
     #  #    #     #
     #  #######     #
 #    #  #    #     #
  #####   #    #     #
```

10) Mv command

mv command is used to move a file from one directory to another directory or to rename a file.

```
$ ls
j1    j2    j4    jay   k1    k2
$ mv j1 jay1
$ ls
j2     j4     jay    jay1   k1     k2
```

11) rm command

To delete files use rm command.

```
$ ls
j2     j4     jay    jay1   k2
$ rm k2
$ ls
j2     j4     jay    jay1
```

12) Cp command

cp command copies a file. If I want to copy a file named oldfile in a current

directory to a file named newfile in a current directory.

```
$ ls
j2    j4    j5    jay
$ cp j5 j6
$ ls
j2    j4    j5    j6    jay
$ cat j5
   this is j5 file
   Example of cat
  i m jay patel
        09bca32
$ cat j6
   this is j5 file
   Example of cat
  i m jay patel
        09bca32
```

13) Who command

who command displays information about the current status of system.

who *options file*
Who as default prints login names of users currently logged in.

Options

- -a use all options.
- -b Report information about last reboot.
- -d report expired processes.
- -H print headings.
- -p report previously spawned processes.
- -u report terminal usage.

```
$ who
FOLDER1          tty10   Jun 24 10:19
$ who -H
USER                LINE    LOGIN-TIME     FROM
FOLDER1          tty10   Jul 01 10:40
```

14) PS command

ps command is probably the most useful command for systems administrators. It reports information on active processes**.**

ps *options*

options.

- -a Lists all processes in system except processes not attached to terminals.
- -e Lists all processes in system.
- -f Lists a full listing.
- -j print process group ID and session ID.

```
$ ps
  PID TT       TIME COMMAND
  480 tty10       0 ps.exe
 2820 tty10       0 ksh.exe
$ ps -a
  PID TT       TIME COMMAND
  600 tty10       0 ps.exe
$ ps -e
  PID TT       TIME COMMAND
 1016 ?       1.00s oracle
  296 ?           0 AvastUI
 2252 ?           0 VProTray
 2408 ?           0 msmsgs
 2744 ?      19.00s explorer
 2748 tty10       0 ps.exe
 2756 ?      19.00s WINWORD
 2820 tty10       0 ksh.exe
 3024 ?           0 igfxpers
 3444 ?           0 igfxsrvc
 3792 ?           0 RTHDCPL
 3796 ?           0 igfxtray
 3824 ?           0 hkcmd
    1 ?           0 init
  584 ?           0 inetd.exe
 1328 ?           0 ums.exe
 3452 ?           0 at.svc
 3228 ?           0 at.svc
    4 ?       7.00s System
  276 ?           0 sqlservr
  512 ?           0 sqlservr
  588 ?       1.00s VProSvc
  676 ?           0 smss
  732 ?        4.00s csrss
  756 ?        8.00s winlogon
  800 ?            0 services
  812 ?            0 lsass
  984 ?            0 svchost
 1052 ?            0 svchost
 1132 ?            0 sqlwriter
 1148 ?        2.00s svchost
 1224 ?            0 wdfmgr
 1264 ?            0 svchost
 1372 ?            0 svchost
 1576 ?        4.00s AvastSvc
 1920 ?            0 spoolsv
 3812 ?            0 alg
$ ps -f
    USER   PID  PPID    START TT         TIME COMMAND
VTCBBC/0  2820     1 10:19:11 tty10         0 -ksh
VTCBBC/0  2828  2820 10:39:17 tty10         0 ps –f
$ ps -j
  PID  PGRP   SID TT         TIME COMMAND
 2820  2820  2820 tty10         0 ksh.exe
 2844  2844  2820 tty10         0 ps.exe
```

15) Cat Command

Used to create file.and edit into file.to display the contain of the file.

```
$ cat >file1
 this is file1


$ cat file1
 this is file1

$ cat >>file1
  Second line...
$ cat file1
 this is file1
  Second line...
```

-e
$ is printed at the end of each line. This option must be used with -v.
-s
Suppress messages pertaining to files that do not exist.
-t
Each tab will display as ^I and each form feed will display as ^L. This option must be used with -v.
-u
Output is printed as unbuffered.
-v
Display control characters and nonprinting characters

16) Tty command

Tty command will display your terminal. Syntax is
tty *options*

```
$ tty
/dev/tty10
```

Options

- -l will print the synchronous line number.
- -s will return only the codes: 0 (a terminal), 1 (not a terminal), 2 (invalid options) (good for scripts)

17) Date command

Date displays todays date, to use it type date at prompt.

```
$ date

Fri Jul  1 10:43:51 IST 2011
```

-a Slowly adjust the time by sss.fff seconds (fff represents fractions of a second). This adjustment can be positive or negative. The system's clock will be sped up or slowed down until it has drifted by the number of seconds specified. Only the super-user may adjust the time.

-u Display (or set) the date in Greenwich Mean Time (GMT-universal time), bypassing the normal conversion to (or from) local time.

```
$ date -u

Fri Jul  1 05:12:44 GMT 2011
```

-s datestr Sets the time and date to the value specfied in the *datestr*. The datestr may contain the month names, timezones, 'am', 'pm', etc. See examples for an example of how the date and time can be set.

```
$ date -s 'am'
Fri Jul  1 10:40:53 IST 2011
$ date -s 'pm'
Fri Jul  1 22:41:05 IST 2011
```

```
$ date '+DATE: %m/%d/%y%nTIME:%H:%M:%S'

DATE: 07/01/11

TIME:10:40:09
```

18) uname command

Print System name

-a
Print basic information currently available from the system.

-i
Print the name of the hardware implementation (platform).

-m
Print the machine hardware name (class). Use of this option is discouraged; use uname -p instead.

-n
Print the nodename (the nodename is the name by which the system is known to a communications network).

-p
Print the current host's ISA or processor type.

-r
Print the operating system release level.

-s
Print the name of the operating system. This is the default.

-v
Print the operating system version.

-X
Print expanded system information, one information element per line, as expected by SCO Unix. The displayed information includes:

- system name, node, release, version, machine, and number of CPUs.
- BusType, Serial, and Users (set to "unknown" in Solaris)

- OEM# and Origin# (set to 0 and 1, respectively)

-S systemname
The nodename may be changed by specifying a system name argument. The system name argument is restricted to SYS_NMLN characters. SYS_NMLN is an implementation specific value defined in <sys/utsname.h>. Only the super-user is allowed this capability.

```
$ uname
UWIN-NT

$ uname -p
i386

$ uname -s
UWIN-NT

$ uname -a
UWIN-NT lab22 (22062001)-5.12600 2600 i686

$ uname -i
2001a8c0

$ uname -n
lab22

$ uname -p
i386
```

19) who am I command

Tells you your user information

```
$ who am i
FOLDER1          tty10   Jun 24 10:19
```

20) echo Command

Print the message on the screen

```
$ echo jay
jay
```

21) Chmod command

chmod command is used to change permissions on a file.

<u>Permissions</u>
u - User who owns the file.
g - Group that owns the file.
o - Other.
a - All.
r - Read the file.
w - Write or edit the file.
x - Execute or run the file as a program.

```
$ chmod 777 file1
$ ls -l file1
-rwxrwxrwx  1 FOLDER1 Domain+Users      32 Jun 24 10:45 file1
```

22) cut

-c list   The list following -c specifies character positions (for instance, -c1-72 would pass the first 72 characters of each line).

-f list   The list following -f is a list of fields assumed to be separated in the file by a delimiter character (see -d); for instance, -f1,7 copies the first and seventh field only. Lines with no field delimiters will be passed through intact (useful for table subheadings), unless -s is specified. If -f is used, the input line should contain 1023 characters or less.

```
$ cat >jay.txt
01 | Jay Patel | Bardoli | 9687748100
02 | Hardip Patel | Bardoli |9958562457
03 | Dhaval Mistry | Bardoli | 9954781236
04 | Piyush Mistry | Bardoli | 9856247851

$ cut -d "|" -f 1,4 jay.txt
01 | 9687748100
02 |9958562457
03 | 9954781236
04 | 9856247851
```

```
$ cut -d"|" -c 1-10 jay.txt
01 | Jay P
02 | Hardi
03 | Dhava
04 | Piyus
```

23)sort

-To sort the file Data

b
Ignores spaces at beginning of the line.
-d
Uses dictionary sort order and ignores the punctuation.
-f
Ignores caps
-i
Ignores nonprinting control characters.
-m
Merges two or more input files into one sorted output.
-M
Treats the first three letters in the line as a month (such as may.)
-n
Sorts by the beginning of the number at the beginning of the line.
-r
Sorts in reverse order
-u
If line is duplicated only display once
+fields
Sorts by fields , usually by tabs
filename
The name of the file that needs to be sorted.

-o outputfile
Sends the sorted output to a file

```
$ cat jay.txt
01 | Jay Patel | Bardoli | 9687748100
02 | Hardip Patel | Bardoli |9958562457
03 | Dhaval Mistry | Bardoli | 9954781236
04 | Piyush Mistry | Bardoli | 9856247851
07 | Akshay Tailor | Surat | 9856235474
05 | Sanny Gajjar | Surat | 9658745236
06 | Washim Sheikh | Kadod | 9875462314
```

```
$ sort jay.txt
01 | Jay Patel | Bardoli | 9687748100
02 | Hardip Patel | Bardoli |9958562457
03 | Dhaval Mistry | Bardoli | 9954781236
04 | Piyush Mistry | Bardoli | 9856247851
05 | Sanny Gajjar | Surat | 9658745236
06 | Washim Sheikh | Kadod | 9875462314
07 | Akshay Tailor | Surat | 9856235474
$ sort -r jay.txt
07 | Akshay Tailor | Surat | 9856235474
06 | Washim Sheikh | Kadod | 9875462314
05 | Sanny Gajjar | Surat | 9658745236
04 | Piyush Mistry | Bardoli | 9856247851
03 | Dhaval Mistry | Bardoli | 9954781236
02 | Hardip Patel | Bardoli |9958562457
01 | Jay Patel | Bardoli | 9687748100
```

24) head

Displays the first n lines of a file, unless otherwise stated.

```
$ head -n 1 jay.txt

01 | Jay Patel | Bardoli | 9687748100
```

25) tail

Displays the last n lines of a file, unless otherwise stated.

```
$ tail -n 2 jay.txt

05 | Sanny Gajjar | Surat | 9658745236

06 | Washim Sheikh | Kadod | 9875462314
```

26) vi

This command starts the vi text editor. To edit a file named `myfile` in the current directory, enter:

```
 $vi f_name.sh
```