

## BASIC UNIX COMMANDS

**Ex.No: 1**

**Date :03.03.2021**

### Aim

To understand the basic commands used to work with Unix environment.

### General Command Structure

#### Syntax:

Command [-options] arguments

#### Where,

Command name of the command

Arguments file name, user name or some other information that the program needs.

Options which is a option for a program. So it is put inside square brackets.

## BASIC COMMANDS

1.Command : **who**

Purpose : It is used to get the information about all the users currently working in the system.

Syntax : who

Example : \$ who

2.Command : **who am i**

Purpose : It is used to know in which terminal the user is currently logged on.

Syntax : who am i

Example : \$ who am I

3.Command : **date**

Purpose : It is used to display the system date and time.

Syntax : date

Example : \$ date

4.Command : **cal**

Purpose : It prints the calender for the specified year and month.

Syntax : cal<month><year>

Example : \$ cal 05 2003

5.Command : **id**

Purpose : It is used to display the login name.

Syntax : id

Example : \$ id

- 6.Command : **clear**  
 Purpose : It is used to clear the screen.  
 Syntax : clear  
 Example : \$ clear
- 7.Command : **uname**  
 Purpose : It is used to display the details about the OS in which we are working.  
 Syntax : uname [options]  
 Example : \$ uname -n
- 8.Command : **tty**  
 Purpose : It is used to know the terminal name on which we work.  
 Syntax : tty  
 Example : \$ tty
- 9.Command : **pwd**  
 Purpose : It is used to display the absolute pathname of current working directory.  
 Syntax : pwd  
 Example : \$ pwd
- 10.Command : **bc**  
 Purpose : It is used to perform simple mathematical calculations.  
 Syntax : bc filename  
 Example : \$ bcpp
- 11.Command : **echo**  
 Purpose : It echoes the argument on the standard output textce.  
 Syntax : echo [options] <string>  
 Example : \$ echo 'BOOM'
- 12.Command : **man**  
 Purpose : It gives details about the unix commands.  
 Syntax : man < command name >  
 Example : \$ man echo

## FILE MANIPULATION COMMANDS

- 1.Command : **cat**  
 Purpose : It is used to create a new file  
 Syntax : cat ><file name>  
 Example : \$ cat >  
 This is sample File in Unix  
 Ctrl – d

To append the content of already existing file.

Example : \$ cat>>

It is also used to display the contents of the fileas well as used to create a new file.

Syntax : cat <file> name >  
Example : \$ cat  
This is sample File in Unix

To display the contents of two or more files, specify the filenames with the cat commands separated by a space in between them.

Example:

```
$cat 1 2 3  
1file  
2file  
3file
```

2.Command : **ls**  
Purpose : It is used to display the files in the current working directory.  
Syntax : ls [options] <arguments>  
Options  
a - to list all directory entries  
d - to list name of directories  
l - to list files in long form  
r - to list file in the reverse order  
t - to list the files sorted by time  
lrl - list content of current directory. Output will be sorted based on modification date & time.

Example : \$ ls -l

3.Command : **tail**  
Purpose : It is used to Prints the last several lines of the specified files.  
Syntax : tail [options] <file name>  
Example : \$ tail -5 text

4.Command : **head**  
Purpose : It is used to display the top portion of the file where top portion represents the no'sof lines.  
Syntax : head [options] <file name>  
Example : \$ head -5 text

5.Command : **cmp**  
Purpose : Compare two files, and if they differ, tells the first byte and line number where they differ.  
Syntax : cmp file1 file2  
Example : \$ cmp a1 a2

6.Command : **diff**  
Purpose : The diff command analyses line by line and displays a list of changes between two files.

Syntax :diff file1 file2  
Example :\$ diff a1 a2

#### 7.Command :wc

Purpose : It is used to count the number of lines,words and characters in a file or group of files.

Syntax :wc [options] <file name>  
Example : \$ wc .txt  
2 15 95 .txt

The options used with **wc** commands are listed.

Option	Description
-c	Count number of characters in the file
-w	Counts the number of words in the file
-l	Counts number of lines in the file

Example:

```
$ wc -c .txt  
95 .txt  
$ wc -w .txt  
15 .txt  
$ wc -l .txt  
2 .txt
```

#### 8.Command :sort

Purpose : Sorts the specified files. The command has many useful arguments..

Syntax :sort [options] <file name>

Option-r reverse

Example : \$ sort .txt

#### 9.Command :pr

Purpose : It is used to display the contents of the file by separating them into pages and each page begins with the header information.

Syntax : pr [options] <file name>

Example : \$ pr

#### 10.Command :

**cut**

Purpose : It is used to extract selected fields or columns from each line of one or more files and display them on the standard output textce.

Syntax : cut [options] <file name >

Example : \$ cut -c5

#### 11.Command :paste

Purpose : It concatenates the line from each input file column by column with tab characters in between them.

Syntax : paste [options] <file name>

Example : \$ paste f1 f2

**12.Command :join**

Purpose : It is used to extracts common lines from two sorted files and there should be the common field in both file.

Syntax : join [options] <file name1 ><file name 2>

Example : \$ join -a1 f1 f2

**13.Command :uniq**

Purpose : It compares adjacent lines in the file and displays the output by eliminating duplicate adjacent lines in it.

Syntax :uniq [options] <file name >

Example :\$ uniq -c text

**14.Command :nl**

Purpose : It is used to add the line numbers to the file.

Syntax :nl [options] [filename]

Example : \$ nl text

**15.Command :tr**

Purpose : It is used to translate or delete a character or a string from the standard input to produce the required output.

Syntax :tr [options] <string1><string2>

Example : \$ tr -s 'a' 'b' < text

**16.Command :tee**

Purpose : It is used to read the contents from standardinput or from output of another command and reproduces the output to both in standardoutput and direct into output to one or more files.

Syntax :command | tee [options] <file name >

Example : \$ date | tee dat.txt

**17.Command :grep**

Purpose : It is used to search the specified pattern from one or more files.

Syntax :grep [options] <pattern><file name >

Example : \$ grep welcometext

**DIRECTORY MANUPULATION COMMANDS****1.Command :mkdir**

Purpose : It is used to create new directory or more than one directory.

Syntax :mkdir<directory name >

Example :\$ mkdirsudhan

**2.Command :cd**

Purpose : It is used to change the control from oneworking directory to another specified directory.

Syntax :cd <directory name >

Example : \$ cd sudhan

3.Command :rmdir  
Purpose : It is used to remove the directory if it is empty.  
Syntax :rmdir<directoryname>  
Example : \$ rmdir sudhan

You can use the -r(recursive) option with the rmdir command so that it deletes the directory even when it is not empty.  
For example, \$rmdir -r haran

4.Command :cp  
Purpose : It is used to copy one or more files.  
Syntax :cp<source file name ><destination file name>  
Example : \$ cp texttext1

5.Command :  
Purpose : It is used to control the information on the screen from a line to a screen full.  
Syntax : more options <file name >  
Example : \$ more

This command displays one screen full of information from the file. To display the next screen press enter key or space bar. To quit press Q.

6.Command :passwd  
Example : passwd user  
Changes the current user's password, or that of the specified user (requires root privileges).  
The command prompts for the new password.

7.Command : mv  
Purpose : It is used to move a file within a directory with different names and also used to move a file to different directory with its original name.  
Syntax :mv<source file name ><destination directory name>  
Example : \$ mv texttext2

## 8.chmod command

You set the access modes of a directory or file by using the chmod command, which has the following pattern:

chmod  
*nnn directory-or-file*

The argument *nnn* is a three-digit number, which gives the access mode for the owner, group, and other users.

For example, the argument 751 is equivalent to rwxr-x--x, which gives the owner every possible permission, gives the group read and execute permission, and gives other users execute permission.

9.Command : write

Purpose : the write command can be used by any user to write something on someone else's terminal, provided the recipient of the message permits communication

Syntax : write <user name>

Example : \$ write user2

This is sample message.

Ctrl d

On executing this command the message would be relayed to the user whose login name is user2. He would hear a beep on his terminal, followed by the message.

There are two prerequisites for a smooth write operation:

1.The recipient must be logged in; else an error message is inevitable.

2.The recipient must have given permission for messages to reach his or her terminal.

## 10.wall command

Prints a message to each user except those who've disabled message reception.Type **Ctrl-D** to end the message.

11.Command : news

Purpose : The system administrator is the sole person who can make news under the Unix OS. He types the information which he wants everyone on the network to know of in different files in /usr/news directory.

Syntax : 1) \$news filename  
2) \$news [-options]

Options : 1) -n option only lists the names of the news items from the /usr/news directory that have not yet been read by you.  
2) -s option which provides a count of the unread new items in the /usr/news directory.  
3) -a shows all the news.

Example 1 : \$ news sample

Where sample is the file name in which the news items are available.

Example 2 : \$news -s

It provides the count of the unread new items.

## Output:

```
#####
# webminal.org - your linux ~ #
#####

- Share files with others, See /common_pool/README.txt
- See 'Root' menu for Webminal Desktop Root and Webminal Root features
- For Teachers/Students, partial sudo (plus C programming) platform available - mail us.

[pavithra2002@webminal.org ~]$who
[pavithra2002@webminal.org ~]$whoami
pavithra2002
[pavithra2002@webminal.org ~]$date
Wed Mar 3 15:29:40 CET 2021
[pavithra2002@webminal.org ~]$cal
      March 2021
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
```

```
[pavithra2002@webminal.org ~]$id
uid=246056(pavithra2002) gid=246115(pavithra2002) groups=246115(pavithra2002) context=guest_u:guest_r:gue
st_t:s0
[pavithra2002@webminal.org ~]$uname
Linux
[pavithra2002@webminal.org ~]$tty
/dev/pts/129
[pavithra2002@webminal.org ~]$pwd
/home/pavithra2002
[pavithra2002@webminal.org ~]$bc
-sh: bc: command not found
[pavithra2002@webminal.org ~]$echo "os"
os
[pavithra2002@webminal.org ~]$
```

```
[pavithra2002@webminal.org ~]$pwd  
/home/pavithra2002  
[pavithra2002@webminal.org ~]$bc  
-sh: bc: command not found  
[pavithra2002@webminal.org ~]$echo "os"  
os  
[pavithra2002@webminal.org ~]$man echo  
[pavithra2002@webminal.org ~]$cat > fruit  
apple  
orange  
grapes  
[pavithra2002@webminal.org ~]$cat fruit  
apple  
orange  
grapes  
[pavithra2002@webminal.org ~]$ls  
colour.txt count.txt file fruit fruits.txt pavi software sofware subject.c  
[pavithra2002@webminal.org ~]$cat fruit  
apple  
orange  
grapes  
[pavithra2002@webminal.org ~]$cat > vegetable  
beans  
carrot  
potato  
[pavithra2002@webminal.org ~]$cat vegetable  
beans  
carrot  
potato  
[pavithra2002@webminal.org ~]$cat fruit vegetable  
apple  
orange  
grapes  
beans  
carrot
```

```
[pavithra2002@webminal.org ~]$ls
colour.txt count.txt file fruit fruits.txt pavi software sofware subject.c vegetable
[pavithra2002@webminal.org ~]$ls -a
. .bash_logout colour.txt file .magic_string.txt sofware .zshrc
.. .bash_profile count.txt fruit pavi subject.c
.bash_history .bashrc .emacs fruits.txt software vegetable
[pavithra2002@webminal.org ~]$ls -d
.
.
[pavithra2002@webminal.org ~]$ls -1
colour.txt
count.txt
file
fruit
fruits.txt
pavi
software
software
subject.c
vegetable
[pavithra2002@webminal.org ~]$ls -l
total 32
-rw-rw-r--. 1 pavithra2002 pavithra2002 17 Mar 3 15:08 colour.txt
-rw-rw-r--. 1 pavithra2002 pavithra2002 200 Mar 3 15:08 count.txt
-rw-rw-r--. 1 pavithra2002 pavithra2002 32 Mar 3 14:41 file
-rw-rw-r--. 1 pavithra2002 pavithra2002 20 Mar 3 15:41 fruit
-rw-rw-r--. 1 pavithra2002 pavithra2002 19 Mar 3 15:07 fruits.txt
-rw-rw-r--. 1 pavithra2002 pavithra2002 13 Feb 27 06:36 pavi
drwxrwxr-x. 2 pavithra2002 pavithra2002 6 Mar 3 13:59 software
drwxrwxr-x. 2 pavithra2002 pavithra2002 6 Mar 3 14:25 software
-rw-rw-r--. 1 pavithra2002 pavithra2002 11 Feb 27 06:33 subject.c
-rw-rw-r--. 1 pavithra2002 pavithra2002 20 Mar 3 15:42 vegetable
[pavithra2002@webminal.org ~]$ls -r
vegetable subject.c software software pavi fruits.txt fruit file count.txt colour.txt
[pavithra2002@webminal.org ~]$ls -t
vegetable fruit count.txt colour.txt fruits.txt file sofware software pavi subject.c
```

```
[pavithra2002@webminal.org ~]$ls -lrt
total 32
-rw-rw-r--. 1 pavithra2002 pavithra2002 11 Feb 27 06:33 subject.c
-rw-rw-r--. 1 pavithra2002 pavithra2002 13 Feb 27 06:36 pavi
drwxrwxr-x. 2 pavithra2002 pavithra2002 6 Mar 3 13:59 software
drwxrwxr-x. 2 pavithra2002 pavithra2002 6 Mar 3 14:25 sofware
-rw-rw-r--. 1 pavithra2002 pavithra2002 32 Mar 3 14:41 file
-rw-rw-r--. 1 pavithra2002 pavithra2002 19 Mar 3 15:07 fruits.txt
-rw-rw-r--. 1 pavithra2002 pavithra2002 17 Mar 3 15:08 colour.txt
-rw-rw-r--. 1 pavithra2002 pavithra2002 200 Mar 3 15:08 count.txt
-rw-rw-r--. 1 pavithra2002 pavithra2002 20 Mar 3 15:41 fruit
-rw-rw-r--. 1 pavithra2002 pavithra2002 20 Mar 3 15:42 vegetable
[pavithra2002@webminal.org ~]$tail fruit
apple
orange
grapes
[pavithra2002@webminal.org ~]$head vegetable
beans
carrot
potato
[pavithra2002@webminal.org ~]$cmp fruit vegetable
fruit vegetable differ: byte 1, line 1
[pavithra2002@webminal.org ~]$diff fruit vegetable
1,3c1,3
< apple
< orange
< grapes
---
> beans
> carrot
> potato
```

```
[pavithra2002@webminal.org ~]$cat > prg.c
#include<stdio.h>
{
}
[pavithra2002@webminal.org ~]$wc prg.c
3 3 22 prg.c
[pavithra2002@webminal.org ~]$wc -c prg.cc
wc: prg.cc: No such file or directory
[pavithra2002@webminal.org ~]$wc -c prg.c
22 prg.c
[pavithra2002@webminal.org ~]$wc -l prg.c
wc: invalid option -- '1'
Try 'wc --help' for more information.
[pavithra2002@webminal.org ~]$wc -l prg.c
3 prg.c
[pavithra2002@webminal.org ~]$sort fruit
apple
grapes
orange
[pavithra2002@webminal.org ~]$pr vegetable
```

2021-03-03 15:42

vegetable

Page 1

beans

```
[pavithra2002@webminal.org ~]$cut -c2 fruit
p
r
r
[pavithra2002@webminal.org ~]$paste fruit vegetable
apple    beans
orange   carrot
grapes   potato
[pavithra2002@webminal.org ~]$join -a1 fruit vegetable
apple
orange
join: fruit:3: is not sorted: grapes
grapes
[pavithra2002@webminal.org ~]$uniq -c vegetable
    1 beans
    1 carrot
    1 potato
[pavithra2002@webminal.org ~]$nl fruit
    1 apple
    2 orange
    3 grapes
[pavithra2002@webminal.org ~]$tr -s 's' 'u'
sort
uort
[pavithra2002@webminal.org ~]$date |tee fruit
Wed Mar  3 16:00:34 CET 2021
[pavithra2002@webminal.org ~]$grep apple fruit
[pavithra2002@webminal.org ~]$uort
-sh: uort: command not found
[pavithra2002@webminal.org ~]$grep apple fruit
[pavithra2002@webminal.org ~]$greps apple fruit
-sh: greps: command not found
```

```
[pavithra2002@webminal.org ~]$grep apple fruit
[pavithra2002@webminal.org ~]$greps apple fruit
-sh: greps: command not found
[pavithra2002@webminal.org ~]$date |tee fruit
Wed Mar  3 16:01:50 CET 2021
[pavithra2002@webminal.org ~]$grep apple fruit
[pavithra2002@webminal.org ~]$mkdir dir1
[pavithra2002@webminal.org ~]$cd dir1
[pavithra2002@webminal.org dir1]$rmdir dir1
rmdir: failed to remove 'dir1': No such file or directory
[pavithra2002@webminal.org dir1]$mkdir dir2
[pavithra2002@webminal.org dir1]$cat > fruit
mango
morishes
[pavithra2002@webminal.org dir1]$cat > vegetable
tomato
pumpkin
[pavithra2002@webminal.org dir1]$cp fruit vegetable
[pavithra2002@webminal.org dir1]$cp fruit vegetable
```

```
[pavithra2002@webminal.org dir1]$more
Usage: more [options] file...

Options:
 -d      display help instead of ring bell
 -f      count logical, rather than screen lines
 -l      suppress pause after form feed
 -p      do not scroll, clean screen and display text
 -c      do not scroll, display text and clean line ends
 -u      suppress underlining
 -s      squeeze multiple blank lines into one
 -NUM    specify the number of lines per screenful
 +NUM    display file beginning from line number NUM
 +/STRING display file beginning from search string match
 -V      output version information and exit
[pavithra2002@webminal.org dir1]$passwd user
-sh: /bin/passwd: Permission denied
[pavithra2002@webminal.org dir1]$mv fruit vegetable
[pavithra2002@webminal.org dir1]$chmod
chmod: missing operand
Try 'chmod --help' for more information.
[pavithra2002@webminal.org dir1]$chmod 754 dir1
chmod: cannot access 'dir1': No such file or directory
[pavithra2002@webminal.org dir1]$mkdir dir2
mkdir: cannot create directory 'dir2': File exists
[pavithra2002@webminal.org dir1]$mkdir dir3
[pavithra2002@webminal.org dir1]$hmod 754 dir3
-sh: hmod: command not found
[pavithra2002@webminal.org dir1]$write pavithra
write: pavithra is not logged in
[pavithra2002@webminal.org dir1]$wall
oops
[pavithra2002@webminal.org dir1]$news ss
-sh: news: command not found
```

```
[pavithra2002@webminal.org dir1]$wall
oops
[pavithra2002@webminal.org dir1]$news ss
-sh: news: command not found
[pavithra2002@webminal.org dir1]$news -n
-sh: news: command not found
[pavithra2002@webminal.org dir1]$news -n ss
-sh: news: command not found
[pavithra2002@webminal.org dir1]$news -a
-sh: news: command not found
[pavithra2002@webminal.org dir1]$
```

<b>Observation(20)</b>	
<b>Record(5)</b>	
<b>Total(25)</b>	
<b>Initial</b>	

**Result:**

Thus the basic Unix commands were executed and outputs were noted.

