

EX NO:1

BASIC UNIX COMMANDS

DATE:27/02/2021

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.

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 – NDS:

To append the content of already existing file.

Example : \$ cat>>

It is also used to display the contents of the files as 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

1 file

2 file

3 file

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

lrt - 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's of 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

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<directory name >

Example : \$ rmdirsudhan

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 : \$ cptexttext1

5. Command : more

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 CtrlD 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.

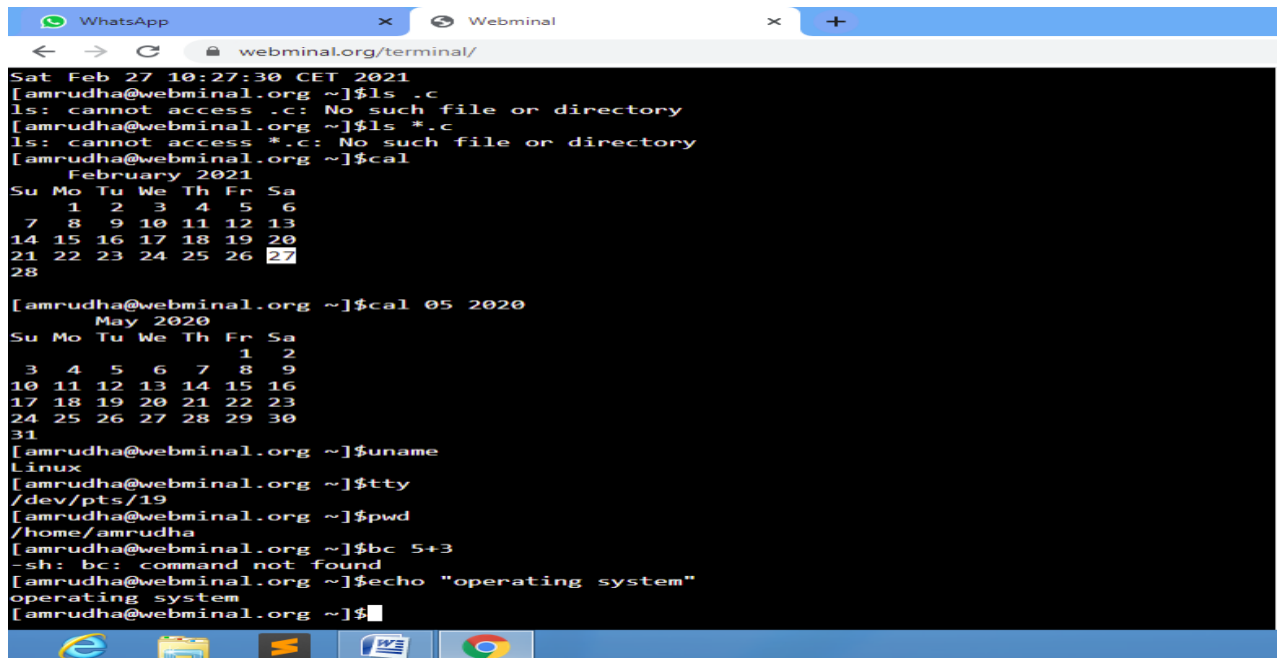
Example 1 : \$ news sample

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:



```
Sat Feb 27 10:27:30 CET 2021
[amrudha@webminal.org ~]$ls .c
ls: cannot access .c: No such file or directory
[amrudha@webminal.org ~]$ls *.c
ls: cannot access *.c: No such file or directory
[amrudha@webminal.org ~]$cal
    February 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

[amrudha@webminal.org ~]$cal 05 2020
    May 2020
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

[amrudha@webminal.org ~]$uname
Linux
[amrudha@webminal.org ~]$tty
/dev/pts/19
[amrudha@webminal.org ~]$pwd
/home/amrudha
[amrudha@webminal.org ~]$bc 5+3
-sh: bc: command not found
[amrudha@webminal.org ~]$echo "operating system"
operating system
[amrudha@webminal.org ~]$
```

```
WhatsApp x Webminal x +
webminal.org/terminal/

[amrudha@webminal.org ~]$cat > subject
OS
DAA
CA
SE
DBMS
PQT
^C
[amrudha@webminal.org ~]$cat subject
OS
DAA
CA
SE
DBMS
PQT
[amrudha@webminal.org ~]$cat >> subject
oops
^C
[amrudha@webminal.org ~]$cat >> subject
OOPS
^C
[amrudha@webminal.org ~]$cat subject
OS
DAA
CA
SE
DBMS
PQT
oops
OOPS
[amrudha@webminal.org ~]$
```

```
WhatsApp x Webminal x +
webminal.org/terminal/

LS(1) User Commands LS(1)
NAME
ls - list directory contents
SYNOPSIS
ls [OPTION]... [FILE]...
DESCRIPTION
List information about the FILES (the current directory by default). Sort
entries alphabetically if none of -cftuvSUX nor --sort is specified.

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

-a, --all
do not ignore entries starting with .
-A, --almost-all
do not list implied . and ..
--author
with -l, print the author of each file
-b, --escape
print C-style escapes for nongraphic characters
--block-size=SIZE
scale sizes by SIZE before printing them; e.g., '--block-size=M'
prints sizes in units of 1,048,576 bytes; see SIZE format below
-B, --ignore-backups
do not list implied entries ending with ~

Manual page ls(1) line 1 (press h for help or q to quit)
```



```
WhatsApp x Webminal x +
webminal.org/terminal/

[amrudha@webminal.org ~]$ls -l
total 36
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:12 c1
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:13 c2
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:13 c3
-rw-rw-r-- 1 amrudha amrudha 29 Feb 27 15:18 currentdate
-rw-rw-r-- 1 amrudha amrudha 22 Feb 27 14:42 file
-rw-rw-r-- 1 amrudha amrudha 19 Feb 27 15:16 fruits
-rw-rw-r-- 1 amrudha amrudha 1 Feb 27 15:19 previousdate
-rw-rw-r-- 1 amrudha amrudha 22 Feb 27 15:11 subject
-rw-rw-r-- 1 amrudha amrudha 0 Feb 27 06:11 subjecty
-rw-rw-r-- 1 amrudha amrudha 34 Feb 27 15:08 word.c
[amrudha@webminal.org ~]$ls -t
previousdate currentdate fruits c3 c2 c1 subject word.c file subjecty
[amrudha@webminal.org ~]$ls -r
word.c subjecty subject previousdate fruits file currentdate c3 c2 c1
[amrudha@webminal.org ~]$ls -lrt
total 36
-rw-rw-r-- 1 amrudha amrudha 0 Feb 27 06:11 subjecty
-rw-rw-r-- 1 amrudha amrudha 22 Feb 27 14:42 file
-rw-rw-r-- 1 amrudha amrudha 34 Feb 27 15:08 word.c
-rw-rw-r-- 1 amrudha amrudha 22 Feb 27 15:11 subject
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:12 c1
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:13 c2
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:13 c3
-rw-rw-r-- 1 amrudha amrudha 19 Feb 27 15:16 fruits
-rw-rw-r-- 1 amrudha amrudha 29 Feb 27 15:18 currentdate
-rw-rw-r-- 1 amrudha amrudha 1 Feb 27 15:19 previousdate
[amrudha@webminal.org ~]$man ls
[amrudha@webminal.org ~]$man ls
[amrudha@webminal.org ~]$
```

```
WhatsApp x Webminal x +
webminal.org/terminal/

[amrudha@webminal.org ~]$cat > subject
OS
DAA
CA
SE
DBMS
PQT
^C
[amrudha@webminal.org ~]$cat subject
OS
DAA
CA
SE
DBMS
PQT
[amrudha@webminal.org ~]$cat >> subject
[amrudha@webminal.org ~]$cat subject
hi
[amrudha@webminal.org ~]$cat > word.c
#include<stdio.h>
void main()
{
}
^C
[amrudha@webminal.org ~]$ls word.c
word.c
[amrudha@webminal.org ~]$wc word.c
 4  5 34 word.c
[amrudha@webminal.org ~]$wc -c word.c
34 word.c
[amrudha@webminal.org ~]$wc -w word.c
5 word.c
[amrudha@webminal.org ~]$wc -l word.c
wc: invalid option -- 'l'
Try 'wc --help' for more information
```

```
WhatsApp Webminal
webminal.org/terminal/
-rw-rw-r-- 1 amrudha amrudha 34 Feb 27 15:08 word.c
[amrudha@webminal.org ~]$ls -t
previousdate currentdate fruits c3 c2 c1 subject word.c file subjecty
[amrudha@webminal.org ~]$ls -r
word.c subjecty subject previousdate fruits file currentdate c3 c2 c1
[amrudha@webminal.org ~]$ls -lrt
total 36
-rw-rw-r-- 1 amrudha amrudha 0 Feb 27 06:11 subjecty
-rw-rw-r-- 1 amrudha amrudha 22 Feb 27 14:42 file
-rw-rw-r-- 1 amrudha amrudha 34 Feb 27 15:08 word.c
-rw-rw-r-- 1 amrudha amrudha 22 Feb 27 15:11 subject
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:12 c1
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:13 c2
-rw-rw-r-- 1 amrudha amrudha 4 Feb 27 15:13 c3
-rw-rw-r-- 1 amrudha amrudha 19 Feb 27 15:16 fruits
-rw-rw-r-- 1 amrudha amrudha 29 Feb 27 15:18 currentdate
-rw-rw-r-- 1 amrudha amrudha 1 Feb 27 15:19 previousdate
[amrudha@webminal.org ~]$man ls
[amrudha@webminal.org ~]$man ls
[amrudha@webminal.org ~]$cat > birds
pегion
parrot
sparrow
dove
^C
[amrudha@webminal.org ~]$diff fruits birds
1,3c1,4
< apple
< orange
< mango
---
> pегion
> parrot
> sparrow
> dove
```

```
WhatsApp Webminal
webminal.org/terminal/
[amrudha@webminal.org indirect]$rmdir indirectname
rmdir: failed to remove 'indirectname': No such file or directory
[amrudha@webminal.org indirect]$mkdir neutral
[amrudha@webminal.org indirect]$ls
neutral
[amrudha@webminal.org indirect]$cd neutral
[amrudha@webminal.org neutral]$ls
[amrudha@webminal.org neutral]$ls
[amrudha@webminal.org neutral]$cd neutral
-sh: cd: neutral: No such file or directory
[amrudha@webminal.org neutral]$cd..
-sh: cd.: command not found
[amrudha@webminal.org neutral]$ls
[amrudha@webminal.org neutral]$ls
[amrudha@webminal.org neutral]$cd ..
[amrudha@webminal.org indirect]$ls
neutral
[amrudha@webminal.org indirect]$ls -d
.
[amrudha@webminal.org indirect]$ls -D
neutral
[amrudha@webminal.org indirect]$touch {a..d}
[amrudha@webminal.org indirect]$touch {a..d} {1..4}
[amrudha@webminal.org indirect]$ls
1 2 3 4 a b c d neutral
[amrudha@webminal.org indirect]$touch {r..x-1..7}
[amrudha@webminal.org indirect]$ls
1 2 3 4 a b c d neutral {r..x-1..7}
[amrudha@webminal.org indirect]$rmdir neutral
[amrudha@webminal.org indirect]$ls
1 2 3 4 a b c d {r..x-1..7}
[amrudha@webminal.org indirect]$cp neutral indirect
cp: cannot stat 'neutral': No such file or directory
[amrudha@webminal.org indirect]$ls
1 2 3 4 a b c d {r..x-1..7}
```

Observation(20)	
Record(5)	
Total(25)	
Initial	

Result:

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