VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY
Permanently Affiliated to JNTU Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified
Nambur, Pedakakani (M), Guntur (Dt) - 522508
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
B.Tech Program is Accredited by NBA

CERTIFICATE

Name of the Lab : OPERATING SYSTEMS
Name of the Student : NAGABABU THOTA

Student Regd. No. : 18BQ1A05K3
CLASS : III B.TECH. I SEM CSE – D
GIT HUB LINK:
https://github.com/nagababuthota984/5K3-OS-LAB

INDEX

AIM: Study of Unix/Linux general purpose utility command list
man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown.

DESCRIPTION:

man : used to display the user manual of any **command** that we can run on the terminal. It provides a detailed view of the **command** which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS

who: used to find out the following information :

1. Time of last system boot
2. Current run level of the system.
3. List of logged in users and more.

The who command is used to get information about currently logged in user on to system.

cat: cat command is used to display the contents onto the screen from an input file or from standard input
cd: cd means change directory .cd command is used to switch between directories .
cp: cp command is used to copy the contents from one location to another location.
ps: ps is short for Process Status, is a command line utility that is used to display or view information related to the processes running in a linux system.
ls: ls command is used to list the files and directories present in the current directory passed as argument.
mv: mv is a command utility in unix environment which moves the contents from one location to another.
rm: rm means remove .It is the command utility used for deleting a file or a directory.
Mkdir: mkdir means make directory and is used to create a directory
Rmdir: rmdir means remove directory and is used to delete a particular directory.
echo: echo command is similar to cat command which is used to display contents and it can also display the value of a variable.
more :used to view the text files in the command prompt, displaying one screen at a time in case the file is large.

date: This command is used to display the system date and time. date command is also used to set date and time of the system. By default the date command displays the date in the time zone on which unix/linux operating system is configured

time: time command is used to execute a command and prints a summary of real-time user CPU time and system CPU time

Kill : shell built-in command used to terminate the process

history : history command is used to show all of the last used commands

chmod:chmod command utility is used to change permissions

chown: allows you to change the user and /or gropu ownership of a file or directory.

Finger:used to give details of all the users logged in.generally used by system adminstrators.

Pwd: pwd means present working directory and it prints the path if current working directory.

Cal: A command utility to display calendar in terminal.

Logout: Logout command allows you to programmatically logout from your session.

Shutdown: This command brings the system down in a secure way.All processes are first notified that the system is going to be shutdown.

SYTAX OF COMMAND:

man: $man [option]…[command name]…

who:$who [options] [filename]

cat:$cat filename [file2]

cd: $cd path_of_directory

cp: $cp source destination

$cp sorce1 source2…….sourcen destination

ps: $ps [option]

ls: $ls directoryname

mv: $mv source destination

rm: $rm [option]…file

mkdir:$mkdir [options]…[directories]

rmdir: rmdir [options]…[directoryname]

echo: echo [option] [string]

more: $more [options][num][pattern][filename]

date:$date

time: $time

kill : $kill -l

$kill pid

history: $history

chmod: $chmod [reference] [operator] [mode] file.

chown: $chown ownername filename

finger:$finger [-lmsp] [user1 user2 user3….]

pwd: $pwd

cal: $cal [year]

logout:$logout

shutdown: $shutdown [options][time][message]

SCREENSHOTS:

1.man cp

```
NAME
       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

       Mandatory  arguments  to  long options are mandatory for short
       too.

       -a, --archive
              same as -dR --preserve=all

       --attributes-only
              don't copy the file data, just the attributes

       --backup[=CONTROL]
              make a backup of each existing destination file

       -b     like --backup but does not accept an argument

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

2.who

3.cat

```
nag-1211@ubuntu-VirtualBox:~$ who
nag-1211 :0           2020-12-10 20:40 (:0)
nag-1211@ubuntu-VirtualBox:~$ cat file1
red blue green
yellow pink black violet
```

4.ls

5.cd

```
nag-1211@ubuntu-VirtualBox:~$ ls
Desktop    Downloads         file1  Pictures  Templates
Documents  examples.desktop  Music  Public    Videos
nag-1211@ubuntu-VirtualBox:~$ cd Desktop
nag-1211@ubuntu-VirtualBox:~/Desktop$ cd ..
```

6.cp

7.ps

```
nag-1211@ubuntu-VirtualBox:~$ cp  file1 copiedfile
nag-1211@ubuntu-VirtualBox:~$ ps
  PID TTY          TIME CMD
 1672 pts/0    00:00:00 bash
 1819 pts/0    00:00:00 ps
nag-1211@ubuntu-VirtualBox:~$
```

8.mv

9.rm

```
nag-1211@ubuntu-VirtualBox:~$ mv file1 file2
nag-1211@ubuntu-VirtualBox:~$ cat file1
cat: file1: No such file or directory
nag-1211@ubuntu-VirtualBox:~$ cat file2
red blue green
yellow pink black violet

nag-1211@ubuntu-VirtualBox:~$ rm file2
nag-1211@ubuntu-VirtualBox:~$ cat file2
cat: file2: No such file or directory
```

10.mkdir

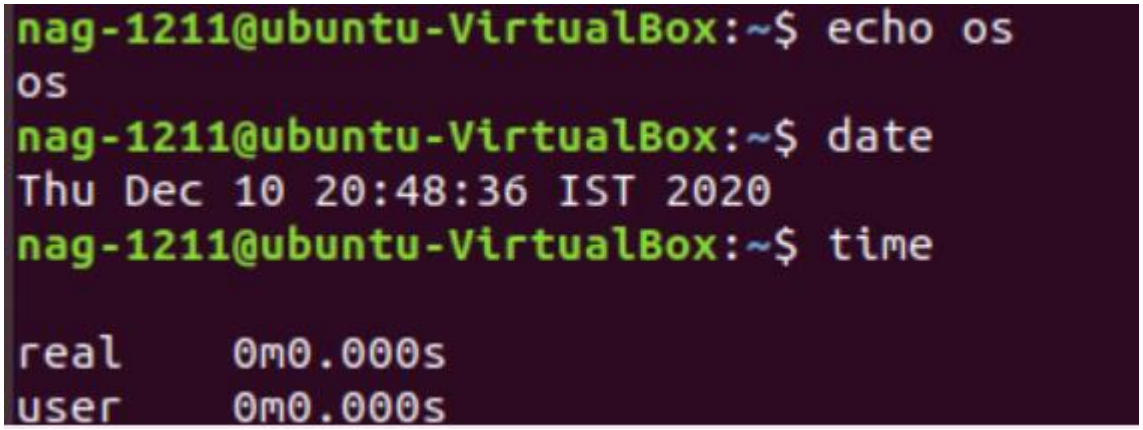
11.rmdir

```
nag-1211@ubuntu-VirtualBox:~$ mkdir newdir
nag-1211@ubuntu-VirtualBox:~$ ls
copiedfile  Documents  examples.desktop  newdir    Public     Videos
Desktop     Downloads  Music             Pictures  Templates
nag-1211@ubuntu-VirtualBox:~$ rmdir newdir
nag-1211@ubuntu-VirtualBox:~$ ls
copiedfile  Documents  examples.desktop  Pictures  Templates
Desktop     Downloads  Music             Public    Videos
```

12.echo

13.date

14.time



15.history

```
nag-1211@ubuntu-VirtualBox:~$ history
    1  man cp
    2  who
    3  clear
    4  ed file1
    5  clear
    6  ls
    7  cat > file1
    8  ls
    9  clear
   10  who
   11  cat file1
   12  ls
   13  cd Desktop
   14  cd ..
   15  cp  file1 copiedfile
   16  ps
   17  clear
   18  mv file1 file2
   19  cat file1
   20  cat file2
   21  rm file2
   22  cat file2
   23  mkdir newdir
   24  ls
   25  rmdir newdir
```

16.pwd

```
nag-1211@ubuntu-VirtualBox:~$ pwd
/home/nag-1211
nag-1211@ubuntu-VirtualBox:~$
```

17.kill

```
nag-1211@ubuntu-VirtualBox:~$ sleep 10 &
[1] 1871
nag-1211@ubuntu-VirtualBox:~$ ps
  PID TTY          TIME CMD
 1848 pts/1    00:00:00 bash
 1871 pts/1    00:00:00 sleep
 1872 pts/1    00:00:00 ps
nag-1211@ubuntu-VirtualBox:~$ kill 1871
[1]+  Terminated              sleep 10
nag-1211@ubuntu-VirtualBox:~$
```

17.chown

18.chmod

```
nag-1211@ubuntu-VirtualBox:~$ ls -l file1
-rw-r--r-- 1 nag-1211 nag-1211 41 Dec 10 20:57 file1
nag-1211@ubuntu-VirtualBox:~$ chmod 101 file1
nag-1211@ubuntu-VirtualBox:~$ ls -l file1
---x-----x 1 nag-1211 nag-1211 41 Dec 10 20:57 file1
nag-1211@ubuntu-VirtualBox:~$
```

19.logout

```
nag-1211@ubuntu-VirtualBox:~$ logout
bash: logout: not login shell: use `exit'
nag-1211@ubuntu-VirtualBox:~$
```

20.shutdown

```
nag-1211@ubuntu-VirtualBox:~$ shutdown
Shutdown scheduled for Thu 2020-12-10 21:03:04 IST, use 'shutdown -c' to cancel
.
nag-1211@ubuntu-VirtualBox:~$ shutdown -c
nag-1211@ubuntu-VirtualBox:~$
```

21 . more

22.cal

```
nag-1211@ubuntu-VirtualBox:~$ more file1
red blue green
yellow pink black violet

nag-1211@ubuntu-VirtualBox:~$ cal
   December 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

nag-1211@ubuntu-VirtualBox:~$
```

23.finger

```
nag-1211@hp:~$ finger nag-1211
Login: nag-1211                         Name: NagababuThota
Directory: /home/nag-1211               Shell: /bin/bash
On since Thu Dec 10 09:38 (IST) on :0 from :0 (messages off)
No mail.
No Plan.
nag-1211@hp:~$ finger user2
Login: user2                            Name: Nagbabu
Directory: /home/user2                  Shell: /bin/bash
Office: 2, 999                          Home Phone: x9999
Never logged in.
No mail.
No Plan.
```

1b.

AIM:

Study of vi editor.

DESCRIPTION:

vi is a text **editor**, not a "what you see is what you get" word processor. **vi** lets you add, change, and delete text, but does not provide such formatting capabilities as centering lines or indenting paragraphs. This help note explains the basics of **vi**: opening and closing a file.

While working with the vi editor, we usually come across the following two modes −

- **Command mode** − This mode enables you to perform administrative tasks such as saving the files, executing the commands, moving the cursor, cutting (yanking) and pasting the lines or words, as well as finding and replacing. In this mode, whatever you type is interpreted as a command.
- **Insert mode** − This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and placed in the file.

VI Editing commands

- i - **Insert** at cursor (goes into **insert mode**)
- a - Write after cursor (goes into **insert mode**)
- A - Write at the end of line (goes into **insert mode**)

- **ESC** - Terminate **insert mode**.
- u - **Undo** last change.
- U - **Undo** all changes to the entire line.
- o - Open a new line (goes into **insert mode**)
- dd - **Delete** line

Quit the vi editor without saving your changes

1. If you are currently in insert or append mode, press Esc .
2. Press : (colon). The cursor should reappear at the lower left corner of the screen beside a colon prompt.
3. Enter the following: q! This will quit the **editor**, and all changes you **have** made to the document will be lost.

To save changes:

:wq is used instead of :q

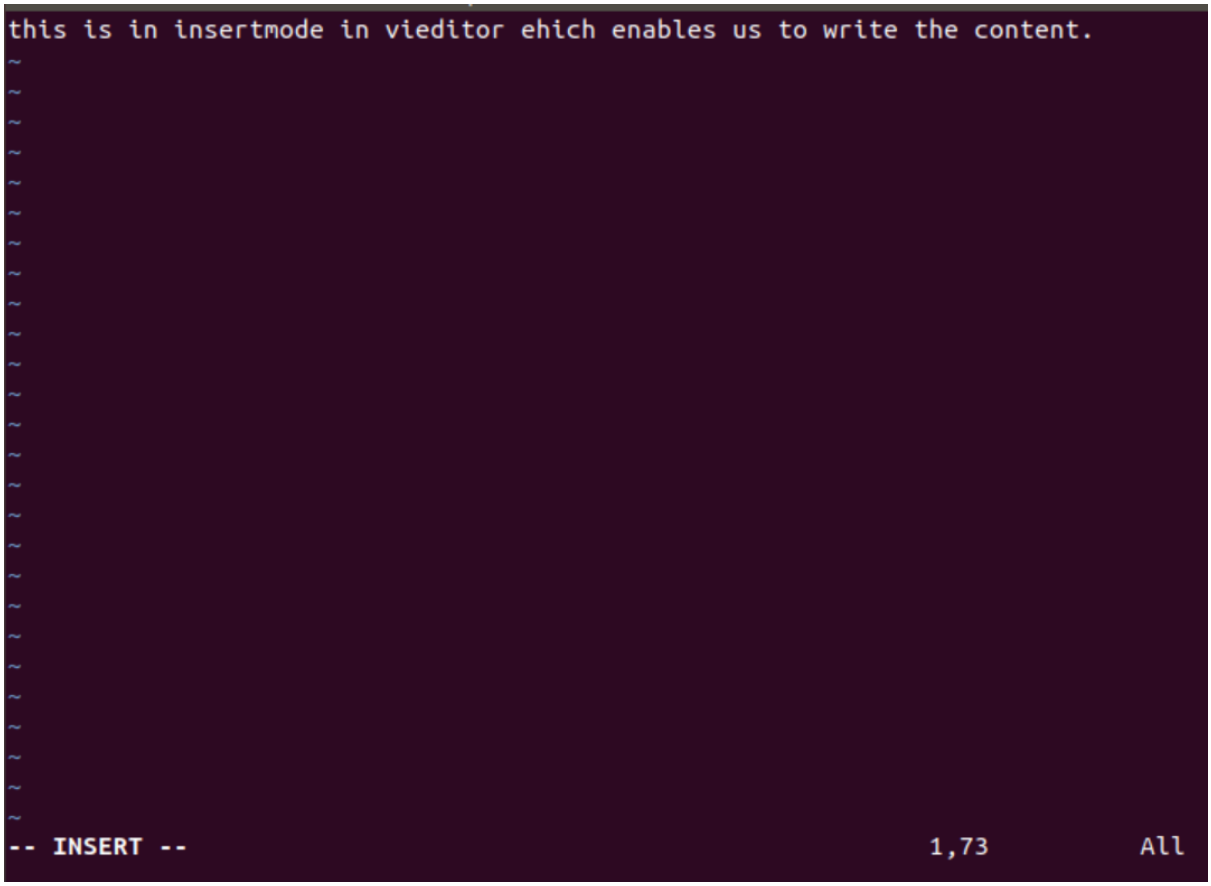
Moving within a File

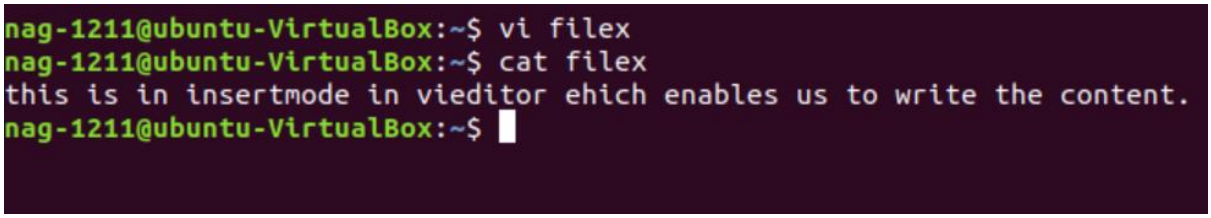
To move around within a file without affecting your text, you must be in the command mode (press Esc twice). The following table lists out a few commands you can use to move around one character at a time –

1	**k** Moves the cursor up one line
2	**j** Moves the cursor down one line
3	**h** Moves the cursor to the left one character position
4	**l** Moves the cursor to the right one character position

Syntax : Vi filename

Output:


this is in insertmode in vieditor ehich enables us to write the content.
-- INSERT --
1,73
All


nag-1211@ubuntu-VirtualBox:~$ vi filex
nag-1211@ubuntu-VirtualBox:~$ cat filex
this is in insertmode in vieditor ehich enables us to write the content.
nag-1211@ubuntu-VirtualBox:~$

1c.

Aim:

Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system.

Description:

Shell : SHELL is a program which provides the interface between the user and an operating system. When the user logs in OS starts a shell for user.

Types of Shells in Linux

- Bash shell
- Bourne shell
- C shell

Bash shell : The popularity of sh motivated programmers to develop a shell that was compatible with it, but with several enhancements. Linux systems still offer the sh shell, but "bash" -- the "Bourne-again Shell," based on sh -- has become the new default standard. One attractive feature of bash is its ability to run sh shell scripts unchanged. Shell scripts are complex sets of commands that automate

programming and maintenance chores; being able to reuse these scripts saves programmers time. Conveniences not present with the original Bourne shell include command completion and a command history.

Bourne shell : The Bourne shell, called "sh," is one of the original shells, developed for Unix computers by Stephen Bourne at AT&T's Bell Labs in 1977. Its long history of use means many software developers are familiar with it. It offers features such as input and output redirection, shell scripting with string and integer variables, and condition testing and looping. The Bourne shell is the Solaris OS default shell. It is the standard shell for Solaris system administration scripts. For the Bourne shell the:

- Command full-path name is **/bin/sh** and **/sbin/sh**.
- Non-root user default prompt is **$**.
- Root user default prompt is **#**.

C shell : Developers have written large parts of the Linux operating system in the C and C++ languages. Using C syntax as a model, Bill Joy at Berkeley University developed the "C-shell," csh, in 1978. Ken Greer, working at Carnegie-Mellon University, took csh concepts a step forward with a new shell, tcsh, which Linux systems now offer. Tcsh fixed problems in csh and added command completion, in which the shell

makes educated "guesses" as you type, based on your system's directory structure and files. Tcsh does not run bash scripts, as the two have substantial differences.

Is a UNIX enhancement written by **Bill Joy** at the University of California at Berkeley.

Incorporated features for interactive use, such as **aliases** and **command history**.

Includes convenient programming features, such as **built-in arithmetic** and a **C-like expression syntax**.

1d.

AIM:

Study of Unix/Linux file system (tree structure).

DESCRIPTION:

A file system is a logical collection of files on a partition or disk UNIX uses a hierarchical file system structure, much like an upside-down tree, with root (/) at the base of the file system and all other directories spreading from there.

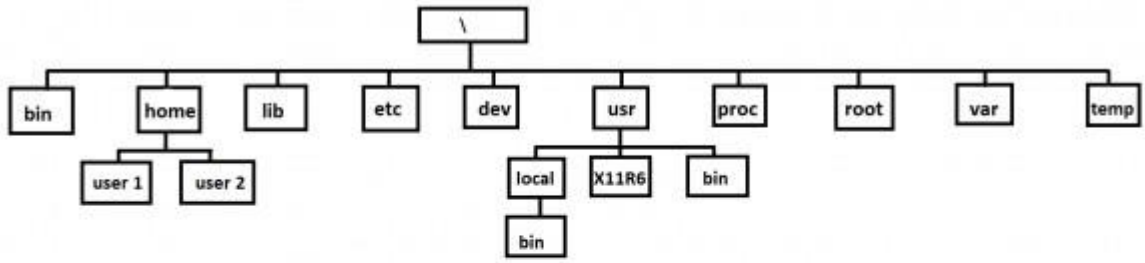


figure-1
to show linux is a hierarchical structure with upside down tree .

A UNIX filesystem is a collection of files and directories that has the following properties :

1. It has a root directory (/) that contains other files and directories.
2. Each file or directory is uniquely identified by its name, the directory in which it resides, and a unique identifier, typically called an inode.

- By convention, the root directory has an inode number of 2 and the lost+found directory has an inode number of 3. Inode numbers 0 and 1 are not used. File inode numbers can be seen by specifying the -i option to ls command.
- It is self contained. There are no dependencies between one filesystem and any other.

The directories have specific purposes and generally hold the same types of information for easily locating files. Following are the directories that exist on the major versions of Unix :

Directory	Description
/	This is the root directory which should contain only the directories needed at the top level of the file structure
/bin	This is where the executable files are located. They are available to all user.
/dev	These are device drivers.
/etc	Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages
/lib	Contains shared library files and sometimes other kernel-related files
/boot	Contains files for booting the system.
/home	Contains the home directory for users and other accounts
/mnt	Used to mount other temporary file systems, such as cdrom and floppy for the

Directory	Description
	CDROM drive and floppy diskette drive, respectively
/proc	Contains all processes marked as a file by process number or other information that is dynamic to the system
/tmp	Holds temporary files used between system boots
/user	Used for miscellaneous purposes, or can be used by many users. Includes administrative commands, shared files, library files, and others
/var	Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data
/sbin	Contains binary (executable) files, usually for system administration. For example fdisk and ifconfig utlities.

Directory	Description
/kernel	Contains kernel files

1e.

AIM:

Study of .bashrc, /etc/bashrc and Environment variables.

Description:

The /etc/bashrc is executed for both interactive and non-interactive
shells. /etc/bashrc or /etc/bash.bashrc is the systemwide bash per-interactive-shell startup file. Is is used system wide functions and aliases.
However, environment stuff goes
in /etc/profile file.the /etc/profile is executed only for interactive shells
.bashrc is a shell script that Bash runs whenever it is started interactively. It initializes an interactive shell session.
.bashrc runs on every interactive shell launch.
Following is the partial list of important environment variables :-

1. **DISPLAY :** Contains the identifier for the display that X11 programs should use by default.
2. **HOME :** Indicates the home directory of the current user: the default argument for the cd built-in command.
3. **IFS :** Indicates the Internal Field Separator that is used by the parser for word splitting after expansion.
4. **LANG :** LANG expands to the default system locale; LC_ALL can be used to override this. For example, if its value is pt_BR, then the language is

set to (Brazilian) Portuguese and the locale to Brazil.

5. **LD_LIBRARY_PATH :** On many Unix systems with a dynamic linker, contains a colonseparated list of directories that the dynamic linker should search for shared objects when building a process image after exec, before searching in any other directories.
6. **PATH :** Indicates search path for commands. It is a colon-separated list of directories in which the shell looks for commands.
7. **PWD :** Indicates the current working directory as set by the cd command.
8. **RANDOM :** Generates a random integer between 0 and 32,767 each time it is referenced.
9. **SHLVL :** Increments by one each time an instance of bash is started. This variable is useful for determining whether the built-in exit command ends the current session.
10. **TERM :** Refers to the display type
11. **VZ :** Refers to Time zone. It can take values like GMT, AST, etc.
12. **UID :** Expands to the numeric user ID of the current user, initialized at shell startup.

DIFFERENCE:

One difference is that /etc/environment contains only variable definitions and doesn't appear to go through any sort of variable

expansion/interpolation. Thus, you can't reference variables in definitions.