

OPERATING SYSTEM LAB MANUAL

Ex.No:1.a	BASICS OF UNIX COMMANDS
	INTRODUCTION TO UNIX

AIM:

To study about the basics of UNIX

UNIX:

It is a multi-user operating system. Developed at AT & T Bell Industries, USA in 1969.

Ken Thomson along with Dennis Ritchie developed it from MULTICS (Multiplexed Information and Computing Service) OS.

By 1980, UNIX had been completely rewritten using C language.

LINUX:

It is similar to UNIX, which is created by Linus Torvalds. All UNIX commands work in Linux. Linux is an open source software. The main feature of Linux is coexisting with other OS such as Windows and UNIX.

STRUCTURE OF A LINUX SYSTEM:

It consists of three parts.

- a) UNIX kernel
- b) Shells
- c) Tools and Applications

UNIX KERNEL:

Kernel is the core of the UNIX OS. It controls all tasks, schedules all processes and carries out all the functions of OS.

Decides when one program tops and another starts.

SHELL:

Shell is the command interpreter in the UNIX OS. It accepts commands from the user and analyses and interprets them.

Ex.No:1.b	BASICS OF UNIX COMMANDS
	BASIC UNIX COMMANDS

AIM:

To study of Basic UNIX Commands and various UNIX editors such as vi, ed, ex and EMACS.

CONTENT:

Note: Syn->Syntax

a) date

–used to check the date and time

Syn:\$date

Format	Purpose	Example	Result
+%m	To display only month	\$date+%m	06
+%h	To display month name	\$date+%h	June
+%d	To display day of month	\$date+%d	01
+%y	To display last two digits of years	\$date+%y	09
+%H	To display hours	\$date+%H	10
+%M	To display minutes	\$date+%M	45
+%S	To display seconds	\$date+%S	55

b) cal

–used to display the calendar

Syn:\$cal 2 2009

c) echo

–used to print the message on the screen.

Syn:\$echo "text"

d) ls

–used to list the files. Your files are kept in a directory.

Syn:\$ls-ls

All files (include files with prefix)

ls-l Ldetail (provide file statistics)

ls-t Order by creation time

ls-u Sort by access time (or show when last accessed together with -l)

ls-s Order by size

ls-r Reverse order

ls-f Mark directories with /, executable with * , symbolic links with @, local sockets with =, named pipes(FIFOs)with

ls-s Show file size

ls-h "Human Readable", show file size in Kilo Bytes & Mega Bytes (h can be used together with -l or)

ls[a-m]*List all the files whose name begin with alphabets From „a“ to „m“

ls[a]*List all the files whose name begins with „a“ or „A“

Eg:\$ls>my list Output of „ls“ command is stored to disk file named „my list“

e)lp

–used to take printouts

Syn:\$lp filename

f)man

–used to provide manual help on every UNIX commands.

Syn:\$man unix command

\$man cat

g)who & whoami

–it displays data about all users who have logged into the system currently. The next command displays about current user only.

Syn:\$who\$whoami

h) uptime

–tells you how long the computer has been running since its last reboot or power-off.

Syn:\$uptime

i)uname

–it displays the system information such as hardware platform, system name and processor, OS type.

Syn:\$uname–a

j) hostname

–displays and set system host name

Syn:\$ hostname

k) bc

–stands for „best calculator“

\$bc	\$ bc	\$ bc	\$ bc
10/2*3	scale =1	ibase=2	sqrt(196)
15	2.25+1	obase=16	14 quit
	3.35	11010011	
	quit	89275	
		1010	
		Ā	
		Quit	
\$bc	\$ bc-l		
for(i=1;i<3;i=i+1)I	scale=2		
1	s(3.14)		
2	0		
3 quit			

FILE MANIPULATION COMMANDS

a) **cat**—this create, view and concatenate files.

Creation:

Syn:\$cat>filename

Viewing:

Syn:\$cat filename

Add text to an existing file:

Syn:\$cat>>filename

Concatenate:

Syn:\$catfile1file2>file3

\$catfile1file2>>file3 (no over writing of file3)

b) **grep**—used to search a particular word or pattern related to that word from the file.

Syn:\$grep search word filename

Eg:\$grep anu student

c) **rm**—deletes a file from the file system

Syn:\$rm filename

d) **touch**—used to create a blank file.

Syn:\$touch file names

e) **cp**—copies the files or directories
Syn:\$cpsource file destination file
Eg:\$cp student stud

f) **mv**—to rename the file or directory

syn:\$mv old file new file

Eg:\$mv-i student student list(-i prompt when overwrite)

g) **cut**—it cuts or pickup a given number of character or fields of the file.

Syn:\$cut<option><filename>

Eg: \$cut -c filename

\$cut-c1-10emp

\$cut-f 3,6emp

\$ cut -f 3-6 emp

-c cutting columns

-f cutting fields

h) **head**—displays10 lines from the head(top)of a given file

Syn:\$head filename

Eg:\$head student

To display the top two lines:

CDLSIET PANNIWALA MOTA SIRSA(HRY)

Syn:\$head-2student

i) **tail**–displays last 10 lines of the file

Syn:\$tail filename

Eg:\$tail student

To display the bottom two lines;

Syn:\$ tail -2 student

j) **chmod**–used to change the permissions of a file or directory.

Syn:\$ch mod category operation permission file

Where, Category–is the user type

Operation–is used to assign or remove permission

Permission–is the type of permission

File–are used to assign or remove permission all

Examples:

\$chmodu-wx student

Removes write and execute permission for users

\$ch modu+rw,g+rwestudent

Assigns read and write permission for users and groups

\$chmodg=rwx student

Assigns absolute permission for groups of all read, write and execute permissions

k) **wc**–it counts the number of lines, words, character in a specified file(s)

with the options as -l,-w,-c

Category	Operation	Permission
u– users	+assign	r– read
g–group	-remove	w– write
o– others	=assign absolutely	x–execute

Syn: \$wc -l filename

\$wc -w filename

\$wc -c filename

Ex.No:1.c	BASICS OF UNIX COMMANDS
	UNIX EDITORS

AIM:

To study of various UNIX editors such as vi, ed, ex and EMACS.

CONCEPT:

Editor is a program that allows user to see a portions a file on the screen and modify characters and lines by simply typing at the current position. UNIX supports variety of Editors. They are:

ed ex vi
EMACS

Vi- vi is stands for “visual”.vi is the most important and powerful editor.vi is a full screen editor that allows user to view and edit entire document at the same time.vi editor was written in the University of California, at Berkley by Bill Joy, who is one of the co-founder of Sun Microsystems.

Features of vi:

It is easy to learn and has more powerful features.

It works great speed and is case sensitive.vi has powerful undo functions and has 3 modes:

1. Command mode
2. Insert mode
3. Escape or ex mode

In command mode, no text is displayed on the screen.

In Insert mode, it permits user to edit insert or replace text.

In escape mode, it displays commands at command line.

Moving the cursor with the help of h, l, k, j, I, etc

EMACS EditorMotion Commands:

M-> Move to end of file

M-< Move to beginning of file

C-v Move forward a screen M -v Move backward a screen C -n Move to next line

C-p Move to previous line

C-a Move to the beginning of the line

C-e Move to the end of the line

C-f Move forward a character

C-b Move backward a character

M-f Move forward a word

M-b Move backward a word

Deletion Commands:

DEL delete the previous character C -d
delete the current character M -DEL
delete the previous word
M-d delete the next word
C-x DEL deletes the previous sentence
M-k delete the rest of the current sentence
C-k deletes the rest of the current line
C-xu undo the lasted it change

Search and Replace in EMACS:

y Change the occurrence of the pattern
n Don't change the occurrence, but look for the other q Don't change. Leave query
replace completely
! Change this occurrence and all others in the file

RESULT:

Ex.No:2

Programs using the following system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir

AIM:

To write C Programs using the following system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir.

1. PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEMS (OPENDIR, READDIR, CLOSEDIR)

ALGORITHM:

- STEP 1: Start the program.
- STEP 2: Create struct dirent.
- STEP 3: declare the variable buff and pointer dptr.
- STEP 4: Get the directory name.
- STEP 5: Open the directory.
- STEP 6: Read the contents in directory and print it.
- STEP 7: Close the directory.

PROGRAM:

```
#include<stdio.h>
#include<dirent.h>
struct dirent *dptr;
int main(int argc, char *argv[])
{
char buff[100];
DIR *dirp;
printf("\n\n ENTER DIRECTORY NAME");
scanf("%s", buff);
if((dirp=opendir(buff))==NULL)
{
printf("The given directory does not exist");
exit(1);
}
while(dptr=readdir(dirp))
{
printf("%s\n",dptr->d_name);
}
closedir(dirp);
}
```

OUTPUT:

2. PROGRAM FOR SYSTEM CALLS OF UNIX OPERATING SYSTEM (fork, getpid, exit)

ALGORITHM:

STEP 1: Start the program.

STEP 2: Declare the variables pid,pid1,pid2.

STEP 3: Call fork() system call to create process.

STEP 4: If pid==-1, exit.

STEP 5: If pid!=-1, get the process id using getpid().

STEP 6: Print the process id.

STEP 7: Stop the program

PROGRAM:

```
#include<stdio.h>
#include<unistd.h>
main()
{
int pid,pid1,pid2;
pid=fork();
if(pid==-1)
{
printf("ERROR IN PROCESS CREATION \n");
exit(1);
}
if(pid!=0)
{
pid1=getpid();
printf("\n the parent process ID is %d\n", pid1);
}
else
{
pid2=getpid();
printf("\n the child process ID is %d\n", pid2);
```

```
}  
}
```

OUTPUT:

RESULT:

.

CDLSIET PANNIWALA MOTA SIRSA(HRY)