INDEX

Sl. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission
1.	Basic commands of unix & file commands	1-3	20/11/25	
2.	Listing files and directories	4-8	27/1/25	
3.	Disk related commands	9-10	3/2/25	
4.	Processes in Unix	11-12	10/2/25	
5.	Communication in Unix	13-17	17/2/25	
6.	Mails commands	18-21	24/2/25	
7.	Shell Programming	22-23	10/3/25	
8.	Variables in scripting	24-25	17/3/25	
9.	Taking Decisions	26-28	24/3/25	
10.	While, Until & For Loop	29-31	7/4/25	

Date

Expt. No. 1 Page No. 1

Basic Unix Commands

$who am i
In this command, the Kernel returns info about you and display it on the screen. The info consist of login name, terminal no. & date, time at which you logged in.
ubuntu : type 2025-02-06 8:50

$who
This command is more powerful and displays data about all the users who have logged into system currently.
aa1 tty3a 2025-02-03 15:05
aa2 tty3c 2025-01-26 12:15

Exit/ ctrl+d
The end of a unix session is marked by a logout.

$touch sample.txt
This creates a file called 'Sample'. However, the size of the file would be zero byte since touch is used to create several Empty files.

$ cat > Test.txt
Hello
This cat creates a file in which we can store the content accordingly. After writing the command, the cursor will be positioned in next line for writing. To come End of file press key ctrl+D.

$ cat Test.txt
This command is used for traversing the file.

$ cat sample1.txt sample2.txt >> newsample.txt
This would create newsample which contains contents of sample1 followed by that of sample2. If newsample already contains something it would be overwritten. If you want that it should remain intact and contents of sample1 and sample2 should get appended to it then you should use the append output redirection operator, >>

$ cp letter.a letter.b
This will copy contents of letter.a into letter.b. If letter.b does not Exist, it will be created.

$ cp letter.a letter.b letters
$ cp /c:\x22\a3c\letter.a D:\\wc\letter.b

$ mv test.txt sample.txt
Renaming of files in DOS is interpreted in unix as moving them. Moving is different then copying in that the source file is Erased from its original location & copied at specified destination so test file will be deleted & content will be shown in sample file as new.

Date

Expt. No. 2 Page No. 4

Listing files and Directories

$umask
0002
stands for user file creation mask, implying which permissions to mask or hide. The umask value tells unix which of three permissions are to be denied rather than granted. The 3 digits refers to the permission to be denied to the Owner, group and others. whenever a file is created unix assumes that the permissions for the file should be 666. but umask is 002, so it will be 664 means rw-rw-r--

$umask 242
To change the umask value of a file or directory.

$rm -i file
rm command in Dos is counterpart in unix. rm removes the file. -i is a switch, removes file interactively; i.e. you are asked for confirmation for deleting.

Teacher's Signature :

Date

Expt. No. Page No. 5

$rm -r dir1
Removes all the contents of ~~file~~ dir1 & also dir1 itself.

$ ls
This command is used to give directory listing.
carribeans
Kiwis
zulus

$ cat >.cricket
Hello Hi Cricket
ctrl d
Well, filename can begin with a dot (.) file with the prefix dot is treated as a hidden file. And if we want to list even the hidden files we need to use the -a option of ls.

$ls -a
.cricket
Carribeans
Zulus

$ls p*
poem
pommies

$ls ?ain
gain
rain

Teacher's Signature :

Date

Expt. No. Page No. 6

$ ls -ld mydirectory
drwxr-xr-x
This ld gives about permission information of directory files.

$ ln poem mypoem
This Establishes one more link for the file poem in form of name newpoem. The concept of having several links to a file offers advantages. If one file is to be shared between several users, instead of giving Each user a separate copy of same file we can create links of this file in Each user's directory. Avoids unnecessary duplication of file or directories.

$ chmod 700 myfile
The way to change file permissions is by using chmod. It changed the mode of file.
rwxrwxrwx to rwx------.

Teacher's Signature :

Date

Expt. No. Page No. 7

$ ls -ld mydir
drwxrwxrwx 2 user1 group 50 feb13 10:50 mydir

$ rm -f letter
Remove a file forcibly which do not have a write permission, if we have permission r-- we can delete

$ lc
Displays the files in columnar fashion.

$ lf
Puts a * after all Executable files and a / after all sub-directories present in current directories.
Carribeans*
Kiwis
Zulus/

$ pwd
/home/ubuntu
Tells current working directory

$ mkdir book
Creates a new directory named book.

Teacher's Signature :

$ rmdir -p works/bpb/unix/book
In order to remove the parent directories of book.

# $ mKdir newdir
$ cd newdir
In unix cd is for changing over to a new directory.
Change over to current user's home directory

Mathematical Commands

$bc
10/2*2
10

$factor
15
3 5

$ expr 3 /* 2
6 (result of Expression)

Date

Expt. No. 3 Page No. 9

Disk Related Commands

$cmchK
BSIZE=1024
This command gives size of block.

$ls -i reports
reports 12324
To find inode number associated with file in unix.

$df -ivt
df is disk free. This command reports the free as well as the used disk space for all file system.
/dev/root 282098 blocks 269146 used 12952 free 27857 i-nod

$./etc/dfspace
dfspace does all the mathematics internally and reports free disk space for root file system.
It is present in /etc directory.

$du/dev
du is reporting number of blocks used by directory.

$ulimit
signifies the largest file that can be created by user.

Teacher's Signature :

$banner YMCA
This command prints a message in large which looks as banner.

$cal 02 feb.
Calender

$sort myfile
Sorting

Date

Expt. No. 4 Page No. 11

Processes in Unix

$ps

PID TTY Time Command
2269 3a 0:05 sh

ps shows the PID for 2 processes being run by us when Executed.

$ps -a

standing for processes of all the users. It gives list.

$ps -t tty3d

It lets you find out the process that have been launched from a particular terminal.

$ps -f

UID PID PPID C STime TTY Time Command
icit 288 1 1 09:32:25 02 0:01 -sh

-f standing for full listing

$ps -e

scheduler process running at all times in memory.

$:sort employee.dat > emp.out &

To run a processes in background, unix provides the ampersand (&) symbol.

12956

Teacher's Signature :

$ kill 6173
6173 Terminated
Kill command is to terminate process.

$ nice cat employee.dat
The priority of a process is decided by a number associated. Higher the nice value of a process lower is its priority.

$ nice --10 sort employee.dat > output.dat
Its only the superuser who can put a process on a higher priority by reducing its nice value.

$ ps -l
gives priority to all

$ at 17:00
echo "Its 5pm! Backup your files and layout"
ctrl d
Job 853158864.a at Wed Jun 14 17:00:00 IST 1996.
At 5PM in evening you would see a message on your screen saying 'you have mail'.
To examine what you have received in mail just type. $ mail

Date

Expt. No. 5 Page No. 13

Communication in Unix Style

$ batch
sort employee.dat | grep Nagpur > address.out
ctrl d
job 692322435.b at Fri Jun 14 17:00:00 IST 1996.

Instead of we specifying that our commands be executed at a precise moment in time sometimes we may let the system decide the best time for executing our commands. The way to achieve this is through a command called batch. When we submit our jobs using this command, Unix executes our job when it is relatively free and system load is light. Once again note that, 'b' extension given to our job id signifies that it has been submitted using batch command.

$ at 5PM
cp *.c ./cbackup

at and batch are powerful tools for scheduling processes, both suffer from an obvious limitation.

$ crontab cmdfile

crontab command excels overat. It can carry out a submitted job every day for years together, without needing any prompting from us.

Teacher's Signature :

Minute Hour DayofMonth Monthofyear Dayofweek command

$cat > cmdfile
30 10 1 * * echo "Word hard on first day of month"
0 0 17 11 * mail aa ? < confi.letter
ctrl d
$

$crontab -l
as with the at command we can view the commands that we have submitted by using the -l option with crontab command.

$crontab -r
To remove the submitted job.

$crontab cmdfile
We are not required to specify job-id since using crontab. If we want to schedule a few more jobs we need to edit cmdfile in our home directory and then resubmit.

$write user 2
Hey there! I am back from Paris
ctrl d

Write command can be used by any user to write something on someone's else's terminal. On Executing this command the message would be relayed to user whose login name is user2. He would hear the beap.

There are Two main prequisites for a smooth write operation:

a) The recipient must be logged in
b) The recipient must have given permission for messages to reach his terminal.

$ mesg -y

Deny → $ mesg -n

$ finger -i

Login	TTY	When	Idle
Veena	*tty01	fri Oct 13 17:25	8 minutes 12 sec
prafull	tty3f	fri oct 13 17:21	49 seconds

Finger is command tells you which users are connected and displays a list of all those who have logged in and places * where mesg is set to -n.

Teacher's Signature :

$who -T
Veena -tty01 Oct 13 17:25
prafull -tty3f Oct 13 17:21

it places a '+' next to users who have allowed messages and a '-' sign beside other.
who list the currently logged in.

$write user3
Permission denied

#/etc/wall
wall is a command to send messages to all logged in users. It is not a standard file but might be used in specific setups for broadcasting system messages.
e.g. # well 'system maintenance'.

$news
Displays system news message (if configure) a shows latest system updates/news.

$user/news
A directory where system news messages may be stored.
e.g. $cat /user/news/*

$news -n
shows only unread news messages.

$news -s
shows a short summary of news messages.

$cat /etc/motd
Welcome
contains a msg displayed when a user login.

$mail -f
Open the user's mailbox to read messages
record stored emails.

Date

Expt. No. 6 Page No. 18

Mail Commands

$ cut the 2nd column using space as:
$ cut -d " " -f2 filename.txt
Extracts specific columns or characters from a file.

$ cut -c1-5 filename.txt
Extract first 5 characters from each line.

$ grep
Searches for a pattern in a file or input
e.g:- $ grep "hello" file.txt
finds all lines containing "hello" in file.txt

$ grep "[Rr]oot" /etc/passwd.
Searches for both uppercase and lowercase versions.
matches "Root", 'root'.

$ grep b??k
users will cards '?' matches a single char
e.g.- $ grep - "b??k" words.txt
Matches words like 'book', 'back', 'bank'.

$ grep -v "error" logfile.txt
Inverts match (shows lines that do not contain pattern).
Shows line without the word "error".

Teacher's Signature :

$ head file.txt
Display first 10-20 lines of files.

$ pg filename.txt
Displays a file one page at a time.

$ more filename.txt
Similar to 'pg' allows viewing a file page by page.

$ lp filename.txt
sends file to the printer.

$ cancel job ID.
cancels a print job.

$ lpstat -t
shows status of printer system

$ lp -w filename.txt
waits for print job to finish before returning to command prompt.

$ compress filename.txt
Compress a file size.

$uncompress filename.txt.z
Decompresses z file
Restores 'filename.txt'

$ Pack filename.txt
Compress a file, similar to compress but older.

$unpack filename.txt.z
Decompress a file compressed with pack

$ man ls
Displays the manual pages for unix commands.
shows documentation for 'ls'.

$ man -K network
searches for commands related to network

Piping ('|')
sends output of one command as input to another.
e.g:- count files in directory:-
$ ls l we -l
- find a specific word in a file and count occurrences.
$ grep "errors" logfile.txt we-l
- show active processes and filters for a specific one.
$ ps au xl grep firefox.

Teacher's Signature :

-Sort a list and remove duplicates.
$ cat names.txt | sort | unique.

$ ls -l | tee file list.txt
sends output to file AND the terminal. saves output to 'file list.txt' and displays it.

$ echo 'New Entry' | tee -a log.txt
Adds 'new entry' to 'log.txt' without overwriting.

Date

Expt. No. 7 Page No. 22

Shell Script / Shell Programming

^ - Control
M - Alt
$ nano file 1
```
echo "What is your name\?"
read name
echo Hello $name
```
This command is for printing the string entered.

Unix is case sensitive language.
PS1 = $
$PS1 = "What next" → to change value of PS1
Pwd → home / Present working directory
There are 27 keywords.

$a = 20
$ read only a
to make value of c constant.

$ nano -- version
tells version of file

$ nano mayankK.sh
for writing code in vi editor use nano filename and with extension .sh

Teacher's Signature :

enter first num
5
enter Second num
2
Sum is = 7

Date

Expt. No. Page No. 23

nano --version
chmod 777 abc.sh
file Permission must be drwXrwXrwX-
nano abc.sh
ctrl+O for save
ctrl+X for exit
./abc.sh
for accessing the output.

```
echo 'calculator'
echo 'enter first num'
read n1
echo 'enter second number'
read n2
sum=$((n1+n2))
echo 'sum is = $sum'
```

Date

Expt. No. 8

Page No. 24

Variables in Scripting

Wiping Out Variable

$unset variable Name
$ unset f
system variable command

$1 --- $9 Positional Parameter

$ Set friend come and go, but enemies assemble.
$ Set You have the capability to long learn from mistakes.

$ echo $1 $2 $3 $4 $5 $6 $7 $8 $9 ~~$10~~
You have the capacity to long learn from mistakes

a=20, b=10
echo 'sum: '$(expr $a+$b)
echo 'Difference:'$(expr $a+$b)
echo 'Product:'$(expr $a* $b)
echo 'Quotient:'$(expr $a/$b)
echo 'Remainder:'$(expr $a%.$b)

Priority range of operators :- /* %. + -

Teacher's Signature :

Date

Expt. No. Page No. 25

$ shift 7

it will print all the words after $7

$1 = $A

⋮ ⋮

$6 = $F

Shift 7

} To print whole $ set sentence---

Date

Expt. No. 9 Page No. 26

Taking Decisions

The if-then-fi statement

```
if control command
    command 1
fi
```

```
Enter source & target filenames
read source target
if cp $ source $ target
then
    echo file copied successfully
fi
```

The if-then-else-fi statement

```
echo Enter a number from 1 to 10:'
read num
    if test $ num -lt 6
    then
        echo 'Hello YMCA'
    else
        echo 'Good Morning'
    fi
```

Test Command

This command investigates the sort of tests that we raised above and then it translates result into the language of success or failure.

Teacher's Signature :

This helps shell in deciding whether to execute commands in if block or commands in Else block.

```
echo "enter basic salary:"
read bs
echo "enter house rent allowance percentage:"
read hra
echo "enter other allowances:"
read oa
hra=$(echo $bs+$hra/100|bc)
total-sal=$(echo $bs+$hra+$oa|bc)
echo "basic salary:$bs"
echo "hra:$hra"
echo "other allowances:$oa"
echo "total salary:$total-sal"
```

Prime number

```
is-prime(){
local num=$1
if((num<=1)); then
echo "$num is not prime."
return
fi
```

Date

Expt. No. Page No. 28

```
for((i=2; i*i<=+num; i++)); do
if((num%i==0)); then
echo "$num is not a prime"
return fi done
echo "$num is a prime"}
echo "enter a number:"
read number
Ps_prime $ number.
```

Expt. No. 10 Date

Page No. 29

While, Until & For Loop

A Loop involves repeating some portion of program either a specific number of times or until a particular condition is being satisfied.

Calculate simple interest for 3 sets of principal

```
count = 1
while ["$count" -le 2]
do
echo -e "\n Enter values of p.n and r \c"
read p n r
si = $ (echo "scale=2; $p*$n*$r/100*|bc)
echo "Simple Interest = Rs $si"
count = $ ((count+1))
done
```

It is often case in programming that you want to do something a fixed number of times.

WAP for printing 1-10 using until loop.

```
i=1
until ["$i" -gt 10]
```

Teacher's Signature :

```
do
echo "$i"
i=$((i+1))
done
```

The statements within the until loop keep on getting Executed till Exit status of control command remain false.
When the exit status become true(0), the control pastes to first command that follows body of until loop.

★ The while loop Executes till exit status of control command is true and terminates when exit status become false. Unlike this until loop executes till exit status of control command is false and terminates when this status becomes true.

```
i=1
while [$i-le 10]
do
echo $i
i='expr $i+1'
done
```

```
i=1
until [$i-gt 10]
do
  echo $i
  i='expr $i+1'
done
```

Teacher's Signature : _____

Expt. No. _____ Date _____

Page No. 31

Write a for loop code.

```
for word in High on a hill was a lovely mountain.
do
   echo $ word
done


for word in $*
do
   echo $ word
done
```

Teacher's Signature : _____