**DAY -3:**

**SHELL VARIABLE:**

echo - is used to print

echo A = A

echo "hello world" = hello world

if we want stored values in variable we have to use '$'

eg: A=10

echo $A = 10

echo PATH - to know the path

echo SHELL - to know which shell we r using

**\*\*script executed automatically:**

history - gives all the commands we r used before

to create shellscript - vi .filename.sh

to execute the shellscript - sh filename.sh

**change permissions:**

**using** **numbers:**

1-execute

2- write

4- read

eg:

chmod 0444 filename - gives read permission for all

chmod 0666 filename - gives read &write (2+4)

chmod 0777 filename - gives read,write,execute (1+2+4)

chmod 0467 filename - give user read , group -r,w and others - r,w,x

**by using characters:**

eg: chmod o-r filename - it remove for only others

o- others r- read

g-group w- write

a- all x- execute

u- user

"-" - take away permission

"+"- give permission

chmod o+r filename - it give read permission to others

chown:

change the owner of the file

$ chown<owner name> <filename>

eg: $ chown priya test.txt - it change to priya

\* change group name

$ chgrp<groupname> <filename>

**program:**

echo "Enter your name"

read name

echo "your name is cp$name"

2. A=10

B=20

sum = `expr $A+$B`

echo "$A + $B = $sum"

3. var = `pwd`

echo $var

4. var=`date`

echo $var

special variable:

1. $0 = gives name of the file
2. $\* = gives complete set of positional parameters
3. $# = count of the arguments
4. $1 = first argument
5. $$ = PID of current shell
6. $? = gives the exit status of the last command
7. $! Gives PID of last background job
8. $@ = similar to $\* but it usually uses for string in looping construct

**find:**

used to find a file in directory

find command recursively decends

find ./ -name filename

find ./ -name “\*end char.extension”

eg: find ./ -name “\*t.sh”

**Filter:**

Filter is a command that takes its i/p from standard i/p process it and sends its o/p to the standard o/p.

Commands such as ls,date,pwd, etc. can not be used as filter as do not require any i/p

Some used filters are

Eg: grep, sort, cut, paste, head, tail, wc, pg, more, tee, tr.

Grep:

Global search for regular expression and print utility.

$ grep[option] <pattern><filename>

**Options:**

* -n : prints line num
* -v : the reverse search criterion
* -c: display only a count of matching patterns

**Regular Expressions:**

* “^” : beginning of line
* “$”: end of line
* “.”: anyb single char
* […]: any one char from list
* [^…]: string beginning from any char after ^ will be excluded
* \: used in conjunction with above

**Library creation:**

1. Static
2. Shared
3. Dynamic

Create the static library:

ar rcs librarylib.a file1.o file2.o

- shared = for shared library = .so

-PIC = for static library

**logical operators:**

* && operator delimits two commands. 2nd command is executed only if the 1st command succeeds.
* || operator delimits 2 commands. 2nd command is executed only if the 1st command fails