#UNIX:- UNIX was originally developed at Bell Laboratories as a private research project with a number of different operating system research effort in the 1970's. The goal of the group were to design an operating system to satisfy the following objectives

★ Simple and elegant
★ Written in high level language rather than assembly language
★ Allow re-use of code

UNIX had a relatively small amount of code written in assembly language (this is called the kernel) and the remaining code for the OS was written HLL called C.

The Kernel:- As its name implies, the kernel is at the core of each UNIX system and is loaded in whenever the system is started up reffered to as a boot of the system

It manages the entire resources of the system, presenting them to you and every other user as a coherent system. You do not need to know anything about the kernel in order to use UNIX system.

⇒ function performed by the kernal

- Managing the machine's memory and allocating it to each process
- Scheduling the work done by the CPU so that the work of each user is carried out as efficiently as is possible
- Organising the transfer of data from one part of the machine to another
- accepting instruction from the shell and carrying them out

enforcing the access permessions that are in force on the file system

Shell:- The shell is the UNIX/UNIX like/ Linux/XENIX system's command interpreter It is program that reads the lines you type in the terminal and

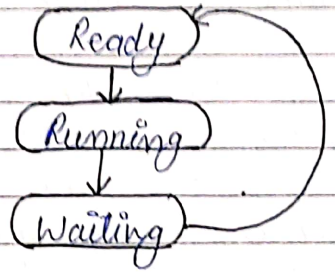
performs various operation depending on what you type in. When user log into UNIX/UNIXWare/Linux/XENIX system, they communicate with one of several interpreters. Each invocation of this interpreter is called a shell, and each shell has one function, to read and execute commands from its standard input.

⇒ Functions performed by the shell

- High level commands —converts→ Machine level language
- Process Management
- Process Scheduling
- Process priority

there are different levels of or states of process :-

- When process goes from hard disk to RAM then it is a 'Ready state'
- When the process that is in ready state will get the processor then it'll be in 'running state'
- When the process is waiting for any I/O device then it is a 'waiting state'



this is process management & scheduling earlier we have 5 stages

New → Ready → security → Waiting → terminated
Now the New state is removed.

#Batch OS :- It permits only one process at one time in RAM

There are different scheduling algorithms
- first come first service
- Round Robin
- Priority Basis
- shortest Job first

Command :-

→ mKdir :- It is command to make directory (or folder)
ex :- Suppose we need to make 50 folders then we will write a command like

mkdir name-of-directory {1, 2, ... 50}

to make 50 directories

⇒ Command prompt in UNIX is known as terminal. Now the terminal has
- CLI :- Command line interface
- GUI :- Graphical user interface

→ touch :- It is a command for making a file

ex :- Syntax :- touch filename.extension of that file

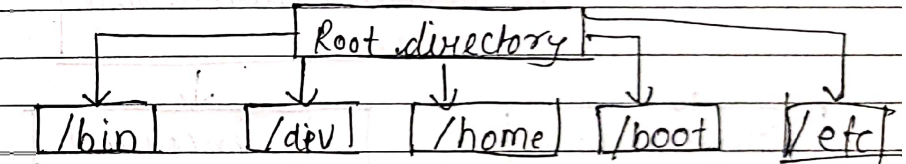
⇒ touch Hello.txt

The Unix OS is divided into two part
1. Shell
2. Kernal

⇒ Shell :- 1. It is the outer layer of UnixOS
2. Shell reads command provided by the user
3. It'll check whether the command is valid or not. If everything is proper the shell converts that command into Kernal understandable form and hand over it to the Kernal
4. Shell acts as an interface b/w the user and the kernel

⇒ Kernal :-
1. It is the core component of Unix OS
2. It is responsible to execute our commands with the help of hardware components
3. Memory allocation and processor allocation will be taken care by the Kernal
4. It acts as an interface b/w Hardware resources & shell

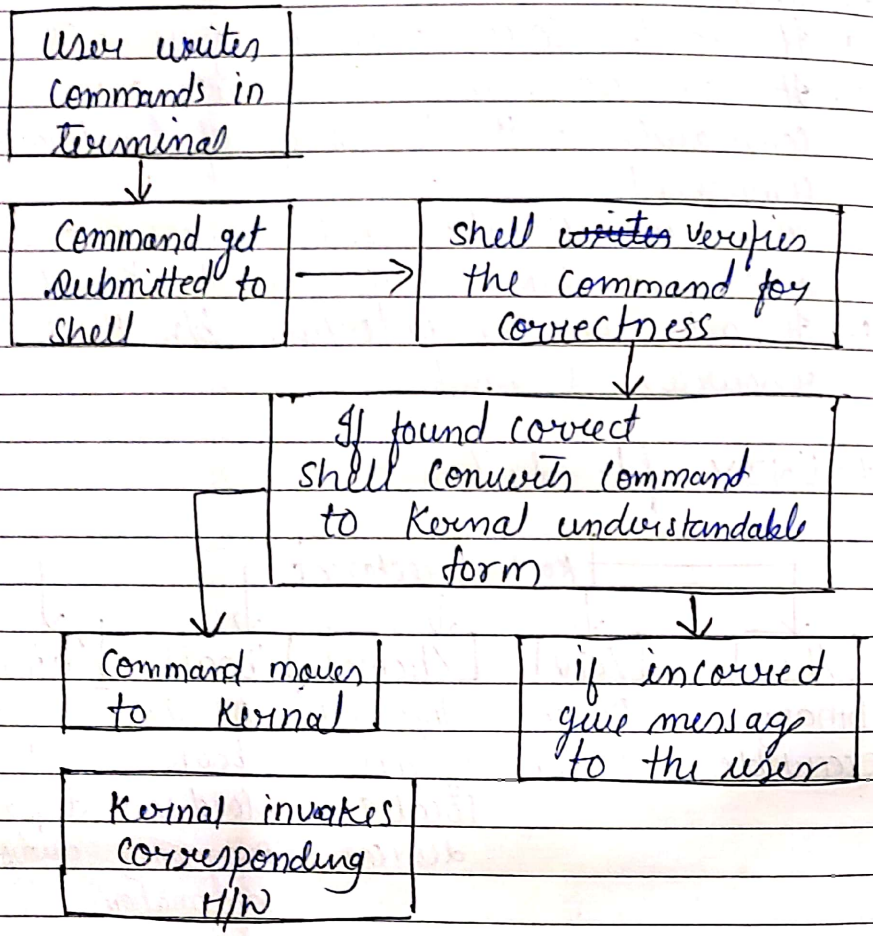
UNIX file system :-



Root directory :- symbol :- /

Command Execution flow

$ = user , # = admin



Commands :-

① Change directory :- cd /

② To check how many files or directory are there :- LS

③ lib 32 / lib 64

④ for going to any directory :-

cd folder_name

ex :- cd bin

⑤ for going back to root directory

cd ..

⑥ To clear consol :- clear

⑦ To Know about the info of particular file or directory

man - manual pages

ex :- man ls

⇒ for getting out of it :- q

⑧ To copy the files and directory :- CP

⑨ To see calender :- Cal

⇒ cal -y = To see whol year calender

⑩ To see date :- date

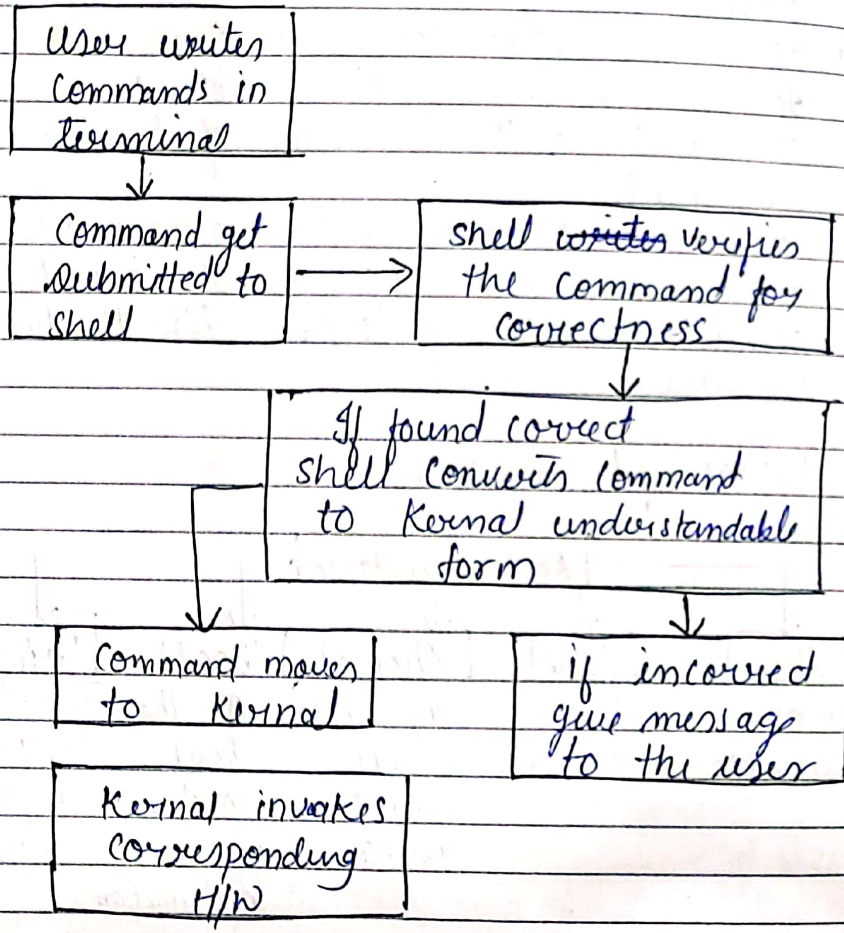
(i) date +%.a = Current day in short form

(ii) date +%.A = Current day in full form

(iii) date +%.b = Current month in short for

(iv) date +%.B = Current month in full form

$ = user , # = admin



Commands :-

① Change directory :- cd /

② To check how many files or directory are there :- LS

③ lib 32 / lib 64

④ for going to any directory :-

cd folder_name
ex :- cd bin

⑤ for going back to root directory
cd ..

⑥ To clear consol :- clear

⑦ To Know about the info of particular file or directory
man - manual pages
ex :- man ls
⇒ for getting out of it :- q

⑧ To copy the files and directory :- CP

⑨ To see calender :- cal

⇒ cal -y = To see whol year calender

⑩ To see date :- date
(i) date +%.a = Current day in short form
(ii) date +%.A = Current day in full form
(iii) date +%.b = Current month in short for
(iv) date +%.B = Current month in full form

>= output redirectional operator
<= Input redirectional operator

(11) Cat :- • To create a file (cat > mca)
Cat < mca ← • Reading / Veiwing the content of file
Cat >> mca ← • appending data to existing file
(By default txt file is created)

(12) Ctrl+d = To save the file

(13) To find No. of lines in file -
Cat -n mca

(14) g++ file.cpp = to invoak the compiler, to run the Cpp file
28/02/2023

file & permission and Access mode

⇒ ls -l file_name :- for viewing all the permission for that file.

★ there are three types of permission/ Access modes
1. Read 2. write 3. execute

★ there are three type of people
1. Owner 2. group 3. other (entire world)

⇒ The permission is written in 9 character and they are 3 character in bunch

owner (u) group (g) other (o)

ex:- rwx --x r--
owner group other
r = read, w = write, x = execute

Changeng Permission
there are two ways
↳ changing permission through Symbolic mode
↳ changing permission through absolute mode

→ Symbolic mode :-
• + :- Adding permission to existing
• - :- Removing existing permission
• = :- Assigning permissions

⇒ Chmod :- to change the mode of acess
ex:- chmod MCA 4 O+wx

before changing	After changing
rwx -w- r--	rwx -w- rwx
u g o	u g o

★ we can also set multiple permission
chmod O-wx, g+rx, u-rx MCA 4

→ Absolute mode :- Permission are represented using Octal number

Base = 8

0 = No permission ---
1 = Execute permission --x
2 = Write permission -w-
3 = Execute, write permission -wx
4 = Read permission r--
6 = Read, execute permission r-x
7 = Read, write, execute rwx
5 =

⇒ chmod 724 MCA4
owner → 7, group → 2, others → 4

Copying file

⇒ Creating a file
cat > file.txt
"This is file 1" : Use ctrl+d to save & exit
cat file.txt
cp file.txt file2.txt : copying a file

Moving a file

mkdir move_command
cat > file1.txt
this file should be moved
mv file1.txt /home/ubuntu/Desktop/file2.txt

Removing a file
↳ rm filename, file2, file3
↳ rm * :- to removing all files in the directory.

changing Ownership
there are two commands
↳ chown :- to change the owner.
↳ chgrp :- to chang the group.

Creating new users
↳ adduser :- to create a new user
↳ su :- switch user
↳ Sudo adduser :- super user do the work that adduser can't do will be done by Sudo adduser.

★ there are two kinds of user
- Your user
- root :- It has administrative setting

★ the works that simple user (add user) can't do :-
1. adding a new user
2. deleting a user and some more that's why we use 'Sudo'

★ this is the interface that will come when we create a user.
full name :-
Room no. :-
work Address :-
Address :-

★ Switching to another user :-

SU :- ubanty su mca 4
current user ↵ ↳ the user in which you want to switch

★ Sudo apt get - update
⇒ If the current user doesn't have sudo permission then "mca4 is not in sudoers list" (current user) will be shown

★ to make any user as Sudo user
Sudo visudo (then press enter, then it will show all the sudo user)

★ for deleting a user :-
Sudo userdel mca4 (after this it'll ask for a password that you have given while creating this user, then it'll delete the user)

★ to changing the ownership of a file from one user to another

⇒ to check current user :- ls -l mca4
then, ↗ new owner
chown mca4 demo.txt
this is how you can change the current owner

★ The UNIX command su stands for 'Substitute user' 'Switch user' or 'Super-user' and allows you to log in as root and do whatever you want with the System. Sudo stands for either "Substitute user do" or "Super user do" and it allows you to temporarily elevate your current user account to have root privilages.

UNIT IIIrd 01/04/2023

filter :- In unix filters are the set of commands that takes input from standard input stream i.e stdin, perform some operation and write output to standard output stream

i.e stdout.

The stdin and stdout can be managed as per preferences using redirections and pipes

Common filters commands are:- grep, more, sort

⇒ Group Command :- It is a pattern and expression matching command It searches for a pattern or regular expression that match in files or directories and their fount how much.

Syntax:- $ Grep [options] "Pattern to be matched" fi

Example:- input : $ Grep 'Hello' ist_file.txt.

o/p :- Searches Hello in the ist_file.txt

and output / returns the line containing Hello

The options in grep commands are:-

S.no	option	Description
1	-v	Returns all the lines that not matches the specify regular expression
2	-n	Returns all the lines that matched the specify regular expression with line number
3	-l	Return only names of the filter matching the specified regular expression
4	-c	Returns the count of lines that match regular expression
5	-i	It is case sensitive option matches either upper case or lower case

grep command can also be used with meta character

$ Grep 'Hello' *

o/p :-

It searches for Hello for all files or directory. * is a meta character and returns matching 0 or more preceeding character

⇒ Sort :- It is a data manipulation command that sort or merge lines in a files by specified fields. In other words, it sorts lines of a text alphabetically or numerically by default sort is alphabetically

Syntax :- $ Sort [option] filename
example :- $ sort fruit.txt
Table :-

Sno	option	Descriptions
1.	-n	It sorts the lines in numericals order
2.	-r	It reverses the order of sorting ex :- If the line were sort A to Z then it sort them Z to A
3	-f	It sorts upper case & lower case together
4	+x	It doesn't consider 1st fields while sorting

example of sort Cmd :- $ sort -n Grades.txt

⇒ More :- It is used to customized the displaying contents of the file It displays the txt file contents on terminal with pageng control following key controls are used

- To display next line, press the Enter key
- To Bring up next screen, press the Space Bar
- To move to the next file, press N
- To quit, press Q

Syntax :- $ More [options] filename
ex :- $ More fruit.txt

↳ Unix text processing commands are divided into three parts :-

- Unix filter
- Unix pipes
- More filter command lik awk & sed

⇒ Uniq :- Read from standard input command that are different from the adjacent lines to standard output

⇒ Cat Command :- Read line from standards input & Concentrate from output

⇒ Cut Command :- Cut specified byte, char, field from each line of stdin and print to stdout.

⇒ paste command :- Read lines from stdin and paste them together by line by line to stdout

⇒ Head Command :- Read the first few lines from stdin and print them to stdout

⇒ Tail Command :- Read the last few line from stdin and print them to stdout

⇒ WC Command :- Read from stdin and print the no. of new lines, words & Bytes to stdout

⇒ Tr Command :- Translate & delete characters from stdin and print to stdout.

05/04/2023

Assignment Question

Q1) Write shell commands in Unix (min 10)
Q2) What do you mean by command line explain each
Q3) Write shell script control statement
Q4) Write short note on
(i) Programmable filter sed
(ii) Loops (iii) Branching
(iv) Test statement
(v) Backup on files
Q5) Explain VI editor

Pipe Command ('|') :- You can connect two commands together so that the output from one program become the input of the next command. Two or more commands connected in this way forms a pipe

To make a pipe put a vertical bar (|) on command line b/w two commands.

When a program takes its input from another program, it performs some operation on that input and writes the result to the standard output. It is refered to as filter.

A pipeline is a mechanism for interprocess communication using message passing. A pipeline is a set of process changed together by their standard stream so that the output text of each process (standard output or stdout) is passed directly as input (stdin) to the next one.

for example :- To list files in a current directory (ls) retain only the lines of ls output containing the string "key" (grep), and view the result in a scrolling page (less), a user types the following into the command line of a terminal
$ ls -l | grep key | less

⇒ egrep :- It is a pattern searching commands which belongs to

Date: / / Page no:

the family of Grep function it works the same way as grep -E does. It treats the pattern as an extended regular expression and prints out the lines that matched the pattern If there are several files with the matching pattern, it also displays the file name for each line

Syntax :- egrep [options] 'PATTERN'

The egrep command used mainly due to the fact that is faster than the grep command. The grep command treats the meta character as they are and do not require to be escaped as is the case with grep egrep is faster than the grep and fgrep.