Operating system concepts (60M)

Theory → 28
Lab → 20
Internal → 12

⊙ To learn Operating system
1) Good user level (learn linux commands)
2) admin level (shell script programming)
3) Programmer level (system calls)
4) design level / internals (mem. management, CPU scheduling)

⊙ What is Linux?
↳ Family of open source UNIX like OS based on linux kernels

⊙ what is an OS?
↳ interface betn user & Hardware as user can not directly interact with any computer hardware & as provides interface for interaction
↳ it is system software (i.e collection of system prog.)

↳ Basic funcⁿ comp.
1) Data storage
2) Data Processing

User [computer] ⟷ OS ⟷ Machine Hardware [CPU, Memory Devices, IO Devices]

⊙ Software is collection of programs

⊙ programs → set of instructions
1) user prog. : main.c, sll.java, prog.cpp etc
2) application Prog: notepad.exe, calculator, MS office, google chrome
3) System prog: Inbuilt prog. of OS
eg. kernel, scheduler, memory, manager, loader, dispatcher, etc.

Scanned by CamScanner

Command

- uname → gives system information

- CUI/CLI : user can interacts with OS by means of ~~inter~~ entering commands in text format

CLI / Command line Interp.

- shell : it is an application prog. which acts as CLI, which accepts command from user in text format, interpret it to kernel for an execution

  In Linux 2 types of shell

  1) CUI shell : terminal

     eg. bsh, bash, csh, ksh, etc.

- GUI : user interacts with OS by making events like right click, double click, menu, start window, etc.

  Linux : GNOME / KDE

  Windows : explorer.exe

  GNOME : it shell prog. in Linux provides Graphical user interface

USER (computer
CLI/CUI (command User Interface)
GUI (Graphical User Interface)
Operating system
Hardware

Scanned by CamScanner

⊙ OS software (collection of thousands of system prog., application prog. and utility prog. which are in binary format) onto HDD

→ Linux kernel vmlinuz is exists in /boot dir in a binary format

→ while installing Linux we need to create 2 partitions:

1) swap partition ⇒ it is a portion of HDD kept reserved by Linux OS ~~while use~~ will be used as swap area, installation will be used as swap area (swap area is a portion of HDD used by an OS an an extention of main memory in which inactive running programs can be kept temporarily)

2) data partition / root partition (/) partition: in which Linux OS keeps / stores data in an organised manner

> pwd : print working directory (absolute path of present working directory)

> whoami : prints name of currently logged in user into system on terminal

> ls : displays list directory contents → sub dir & files

> clear : to clear screen [ctrl + l]

> top : to display information of running prog. / processes [press q] to get out of top command

↳ similar like task Manager of Windows

Scanned by CamScanner

$ uname : To display system info. (eg. version etc)
$ uname -a : All system info
$ uname -v : to display only system of version

$uname → o/p: Linux
$uname -a → o/p: Linux —— ubuntu ——
(os kernel version distribu, etc. mentioned)

$ man uname : Print system info,
man means manual like a guide

$ top : like task manager in windows gives all running prog. & CPU utilisation

$ ls : Content in directories (in Home directory)

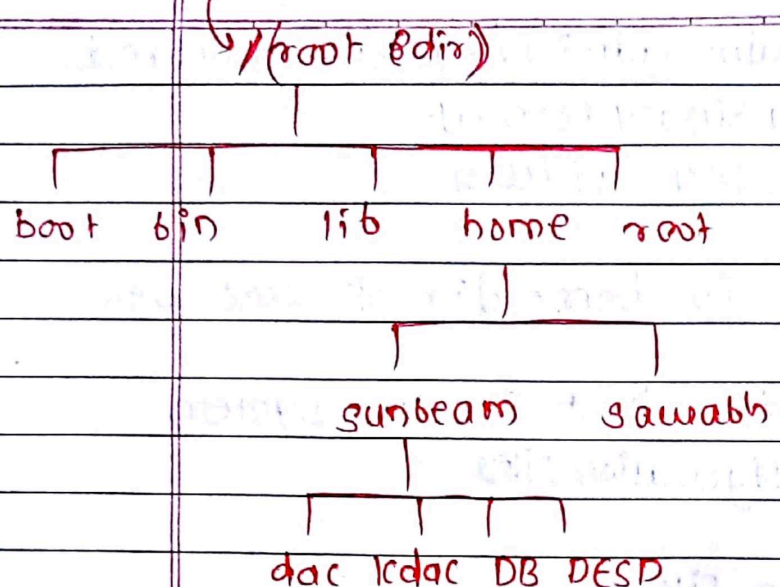
$ pwd : o/p: /home/osboxes
↳ I have currently logged in using username osboxes

$ whoami : o/p: osboxes
Prints effective username

$ clear : clear screen shortcut key [ctrl + l]

Scanned by CamScanner

/ mean root directory



All the data will be kept under /root directory in hierachical manner

$ cd /<dirpath> : To change directory
(cd /<dirpath>) used for navigation through file system (one directory to other)
↓
give space

eg. $ pwd → o/p: /home/osboxes
$ cd / → o/p: I will reach in /root direct.
/$ ls → o/p: bin lib lib64 mnt proc usr boot tmp var home media dev

/boot → info about booting system
↳ contains linux kernel (vmlinuz)

/bin → Contains user commands in binary format eg. ~~lscpu~~, ~~adduser~~
cat, cp, mv, etc

/home → contains home dir's of all users in which can store data, can read & write as well execute perms in that dir
(home direct. of system)

(upon creation of new user, sub directory by username gets created under home directory & that user can write in that dir

/sbin → contains admin/system commands in a binary format
eg. lscpu, adduser

/root → root is home dir of root user

/etc → contains host specific system configuration files

/dev → device files

/lib → shared libraries required for /bin & /sbin

/mnt → contains temporarily mounted filesystem

/media → mount point for removable media such as CDROM's

/opt → optional application software packages

/proc → virtual filesystem provides process & kernel info. as files
In linux, corresponds to a procfs mount.
Generally automatically generated & populated by system

Scanned by CamScanner

⊙ Filesystem → a way to store data onto disk/partition in an organized manner so that it can be accessed by system effectively

⊙ Path ⇒ unique location of any filesystem structure presented in string format. seperated by delimiter
- ↳ slash (/) in linux
- ↳ Backslash (\) in windows
- ↳ colon (:) in linux, windows

① Absolute path : path w.r.t to root directory ("/"). i.e it is full path which starts with root dir

② relative path : path of current working directory

ls /bin → absolute path
ls bin → relative path

⊙ shell Metacharacters
- / (root dir)
- ~ (current logged in user's home dir)
- . current dir
- .. parent dir

$ cd / ⇒ go to root dir
$ cd ~ ⇒ go to user's home dir
$ cd . ⇒ goto current dir / remains in curr. dir. only
$ cd .. ⇒ go to parent dir of current directory

Scanned by CamScanner

- **mkdir** : used to create new dir / directories
  $ mkdir <dirpath> eg. $mkdir newFile

- **rmdir** : to remove / delete the directory
  eg. $ rmdir newFile/
  ↳ using rmdir we can delete only empty directory

- **cd -** : (cd space hyphan) to goto previously accessed directory

- **mkdir -p one/two/three** : dir by name 'one' get created parent dir in a mentioned dir, then sub-dir 'two' get created inside dir one & then sub-dir 'three' get created inside dir two
  ↳ hyphan (-) p : to create directory hierarchy

- **cd ../..** : goto parent dir of parent directory of current directory
  eg. home/osboxes/one/two $ cd ../..
  ↳ parent of two → one
  ↳ parent of one → osboxes
  ∴ o/p: home/osboxes

- **rm** : remove file / files → to delete non-empty dir
  $ rm -R dirpath
  ↳ it deletes first contents of all sub dir's & file of ~~in it~~ dir recursively in it & then delete dir

  eg. $ rm -R one/

Scanned by CamScanner

- **ls -R** : shows hierarchy of current dir
  eg. /newfile $ mkdir -P one/two/three/four
  /newfile $ ls -R
  o/p: one

  ./one:
  two

  ./one/two:
  three

  ./one/two/three:
  four

  ./one/two/three/four

- **ls** : By default it displays content of cur dir in a columnwise alphabatically sorted manner

- **ls -1** : it displays contents of cur dir in alpha sorted manner but one line per entry

- **ls -l** : longlist format of contents in dir
  total n n → total no. of data blocks allocated for files

- **ls -s** : total no. of data blocks

Scanned by CamScanner

- In linux/unix based OS, 7 types of files:
  1) regular file (-) : all text, source files, etc.
  2) directory file (d) : name of sub-dir's
  3) character special device files (c)
  4) block special device files (b)
  5) linkable file (l)
  6) pipe file (p) } interprocess communication
  7) socket file (s) }

- In linux/unix based OS devices can be categorised into two categories:
  1) character devices : devices from which data gets transfer char by char i.e. byte by byte
     eg. keyboard, printer, etc
  2) block devices

col 1 : Type

- rwx → read write execute permission

eg. drwx rwxr-x

- d → directory file
- rwx → curr. directory can read write execute
- rwxr → group mem. can read write execute
- -x → others can read & execute, no write

col 2 : Link counts : no. of links in file system
i.e How many ways can access file

By default → Reg. file = 1
↳ direct. file = 2

Scanned by CamScanner

- **ln -s <src-file-path> <dest-file-path>**
  → used to create symbolic link (shortcut)
  i.e linkable file for an existing file which is located deeply into filesystem

- man → to display guide/manual of commands

- whatis → display one line description of command from man pages
  eg. $whatis ls

- **cat**
  → cat command concatenates contents of file and display onto console
  → cat command also used to display contents of regular file
  → cat command also can be used to write data into file as well as append data into file

- **$cat > filename.text** → to create a file & write contents into that file & press [ctrl+d] to stop file writing

- $ cat filename.txt → display contents of file

- cat command cannot be used for file editing

Scanned by CamScanner

- By default any prog. reads input from stdin file
  by default any prog. writes output ~~from~~ into stdout file
  errors → stderr file

- Linux Commands
  1) internal commands / inbuilt commands
  → commands whose logic is written into shell prog. itself
  no seperate binary exists for an internal commands.
  eg. cd, exit
  2) External commands
  → commands whose logic exists into seperate binaries which can be executed by shell
  eg. ls, clear, mkdir, cp, mv etc

- touch : used to update time stamp of file
  ↳ By using touch command empty file also can be created

- cp : copy file/files from one location to another
  $ cp <src-dir-path>/file(s) <dest-dir-path>
  ↳ may be absolute path or relative path

- mv : move file/files from one location to another
  $ mv <src> <dest>
  if src and destination are same then mv command used for rename purpose

Scanned by CamScanner

- ctrl + U → whole command line erased
- ctrl + W → only word erased
- lscpu → info about CPU architecture.
  cd / proc/cpuinfo

- echo $HOME → o/p: /home/osboxes
- pwd → o/p: /home/osboxes
  ~~echo is shell variable~~
  ↳ pwd : it contains absolute path of present working dir
  HOME : it contains absolute path of user's home dir of currently loggedin user
  Command Name : echo → to print value of variable
  $HOME is variable here

- echo $USER → o/p: osboxes (current user)
- echo $SHELL → o/p: /bin/bash (current working (absolute path of curr Active shell)

- stat → display info. ~~of~~ about File & Filesystem
  ↳ stat command reads info. from INODE of File & display it.
- USER : contains name of currently logged in user
- PATH : contains paths of binaries of user commands, system commands etc.

- stat filename → display info about File
- stat -f filename → display info about Filesystem

Scanned by CamScanner

- All linux commands (external commands) are nothing but programs only which are in binary format if we want to execute any binary either we can use its absolute path or relative path

- **wc** → word count → used to count no. of lines, no. of words and no. of characters

- Pipe command **(|)** → used to pass an output of one prog. / command as input to another prog. / command

- we want to count no. of lines, no. of words & no. of characters in an output of ls -l command

```
$ ls -l > out.txt
$ wc < out.txt

$ ls -l | wc
```

Scanned by CamScanner

- lscpu → gathers cpu architecture info from /proc/cpuinfo
- $ cd /proc → $ cat cpuinfo
- to execute current directory binary file
  ./emp.out
  → we can execute binary by using its relative path
  OR
  → we can execute binary from any location with help of absolute path
  /home/sunbeam/feb2022/DBDA/employee/emp.out
  (complete path)
- $ file ls : info. of ls file (/bin/ls) → abs. path
- $echo $PATH : shell variable path
  in this shell variable multiple paths of binaries stored seperated by :
- printenv ⇒ to print environment variables of shell variable
- read ⇒ to scan/read value of variable of any type from user through terminal
  $ read name $read num
  $ echo $name $echo $num

Scanned by CamScanner

⊙ Redirection operators

1) input redirection (<) left chevron
2) output redirection (>) righ chevron
3) error redirection (2>)

1) we can pass contents of any file as input to to any prog.
(Prog. can take i/p from specific file) instead of stdin
2) prog. can write output into specific file instead stdout
3) list of errors into specific file instead stderr

eg. ls -l > out.txt
↳ this file redirected perpanantly

⊙ cat > india.txt
India is my country
I love my India
(ctrl+d)
] 2 lines 8 words

⊙ wc < india.txt
// to wc command we are giving input from india.txt file

o/p : 2 8 33
line words char

If I want to store this o/p permanently in some file → ~~use pipe operator/ command~~
$ wc < india.txt > wc.output.txt

Scanned by CamScanner

- count no. words, lines, characters of output of ls-l

① $ ls -l > out.txt
$ wc < out.txt

② $ ls-l | wc using pipe command we are giving output of ls-l comm. to wc command as input

- scroll up : [ctrl + shift + up]

- $ ls -l -a | less (use down & up key only)
  pipe q → to quit

- Sort number.txt (sorting as per ASCII value)

- sort -n number.txt (sorting numerically)

- sort -n number.txt | uniq (to apply uniq comm. content need to be sorted)
  ⇒ we get only distinct value / no duplicate values

- sort -n numbers.txt | uniq > uniq_numbers.txt
  o/p redirected to this file

- cat numbers.txt | head ⇒ display first 10 values

cat numbers.txt | head -5 ⇒ display first 5 lines

cat numbers.txt | tail -5 ⇒ display last 5 lines

Scanned by CamScanner

By default last 10 values

cat numbers.txt | tail | head -5 (last 5 values)

① cat > records.csv (writing contents) (CSV means comma seperated files)

1, sachin, mumbai, 111
2, saurabh, pune, 222
3, preetam, pune, 555
4, Rani, Beed, 777

cat records.csv (displaying contents)

② cat records.csv | cut -d ",," -f1 comma Field 1

↳ delimiter

↳ cutting vertically contents of file on basis of ","

O/P: 1
2
3
4

cat records.csv | cut -d ",," -F2

O/P: sachin
saurabh
preetam
Rani

Scanned by CamScanner