

# LINUX - OPERATING SYSTEM

- ⇒ Operating System (OS) is a software that acts as an interface between computer hardware components and user.
- ⇒ Every computer must have atleast one operating system to RUN other program.
- ⇒ Operating System providing platform/environment to RUN our applications.  
Ex:- Browsers, Notepad, MS Paint, calculator etc.
- ⇒ Operating Systems helping us to communicate with computer without knowing how to talk to computer language.  
Note:- We can't use a computer without having operating system.



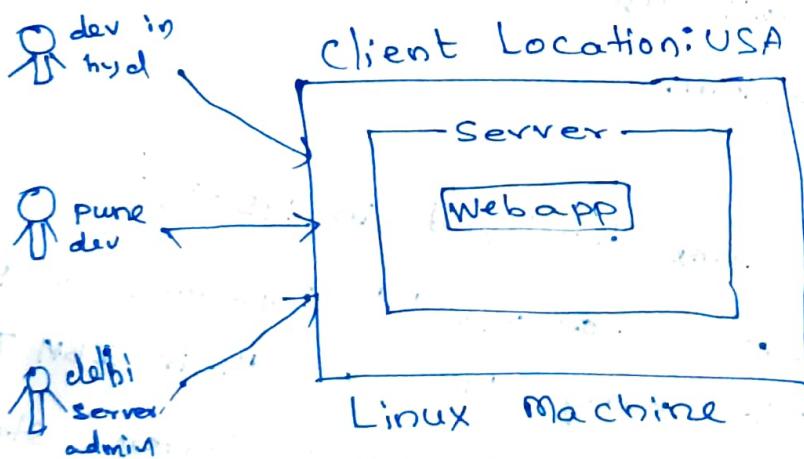
hardware component

- ### History OS
- ⇒ Operating Systems developed in 1950's.
  - ⇒ General Motors research lab implemented the first OS in the early 1950 for their IBM 701.
  - ⇒ In the mid. of 1960's, operating systems started using disks.
  - ⇒ In the late 1960's, the version of UNIX operating system developed by <sup>(1981)</sup> Microsoft is MS DOS.
  - ⇒ The first OS developed by <sup>(1985)</sup> with GUI is released.
  - ⇒ In 1985 operating system released with Graphical User Interface.

- ⇒ we are using windows operating system in our computer / laptops
- ⇒ windows OS is recommended for personal use
- ⇒ windows is single user based operating system
- ⇒ windows is commercial software.



- ⇒ In realtime environment, we will use LINUX OS to run our applications
- ⇒ Linux is free & open source
- ⇒ Linux is multi user Based operating System
- ⇒ Linux is highly secured operating System.



- Multiple users can connect to Linux machine at a time that's why it is called as Multi User Based operating System.
- ⇒ cloud Engineers will setup Linux Machines in cloud
  - ⇒ DevOps engineers will deploy our applications in the servers which are running in Linux OS.
  - ⇒ Developers will connect to Linux machine to check server/application log.

\$ sudo su - maddius

class-2

## Linux - Commands part 2

Q What is operating system?

- \* Operating System is acting as mediator between user and computer hardware components.
- \* Operating system is mandatory to use any computer
- \* OS provides environment to run other applications  
(browser, notepad, paint, calc)
- \* The OS came into market in 1950
- \* Microsoft released it's first in 1981 (MS-DOS)

### Windows OS

- ⇒ Developed by Microsoft
- ⇒ It is having GUI
- ⇒ It is commercial (paid)
- ⇒ Less security
- ⇒ It is recommended for personal use.

### Linux OS

- ⇒ Linux is community Based OS
- ⇒ Linux is free & open source
- ⇒ Linux is multi user Based OS
- ⇒ High Security
- ⇒ Recommended to use for Application, Servers, Database etc

Note:- In realtime, we will use Linux OS only to setup infrastructure required to run our application.

- ⇒ Linux OS is not only for administrators, even developer and testers also ~~use~~ will use Linux OS in realtime to monitor our application and application servers.

## History of Linux

⇒ In 1991, a student 'Linus Torvalds' developed this

Linux OS

- ⇒ Linus Torvalds identified some challenges in UNIX OS and he suggested some changes for UNIX OS but UNIX OS Team rejected 'Linus Torvalds' suggestion
- ⇒ Linus Torvalds used MINUX OS to develop Linux

Linus + MINUX

⇒ first two letters from his name and last 3 letters from MINUX OS

LI + NUX ⇒ LINUX

- ⇒ Linus Torvalds released LINUX OS with source code into market so that anybody can modify LINUX OS, or that's why it is called as open source OS.
- ⇒ As Linux OS is open source, so many people and companies taken that Linux OS and modified according to their requirement and released into market with different names those are called Linux Distributions.

RHEL -- RED HAT

Ubuntu OS

Cent OS

Fedora

Open SUSE

Kali Linux

Debian

NOTE:- 200+ Linux Distributions are available in the market

## Environment Setup

→ We can setup Linux Machine in 2 ways

① By using Hypervisor with Virtual Box

② Cloud VM

→ Documentx shared to create AWS account &

Linux VM Launching in AWS ..

[follow the given doces and complete the assignment)

-----  
① Create account in AWS (it will ask debit/credit card)  
- selected cards are accepted

② Create Linux VM using AWS EC2 service (download key pair)

③ Convert .pem to ppk using puttygen software

④ Open putty S/W and connect to EC2 VM using IP&ppk

-----  
Linux VM is ready. It is up and running in AWS cloud

We are able to connect with Linux VM using Putty S/W

## Linux Commands

whoami: It will display currently Logged in Username

pwd: present working directory

date: To display current date

cal: To display calendar

⇒ In Linux everything will be represented as file

⇒ we have 3 types of in Linux.

① Ordinary File/Normal file (it will start with -)

② Directory File (it will start with d)

③ Link File (it will start with l)

⇒ The file which contains data is called as ordinary file

⇒ Directory file is equal to folder (it can contain files and files)

⇒ The file which is having linking is called as Link file

⇒ touch: It is used to create empty file

\$ touch f<sub>1</sub>.txt

\$ touch f<sub>2</sub>.txt

\$ touch f<sub>3</sub>.txt f<sub>4</sub>.txt

⇒ ~~ls~~

⇒ ls: To display files we will use 'ls' command  
\$ ls

⇒ cat: To create a file with data we will use 'cat'

\$ cat > hello.txt

// write data

Press CTRL+d (To save and exit)

~~\$ cat > hello.txt~~

\$ cat hello.txt [To display file data]

\$ cat >> hello.txt (To append data in the file)

// write data

Press CTRL+d (to save and exit)

⇒ mkdir: To create directory we will use 'mkdir' command

\$ mkdir dirname

⇒ rm: To remove the file we will use 'rm' command

\$ rm filename

⇒ rmdir: To remove empty directory we will use 'rmdir'

\$ rmdir dirname

⇒ ls is used to list out all files & directories available in the given directory

Note: We can pass several options for 'ls' command

- ⇒ `ls`: It will display all files in alphabetical order (a to z)
- ⇒ `ls -r`: It will display all files in reverse of alphabetical order (z to a)
- ⇒ `ls -l`: It will display long listing of files.
- ⇒ `ls -t`: It will display all files based on last modified date and time. Most recent file will be displayed at top and old files will display at bottom.
- ⇒ `ls -rt`: It will display all files based on reverse of last modified date and time. Old files will display at top and recent files will display bottom.
- ⇒ `ls -a`: It will display all files including hidden files (hidden files will start with .)
- ⇒ `ls -li`: It will display files with inode.
- ⇒ `ls -LR`: It will display all files and directories along with sub directory's content

Note:- -R represents recursive.

Note:- we can use several options for ls command at a time. When we are using multiple options order of those options is not important

\$ `ls -ltr`

\$ `ls -tlr`

\$ `ls -l -t -r`

\$ `ls -rtl`

Note:- All the above commands will give same output

⇒ To display content of given directory we can execute like below

\$ ls <dirname>

⇒ To delete a file we will use 'rm' command

\$ rm <filename>

⇒ To delete empty directory we will use 'rmdir' command

\$ rmdir dirname

⇒ To delete non-empty directory we will use 'rm' like below

\$ rm -r dirname

⇒ To display file content we will use 'cat' command

\$ cat filename

⇒ To display file content with line numbers we will use '-n' option

\$ cat -n filename

⇒ To display multiple files content at a time execute command like below.

\$ cat file1 file2 file3

⇒ Copy one file data into another file using 'cat' command

\$ cat f1.txt > f8.txt

⇒ Copy more than one file data into another file

\$ cat f1.txt f2.txt > f9.txt

### Reversing file content

⇒ 'tac' command is used to reverse file content

\$ tac filename

⇒ 'rev' command is used to reverse each line content of the file

\$ rev filename

## head Command

⇒ head command is used to display file data from top (default 10 lines)

\$ head filename

\$ head -n 5 data.log (first 5 lines data)

\$ head -n 20 data.log (first 20 lines data)

## tail command

⇒ tail command is used to display file data from bottom (default 10 lines)

\$ tail filename (last 10 lines data)

\$ tail -n 100 filename (last 100 lines data)

\$ tail +25 filename (it will display data from 25th line to bottom)

Note: To see on-growing logs we can use -f option

\$ tail -f data.log (live log message we can see)

## WC Command

⇒ It is used to count no. of lines, no. of words and no. of characters in the file

ubuntu@ip-172-31-47-242:~\$ wc -l f1.txt

2 8 45 f1.txt

## class - 5

- When application running, it will generate log messages and it will store log messages at bottom of the file.
- To see log messages of the application we will use 'tail' command.

### CP command

- To copy the data from one file to another file
- ```
$ cp one.txt two.txt (or) $ cat one.txt > two.txt
```
- \$ cp f<sub>1</sub>.txt f<sub>2</sub>.txt f<sub>3</sub>.txt (invalid syntax)
- ⇒ we can't copy more than one file data using 'cp' command.

To copy multiple files data we should go for 'cat' cmd

```
$ cat f1.txt f2.txt > f3.txt.
```

### Rename the files (or) directories

⇒ To rename files we will use 'mv' command

```
$ mv f1.txt f11.txt
```

```
$ mv dirname dirnewname
```

Note:- We can use 'mv' command for renaming

### Comparing files

⇒ To compare files we can use below commands

```
$ cmp f1.txt f2.txt
```

```
$ diff f1.txt f2.txt
```

⇒ cmp command will display only first difference in given 2 two files.

⇒ diff. command will display all the differences in the content.

## Grep Command

⇒ 'grep' stands for global regular expression print.  
⇒ 'grep' command will process the text line by line and prints, any lines which matches given pattern.

Ex:- I want to print all lines with contains 'NullPointerException'

```
$ grep -i 'NullPointerException'
```

Note:- We can install Grep using below command  
\$ sudo yum install grep

//Search for the lines which contains given word  
in the given filename

```
$ grep 'word' filename
```

//Search for the lines which are having exception  
keyword in server.log file

```
$ grep -i 'exception' server.log
```

//Search for the given text in present directory and  
in sub-directories also

```
$ grep -R 'exception'
```

⇒ We can pass several options for 'grep' command

-c: This prints only the count of files that matches  
give pattern

-i: ignore case-sensitivity

-n: Display the matched lines and their line number

-l: Display only file names that matches the pattern

-b: Displays matched lines without file names

-R: Displays matched lines with file names.

## Working with Text Editors in Linux

(Class - 6)

The default editor that comes with the UNIX operating system is called vi (visual editor).

\* Using vi editor, we can edit an existing file (or)

Create a new file from scratch.

\* We can also use this editor to just read a text file.

Modes of operation in vi editor :-

There are three modes of operation in vi:

- ① command mode
- ② insert mode
- ③ escape mode

Command mode:-

When vi starts up, it is in command mode. This mode is where vi interprets any characters we type as commands and thus does not display them in the window.

\* This mode allows us to move through a file, and to delete, copy or paste a piece of text.

\* To enter into a command mode from any other mode, it requires pressing the [esc] key.

If we press [esc] when we are already in command mode, then vi will beep or flash the screen.

Insert Mode:-

\* This mode enables you to insert text into the file.

\* Everything that's typed in this mode is interpreted.

as input and finally, it is put in the file.

- \* the vi always starts in command mode. To enter text, you must be in insert mode.

- \* To come in insert mode you simply type i.
- \* To get out of insert mode, press the Esc key, which will put you back into command mode.

### Last Line mode (Escape mode)

- \* Line mode is invoked by typing a colon [:], while vi is in command mode.
- \* The cursor will jump to the last line of the screen and vi will look for a command.
- \* This mode enables you to perform tasks such as saving files, executing commands.

There are following ways you can start using vi editor

### Commands and their description

- = vi filename: Creates a new file if it already not exist, otherwise opens existing file.
- = vi -R filename: Open an existing file in read only mode
- = view filename: Open an existing file in read only mode

vi file.txt

→ After making changes if we don't want to save those changes then execute :q!

### Moving within a file (Navigation) :-

To move around within a file without affecting text must be in command mode (press Esc twice).

Here are some of the commands can be used to move around one character at a time.

## Commands and their Description

K: Moves the cursor up one line.

J: Moves the cursor down one line.

h: Moves the cursor to the left one character position.

l: Moves the cursor to the right one character position.

O(m): Positions cursor at beginning of line

\$: Positions cursor at end of line.

W: Positions cursor to the next word.

B: Positions cursor to the previous word.

(: Positions cursor to beginning of current sentence

): Positions cursor to beginning of next sentence

H: Move to top of screen

nf: Moves to nth line from the top of the screen

M: Move to middle of screen

L: Move to bottom of screen.

nl: Moves to nth line from the bottom of the screen along with x: colon followed by a number would position the cursor on line number represented by X.

## Use Case:-

⇒ We will use vi editor to perform below actions

- ① To edit config files of our application
- ② To edit shell script files.

## Working with SED command

- \* SED command in UNIX stands for stream editor and it can perform on file like searching find and replace, Insertion (or) deletion
- \* though most common use of SED command is in UNIX.
- \* By using SED you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.
- \* SED is a powerful text stream editor. can do insertion, deletion, search and replace(substitution).
- \* SED command in UNIX supports regular expression which allows it perform complex pattern matching

### Ex:-

\$ cat > myfile.txt

unix is great OS. unix is open source. unix is free OS.

Learn operating system.

unix linux which one you choose.

unix is easy to learn. unix is a multiuser OS.

Learn Unix. Unix is a powerful.

### Replacing or substituting string:

SED command is mostly used to replace the text in a file. the below simple sed command replaces the word 'unix' with 'linux' in the file.

\$ sed 's/unix/linux/' myfile.txt

my default, the sed command replaces the first occurrence of the pattern in each line and it won't replace the second occurrence in the line.

Replacing the  $n^{th}$  occurrence of a pattern in a line

Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word "unix" with "linux" in a line.

```
$ sed 's/unix/linux/2' geekfile.txt
```

Replacing all the occurrence of the pattern in a line

The substitute flag /g (global replacement) specifies the sed command to replace all the occurrences of the string in the line.

```
$ sed 's/unix/linux/g' myfile.txt
```

Deleting lines from a particular file

SED command can also be used for deleting lines from a particular file. SED command is used for performing deletion operation without even opening the file.

To ~~d~~ Delete a particular line say n in this example

```
$ sed '5d' my file.txt
```

To Delete a last line

```
$ sed '$d' myfile.txt
```

To delete from  $n^{th}$  to last line

```
$ sed '12,$d' myfile.txt
```

Note: By default SED command changes will not store in file.

⇒ To make SED command changes to file permanently we use '-i' option.

\$ sed -i s/unix/linux/g' my file.txt

Note:- with above command 'unix' keyword will be replaced with 'linux' keyword in the file permanently.

## File permissions :-

To Create a secure environment in Linux, you need to learn about user groups and permissions. For example, If you work in a company and you want the finance department to read a file but not make any modification to it, then you ~~need~~ to use permissions in Linux. It is a must for every programmer working with Linux nowadays. Let's start by talking about the ownership of Linux files.

User: the owner of the file (Person who created the file)

Group: The group can contain multiple users.

Therefore, all users in that group will have the same permissions. It makes things easier than assign permission for every user you want.

Other: Any person has access to that file, that person has neither created the file, nor are they in any group which has access to that file.

⇒ Execute 'ls -l' command to file's permissions.

We will work this part "`-rw-r--r-`".

The characters mean:

'r' = read

'w' = write

'x' = execute

'-' = no permission

-rwxr--r--

\* It represents file

rwx: user

r: group

r: other

\* As we see above, the empty first part means that it is a file. If it were a directory then it will be the letter "d" instead.

\* The second part means that the user "Home" has read and write permissions but he does not have the execute one.

\* The group and others have only the read permission.

Let's change the permissions using the chmod command.

=> chmod o+wx section.txt

This command will add the write permission for other user to my text file "section.txt".

=> Now if you try to execute ls -l then you will see

-rwxr--rwx-

=> "o" refers to others, "g" for the group, "u" for the user, and "a" for all.

=> Now let's add the execute permission to the user

=> chmod u+x section.txt

The permissions will be -rwxr-xr--

If you want to remove the permission, you can use the same method but with "-" instead of "+".

\* for example, Let's remove the execute permission from the user by :

$\Rightarrow \text{chmod } u-x \text{ section.txt}$

and the permissions now are: -rwx-T-rw-

Also, you can use symbolic mode to modify permission like the following

Number      Permission

0            No permission

1            Execute

2            Write

3            Execute and write

4            Read

5            Read and Execute

6            Read and Write

7            Read, Write and Execute

for example, let's give every permission for all with:

$\Rightarrow \text{chmod } 777 \text{ section.txt}$

Then the permissions will be: -rwxrwxrwx.

let's remove the execute from the group and the write from

other by:

$\Rightarrow \text{chmod } 765 \text{ section.txt}$

then the permission will be: -rwxrw-r-x

Working with find and locate commands :-

The locate command find will search for data in bsd db

& sudo apt install mlocate

\$ locate apache

\$ locate -c apache

\$ locate -c \*.txt

\$ locate -S (to see locate database)

2. Find files under Home Directory

# find /home -name f1.txt

7. Find files with 777 permissions

# find . -type f -perm 0777 -print

19. Find all Empty files

# find /tmp -type f -empty

20. Find all Empty Directories

# find /tmp -type d -empty

Working with user accounts :-

Create a user

⇒ sudo adduser <sup>username</sup> testuser

After creating a new user and setting a password to it, you can log in two ways.

By the terminal : su - testuser

Delete a user

⇒ sudo userdel testuser

If you try that command, you will notice that the user directory has not been deleted and you need to delete it by yourself.

\* You can use this automated command to do everything for you:

\* sudo deluser --remove-home testuser

## User groups

A group is a collection of users. The primary purpose of group is to define a set of privileges like read, write, or execute permission for a given resource that can be shared among the users within the group.

## Create a group

A group is a collection of users. The primary purpose of the groups is to define a set of privileges like read, write, or execute permission for a given resource that can be shared among the users within the group.

## Create a group

You can see all of the groups you have by opening the following file:

You can see all of the groups you have by opening the following file

cat /etc/group

## Create a group

→ sudo groupadd section

## Add user to a group

→ sudo groupadd section → sudo usermod -aG section testuser  
→ Verify user groups using command → id username

## Delete user from a group

sudo groupmod -d testuser section

## Delete a group

sudo groupdel section