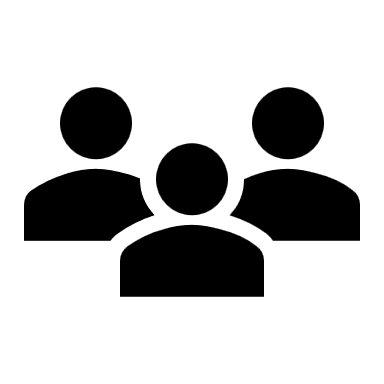
**Operating System?**

* Operating System (OS) is a software that acts as a interface between Computer hardware and user

Operating System



**Users Hardware Component**

* Every computer must have at least one operating system to run other programs.
* Operating System providing platform/environment to run our applications.
  + Examples: Browsers, Notepad, MS-Paint, Calculator etc.
* Operating Systems help us to communicate with computers without knowing how to speak computer language.

Note: we cannot use a computer without having an operating system.

* An Operating System is a software which will allocate the resources to my program for its successful execution. It will take the resources back once my program is completed.
* Operating System acts as a Resource Allocation & De-Allocation Manager

**History Operating System:**

* Operating systems developed in 1950’s.
* General Motors Research Lab implemented the first Operating System in the early 1950 for their IBM 701
* In the mid 1960’s, the version of the UNIX Operating System developed.
* The first operating system developed by Microsoft is MS DOS (1981)
* In 1985 operating system released with GUI (Graphical Based Interface)

**General Use:**

* We are using Windows operating system in our computers/laptops.
* Windows operating system is recommended for personal usage.
* Windows is a single user-based operating system.
* Windows is commercially available software.

A computer screen with black text

Description automatically generated

* In real time environments, we will use LINUX operating systems to run our applications.
  + Linux is Free & Open Source.
  + Linux is Multiuser Based Operating System
  + Linux is a Highly Secured Operating System.

A computer screen with arrows pointing to the screen

Description automatically generated

* Multiple users can connect to Linux machine at a time that’s the reason we called it a multiuser based operating system.

Activities performed by various members of the IT:

* Cloud Engineers will set up Linux Machines in Cloud.
* DevOps Engineers will deploy our applications to the servers which are running in Linux Operating System.
* Developers will connect to Linux Machine to check server/application logs.

Windows OS:

* Developed by Microsoft
* It is having GUI.
* It is a single user-based operating system.
* It is commercial, Less Security, it is recommended for personal use.

Linux OS:

* Linux is Community Based Operating System
* Linux is Free & Open Source
* Linux is Multiuser Based Operating System
* High Security
* Recommended to use for Applications, Servers, Databases etc...

History of Linux:

* In 1991, a student “Linus Torvalds” developed this Linux Operating System
* Linus Torvalds Identified some challenges in UNIX Operating System & he suggested some changes for UNIX Operating System, but UNIX Operating System rejected Linux Torvalds Suggestion.
* Linus Torvalds used Minux Operating System to Develop Linux
  + - * Linus + Minux
* First 2 letters from his name & last 3 letters from Minux Operating System.
  + - * LI + NUX => LINUX
* Linus Torvalds released LINUX OS with source code into market so that anybody can modify LINUX OS that’s why it is called as Open-Source Operating System.
* As Linux Operating System is Open-Source so many people & companies taken that Linux Operating System & modified according to their requirement & 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/VM Ware.
  2. By using Cloud Service

Approach-1:

A diagram of a computer program

Description automatically generated with medium confidence

Approach-2:

A computer and data center logo

Description automatically generated

1. Create an account in AWS!
2. Launch Virtual Machine in AWS

(Account is free for 1 Year)

AWS: Amazon Web Services

EC2 INSTANCES = ELASTIC COMPUTE CLOUD.

TO CREATE EC2 WE NEED TO PERFORM 7 STEPS:

SERVER = COMPUTER

1. TAGS = NAME
2. AMI = OPERATING SYSTEM, SOFTWARE PACKAGES
3. INSTANCE\_TYPE = CPU & RAM
4. KEY\_PAIR = LOGIN (public=aws, private=user)
5. NETWORK = VPC, SECURITY GROUPS (port numbers=0-65535)
6. STORAGE = 8 GB – 16 TB
7. SUMMARY = TO REVIEW

For Connecting EC2 Instances we have several options available. They are:

1. Git Bash
2. EC2 Instance Connect
3. By using SSH Client (Putty, Mobaxterm, Tabby,…..)

* Convert .pem to .ppk using puttygen software
* Open putty software and connect to EC2 VM using IP & PPK File.

Linux Commands:

whoami : it will display currently logged in username\

pwd : present working directory / print working directory

date : to display current date

cal : to display calendar.

In Linux everything will be represented as file

We have 3 types of files in Linux

1. Ordinary file / Normal file
2. Directory file
3. Link file

* The file which contains data is called as ordinary file
* Directory file is equal to the folder (it can contain files and folders)
* The file which is having linking is called as Link File

touch : it is used to create empty file

$ touch f1.txt

$ touch f2.txt

$ touch f3.txt f4.txt

To display files, we will use ‘ls’ command.

$ ls

To create a file with data we will use ‘cat’ command.

$ cat > hello.txt

// write data

Press CTRL + d ( to save and exit )

$ 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 )

To create directory, we will use ‘mkdir’ command.

$ mkdir dirname

To remove the file, we will use ‘rm’ command.

$ rm filename

To remove empty directory, we will use ‘rmdir’ command.

$ rmdir dirname

* ‘ls’ is used to list out all files & directories available in the given directory.

Note: we can pass several options for ‘ls’ commands

* 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 files in a long listing of files.
* ls -t : it will display all files based on last modified data and time. Most recent files will display 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 & directories along with sub directories 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 the options is not important.

$ ls -ltr

$ ls -tlr

$ ls -l -t -r

$ ls -trl

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’ command 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 hello.txt (first 5 lines data)

$ head -n 20 hello.txt (first 20 lines data)

Tail command:

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

$ tail filename

$ tail -n 50 hello.txt (last 50 lines data)

$ tail -n 200 hello.txt (last 200 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

wc command:

* it is used to count no. of lines, no. of words and no. of characters in the file.

$ wc f1.txt

* 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 f1.txt f2.txt f3.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’ command

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

**Rename the file or directory:**

* To rename the files we will use ‘mv’ command

$ mv f1.txt f111.txt

* To rename the directory we will use the same ‘mv’ command.

$ mv dirname dirnewname

Note: we can use ‘mv’ command for renaming and moving files

**Comparing Files:**

* To compare file we can use the below commands

$ cmp f1.txt f2.txt

$ diff f1.txt f2.txt

* cmp command will display only first difference in given 2 files
* diff command will display all the differences in the content

**grep command:**

* grep stands for global regular expression print.
* grep command will process text line by line and prints any line which matches given pattern.

Example:

I want to print all line which 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 numbers

-l : Displays only file names that matches the pattern

-h : Displays matched lines without file names

-R : Displays matched lines with file names

**Working with Text Editors in Linux**

* 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

1) 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 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.

2) 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.

3) escape mode (last line 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 wait for a command.

-> This mode enables you to perform tasks such as saving files, executing commands.

-> There are following way 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 : Opens an existing file in read only mode.

view filename : Opens an existing file in read only mode.

vi f1.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.

0 or | : Positions cursor at beginning of line.

$ : Positions cursor at end of line.

W : Positions cursor to the next word.

B : Positions cursor to previous word.

( : Positions cursor to beginning of current sentence.

) : Positions cursor to beginning of next sentence.

H : Move to top of screen.

nH : 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.

colon along with x : Colon followed by a number would position the cursor on line number represented by x.

**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 modifications to it, then you need to use permissions in Linux.
* It is a must for every programmer/devops/cloud engineer 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 with this part “-rw-r–r–".

The characters mean:

‘r’ = read.

‘w’ = write.

‘x’ = execute.

‘-’ = no permission.

-rw-r--r--

-: It represents file

rw: 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+w section.txt

This command will add the write permission for other users to my text file “section.txt”.

Now if you try to execute ls -l then you will see -rw-r--rw-

“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 with:

chmod u+x section.txt

The permissions will be -rwxr--rw-

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:

chmod u-x section.txt

And the permissions now are: -rw-r--rw-

Also, you can use Symbolic Mode to modify permissions 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:

chmod 777 section.txt

Then the permissions will be: -rwxrwxrwx.

Let’s remove the execute from the group and the write from other by:

chmod 765 section.txt

**SED command:**

* SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion.
* Though most common use of SED command in UNIX is for substitution or for find and replace.
* 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.

Example:

$ cat > myfile.txt

unix is great os. unix is opensource. 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

* By default, the sed command replaces the first occurrence of the pattern in each line and it won’t replace the second, third…occurrence in the line.
* Replacing the nth 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 Delete a particular line say n in this example

$ sed '5d' myfile.txt

-> To Delete a last line : $ sed '$d' myfile.txt

-> To Delete from nth to last line : $ sed '12,$d' myfile.txt

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