```
==============
Day-01 : Linux
==============

1) Pre-Requisites

2) Course Content

3) Who can learn this course

4) What is the benefit you are goning to get from this course

5) Realtime usecases of Linux OS

6) Course Info

===============
pre-requisites
===============

1) Laptop/Desktop => 8 GB RAM
2) AWS Cloud Account (Free Tier)

==============================
Who should learn this course ?
==============================

1) Freshers
2) Working Professional (java, net, python, tester, devops, aws)
3) Non IT people

====================================
Where we will use Linux in Realtime ?
====================================

To setup Tomcat server

To setup jenkins

To setup docker

To setup k8s

To setup sonar

To setup nexus

To setup ELK

To store logs

========================================
What we will learn as part of this course
========================================

Name : Linux + Shell Scripting

Duration : 2 Weeks

Class Time : 6:30 PM to 8:00 PM (IST) ==> Mon-Sat

Course Fee : Zero (Live Classes + Softcopy material)
```

Backup Videos : 499 INR (1 year validity)

==========
What is Operating System
==========

=> It is a software which acts as mediator between users and computers

=> Users will communicate with computers using Operating System (OS)

=> Without OS we can't use any computer

=> OS provides platform/environment to use computers

Ex: Notepad, Calculator, Browsers....etc

=> We have several Operating Systems in market

Ex: Windows, Linux, MAC, Android, IOS.....

==========
Windows OS
==========

=> Developed by Microsoft

=> It is GUI based OS (Graphical User Interface)

=> It is commercial OS (paid)

=> Security features are less (Anti-virus is required)

=> Windows OS is recommended for personal use

Ex: Play Games, Watch Movies, Internet Browsing, Store data, Online Classes...

=> Single User based OS

Note: Windows is not recommended for business usecases (servers, tools, application deployment..)

==========
Linux Operating System
==========

=> Linux is a community based OS

=> Linux is free & Open Source

=> Linux Provides High Security

=> Linux is Multi User based OS

=> Linux is highly recommended for project operations

=> Linux is CLI based OS (Command Line Interface)

```
touch
mv
rm
vi
mkdir
cat
```

Note: In real-time we will use Linux machines to setup our infrastructure.

Ex: jenkins, sonarqube, nexus, docker, k8s, ansible etc.....

==========
Linux OS History
==========

=> Developed by Linus Torvalds

=> Initially Linus Torvalds was using Unix OS and found some challenges in that and informed to that company but they did not accept his suggestions.

=> Linus started doing research and he found "Minux OS" is similar to his ideas.

=> He has taken Minux OS and made few changes to that and released into market as Linux OS.

(Lin)us + Min(ux) = Linux

==========
Linux Distributions / Flavours
==========

=> Linus Tarvalds given Linux os as free and open source

=> Many companies downloaded Linux OS code and modified according to their requirement and released into market with different names.

Those are called as Linux Distributions/Flavours.

Ex: Amazon Linux, Red Hat, Ubuntu, Cent OS, SuSe, Kali, Fedora.....

Note: We have 200+ linux distributions

==========
Summary
==========

1) What is OS & Why we need it ?
2) Windows OS
3) Linux OS
4) Linux OS History
5) Linux Distributions

==========
Linux Machine Setup
==========

1) Login into AWS cloud account

2) Create Linux Virtual Machine using AWS Ec2 service

3) Connect with Linux VM using Git Bash/ MobaXterm / Putty

ðŸ”¥ Thankyou all for being part of *Ashok IT - Family*

âœ… Please go thru below videos and get ready with setup before coming to tomorrow's class.

ðŸ‘‰ Linux Free Material Access: https://youtu.be/GySi7CuOZdk?si=2eIfmSRY5_WUAwHl

ðŸ‘‰ AWS account setup : https://www.youtube.com/watch?v=xi-JDeceLeI

ðŸ‘‰ Linux Machine with Git Bash : https://www.youtube.com/watch?v=JMlQaTXvw5o

ðŸ‘‰ Linux Machine with MobaXterm : https://youtu.be/uI2iDk8iTps?si=ZuZs0lQTxoRpbRMk

ðŸ‘‰ Linux Machine with putty : https://youtu.be/GXc_bxmP0AA?si=HgSydrP89mPxv23s

ðŸ’¥ *Linux Backup Videos (499 INR) : https://rzp.io/l/linux-os*

-
Ashok IT

==========
Linux File System
==========

=> Everything is represented as a file only

=> 3 Types of files

1) Ordinary file / Normal file (starts with -)

2) Directory file (folder) (starts with d)

3) Link File (starts with l)

==========
Linux Commands
==========

whoami : To display logged in username

date : displays current date

cal : display calendar

pwd : display present working directory path

cd : change directory

$ cd <dirname>

$ cd ..

ls : list content

mkdir : make directory (create)

rmdir : delete empty directory

ls -l : Long listing with alphabetical order (ascending)

ls -lr : reverse of alphabetical order (descending)

ls -lt : Display latest files on top

ls -ltr : old files on top

```
ls -la : display hidden files (starts with .)

touch : to create empty files

                $ touch f1.txt f2.txt

rm : to delete files/non-empty directories

                $ rm <file-name>

                $ rm *.txt

                $ rm a*.txt

                $ rm -rf <dir-name>

mv : rename & move from one location to another location

                $ mv  <present-name> <new-name>

                $ mv  <present-location>  <new-location>


cat : create file with data + append data to file + print file data

                $ cat > f1.txt

                $ cat >> f1.txt

                $ cat f1.txt

                $ cat -n f1.txt

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


tac : To print file content from bottom to top

    $ tac f1.txt

rev : To reverse each line of data

    $ rev f1.txt

cp : To copy one file data into another file

    $ cp f1.txt f2.txt

Note: To copy more than one file data into another file we will use cat command

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

wc : word count command

      $ wc f1.txt  (no.of lines, no.of words, no.of chars)

cmp : Compare file

          $ cmp f1.txt f2.txt


head : To display file data from top (default 10 lines)

        $ head f1.txt
```

```
        $ head -n 5 f1.txt   (print first 5 lines only)

        $ head -n 15 f1.txt   (print first 15 lines only)

tail : To display file data from bottom (default 10 lines)

      $ tail f1.txt

      $ tail -n 20 f1.txt  (print last 20 lines of file)

      $ tail -n 100 f1.txt (print last 100 lines of file)

      $ tail -f f1.txt (to get live data)

grep : grep stands for global regular expression print

          $ grep 'aws' f1.txt   (print lines having aws keyword)

      $ grep -i 'AWS' f1.txt (ignore case sensitive)

      $ grep -n 'aws' f1.txt (print lines having aws with line number)

      $ grep -v 'aws' f1.txt (print lines which doesn't have aws keyword)

      $ grep -i -n 'linux' *  (search linux in all files of pwd)


=======================
Text Editors in Linux
=======================

=> vi (visual editor) it is default editor in linux machines

=> Using 'vi' we can create new files and we can modify existing file data

                $ vi f1.txt

=> vi command is having 3 modes

        a) command mode (just to open the file)

        b) insert mode (to edit the file ) ---> press 'i' in keyboard

        c) esc mode (to comeout from insert mode) --> press 'esc' in keyboard


    ## Save changes & close the file =>  :wq

    ## Without saving changes close the file => :q!


Note: vi command will open the file if it is already avilable otherwise it will create new file and
it will open that file.

==================================
file creation commands in linux
==================================

touch : to create empty file

cat : create file with data

cp : copy one file into another file (cp f1.txt f2.txt)
```

```
vi : create and open file for editing (vi f3.txt)

===================================
Reading file data commands in linux
===================================

cat : print file data from top to bottom

tac  : print file data from bottom to top

rev : print each line in reverse order

head : print first 10 lines of file data

tail : print last 10 lines of file data

vi : open the file


=============
SED command
=============

=> SED stands for stream editor

=> SED is used to process the data (substitute,delete,insert)

=> Using SED command we can perform operations on the file without opening the file.

=> SED is very powerful command in linux

=> Create a file with below data to practice sed command

java is a programming language
java is free
java is open source
java is easy
i want to learn java
java is having good demand in market
java is oop based. java is easy.


# substitute first occurance of 'java' with 'python' in every line
$ sed 's/java/python/' data.txt

# Replace second occurance of 'java' with 'python' in every line
$ sed 's/java/python/2' data.txt

# Replace third occurance of 'java' with 'python' in every line
$ sed 's/java/python/3' data.txt

# Replace all occurances of 'java' with 'python' in every line
$ sed 's/java/python/g' data.txt

# Substitute and save changes in original file
$ sed -i 's/java/python/g' data.txt

# Delete 4th line of file
$ sed -i '4d' f1.txt

# Delete last line of file
$ sed -i '$d' f1.txt

# Delete data from 3rd line to last line
$ sed -i '3,$d' data.txt
```

```
# Delete data from 5th line to 10th line
$ sed -i '5,10d' data.txt

# print all lines which contains 'python' keyword
$ sed '/python/p' data.txt

# delete all lines which contains 'python' keyword
$ sed '/python/d' data.txt

# print data from 3rd line to 6th line
$ sed -n '3,6p' data.txt

# insert data at 4th line
$ sed '4i\i am from ashokit' data.txt

# Add given text at end of the file
$ sed '$a\i love linux' data.txt


====================
What is AWK Command
====================

=> The awk command is a versatile text processing tool available Linux.

=> It allows you to manipulate and extract data from structured text files, usually in a columnar
format.

=> awk takes input, processes it line by line, and performs actions based on specified patterns and
rules.

Sytax : awk 'pattern { action }' file

$ cat > employees.txt

Ashok manager account 45000
John clerk account 25000
Smith manager sales 50000
Charles manager account 47000
Ganesh peon sales 15000
Mahesh clerk sales 23000
Ram peon sales 13000
Cathy director purchase 80000

# Print entire file data
$ awk '{print}' employees.txt

# print only manager role data
$ awk '/manager/ {print}' employees.txt

# print first & fourth columns data
$ awk '{print $1,$4}' employees.txt

# print file data with line numbers
$ awk '{print NR,$0}' employees.txt

# print first column data along with line number by appending @
$ awk '{print NR "@" $1 }' employees.txt


==========================
Users & Groups Management
==========================
```

```
=> Linux is a multi user based OS

=> Within one linux machine we can create multiple user accounts

=> Multiple users can acces single linux machine at a time and can perform multi tasking

Note: "ec2-user" is default user in amazon linux vm.

# Create new user
$ sudo useradd <username>

# set password for user
$ sudo passwd <username>

# display all users created
$ cat /etc/passwd

# swith user
$ sudo su <username>

# Go to logged in user home directory
$ cd ~

# Delete User
$ sudo userdel <username>

# Delete user along with user home directory
$ sudo userdel <username> --remove

# how to change username
$ sudo usermod -l <new-name> <old-name>


===========================
Working with User Groups
===========================

=> When we create user in linux, for every user one user group also will be created with the given
username.

# Display all groups in linux
$ cat /etc/group

# create group
$ sudo groupadd <group-name>

# Add user to group
$ sudo usermod -aG <group-name> <username>

# Remove user from group
$ sudo gpasswd -d <username> <group-name>

# Display users belongs to a group
$ sudo lid -g <group-name>

# Display one user belongs to which grops
$ id <username>

# Delete Group
$ sudo groupdel <group-name>

# Changing groupname
$ sudo groupmod -n <new-name> <old-name>

================================
```

```
What is sudoers file in Linux
================================

=> It is very important configuration file in linux machine.

=> Using this file we can control which user can run command as a superuser.

# print sudoers file content
$ sudo cat /etc/sudoers

Note: We should be very careful while working with sudoers file. If we do any mistakes in sudoers
file then system will be crashed.

########## Giving sudo previliges for user #######

# Open sudoers file
$ sudo visudo

# Add below line to give sudo permission for particular user
username ALL=(ALL:ALL) ALL

Ex : ashok ALL=(ALL:ALL) ALL

=> After making changes, to close sudoers file => ( CTRL + X + Y + Enter)

=========================================================
How to enable password based authentication for user ?
=========================================================

=> To enable password based authentication we need to deal with sshd_config file

=> in sshd_config file, by default PassswordBasedAuthentication is no.

=> To enable password based authentication we need to set the value as yes.

# Display sshd_configurration file data
$ sudo cat /etc/ssh/sshd_config

# Open sshd_config file
$ sudo vi /etc/ssh/sshd_config

# restart sshd service
# sudo systemctl restart sshd

=============================================
Connect with Linux VM using username and pwd
=============================================

Step-1) Connect to Linux VM as ec2-user using pem file

Step-2) Create new user 'ram'

        $ sudo useradd ram

Step-3) Update password for 'ram'

        $ sudo passwd ram

Step-4) Configure 'ram' in sudoers file

        $ sudo visudo

Step-5) Enabled PwdBasedAuthentication in 'sshd_config' file

        $ sudo vi /etc/ssh/sshd_config
```

```
Step-6) Restart sshd service

        $ sudo systemctl restart sshd

Step-7) Use putty & Connect to linux vm as 'ram' user using username & pwd


==================
File Permissions
==================

=> In linux, file permissions are divided into 3 types.

    r => read
    w => write
    x => execute

=> Every file will have 3 sections in permissions

      => user (owner)  (u)
      => group  (g)
      => other users (o)

      rwxrwxrwx

      => first 3 chars will represent user(owner) permissions

      => middle 3 chars will represent group permissions

      => last 3 chars will represent other users permissions


    1)  rw-rwxrw-  f1.txt

                => rw- : user having read + write
                => rwx : group having read+write+execute
                => rw- : others having read + write

    2)  r-xr--r--  f2.txt

                => r-x : user having read + execute
                => r-- : group having only read
                => r-- : others having only read

    3)  -w-r-xrwx  f3.txt

                => -w- : user having only write
                => r-x : group having read + execute
                => rwx : others having read + write + execute


=> To change file permissions in linux, we will use 'chmod' command


# Giving execute permission for user
$ chmod u+x f1.txt

# giving write permission for group
$ chmod g+w f1.txt

# Remove execute permission for others
$ chmod o-x f1.txt

# Remove all permissions for other users
```

```
$ chmod o-rwx f1.txt

==================================
File Permissions in Numeric Format
==================================

0 => No permissions

1 => Execute

2 => Write

3 => (2+1) => Write + Execute

4 => Read

5 => (4+1) => Read + Execute

6 => (4+2) => Read + Write

7 => (4 + 2 + 1) => Read + Write + Execute


$ chmod 777 f1.txt

$ chmod 111 f2.txt

$ chmod 222 f2.txt

$ chmod 456 f2.txt


Q-1) What is default permissions for file in linux ?

                644

Q-2) what is default permissions for directory in linux ?

                755

==============
chown command
==============

=> It is used to change file/directory ownership

# changing owner
$ sudo chown new-owner file/directory

# changing owner-group
$ sudo chown :newGroup file/directory

# changing owner & owner-group
$ sudo chown owner:group file/directory


==========================================
Q) What is the diff between chmod & chown ?
==========================================

chmod => To change file/directory permissions

chown => To change owner/group
```

```
==================================
Working with zip files in linux
==================================

=> Zip is used for files archieve (compress)

####  create zip file syntax : zip <zip-name>  <content>

# create zip with all txt files
$ zip ashokit *.txt

# print content of zip file
$ zip -sf ashokit.zip

# Add new file to existing zip file
$ zip -r ashokit.zip f4.txt

# Delete file from existing zip file
$ zip -d ashokit.zip f4.txt

# create zip file with password
$ zip -e ashokit *.txt

# Extract zip file content
$ unzip ashokit.zip

======================
Networking Commands
======================

ping : To check connectivity

    $ ping 192.168.8.0

    $ ping www.google.com

wget : It is used to download files from internet

    $ wget <url>

    $ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.89/bin/apache-tomcat-9.0.89.zip

curl : It is used to send http request to server & get the response

    $ curl <url>

    $ curl https://type.fit/api/quotes

ifconfig: To get the IP address of the machine

      $ ifconfig


==================================================

free : to display memroy level details

top : to display running processes

htop : to display running processes in table format

# install htop command
$ sudo yum install htop

=========================
```

```
Working with Link Files
==========================

=> In linux we can create link files ( similar to shorcut files in windows )

=> We can create link files in 2 ways

  1) hard link

  2) soft link

-------------------------------
Syntax To create Hard Link
-------------------------------

$ ln <original-file> <link-file>


$ touch m1.txt

$ ln m1.txt f1.txt

Note: If we write some data to m1.txt that data will reflect in f1.txt file also

Note: If we delete m1.txt file there is no effect on f1.txt

Note: If we delete original file there is no effect on hard link file

-----------------------------
Syntax To create Soft Link
-----------------------------

$ ln -s <original-file> <soft-link-file>

Note: When we remove original file then soft link file will become dangling file. We can't access
that file.

============================================================
Q) what is the diff between find and locate commands ?
============================================================

=> find and locate commands are used for file location search

=> locate command will search for files in locate db.

=> find command will search for files in entire linux file system based on given path


# search for the files which are having name as f1.txt
$ sudo find /home -name f1.txt

# search for all empty files inside /home
$ sudo find /home -type f -empty

# search for all empty directories inside /home
$ sudo find /home -type d -empty

# print 30 days old files in linux
$ sudo find . -mtime 30 -print

# delete 30 days old files in linux
$ sudo find /home -mtime 30 -delete

=======================
package managers linux
```

```
=========================

=> Package Managers are used to install / update / remove / manage software packages in linux
machines.

  Ex : java, python, webservers, mysql ....

=> Package managers are specific to linux distribution.

     Amazon linux / Red Hat / Cent OS => YUM

     Ubuntu / Debian => APT

# install git in amazon linux vm
$ sudo yum install git

# install git in ubuntu linux vm
$ sudo apt install git

# install maven in amazon linux
$ sudo yum install maven

Note: Maven developed using java so when we install maven s/w then java s/w will be installed by
default.

# install java in amazon linux
$ sudo yum install java

==============================
Install MySQL in Amazon linux
==============================

# execute below commands to install client & server

sudo wget https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm
sudo dnf install mysql80-community-release-el9-1.noarch.rpm -y
sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023
sudo dnf install mysql-community-client -y
sudo dnf install mysql-community-server -y

# edit my.cnf file
$ sudo vi /etc/my.cnf

# add below line
skip-grant-tables

# After modifying the file save and restart mysql again by running
$ sudo systemctl restart mysqld

# check connectivity using below command
$ mysql

====================================
Installing Web Server in Linux VM
====================================

=> Webserver is a software which is used to run websites

=> We can use 'httpd' as a webserver in amazon linux machines

# install httpd webserver
$ sudo yum install httpd

# start httpd server
$ sudo service httpd start
```

```
Note: Httpd webserver runs on http protocol with 80 port number.

Note: To access our webserver, we need to enable http protocol in ec2 vm security group inbound
rules.

=> We can access our webserver using ec2-vm public ip.

# Navigate to website content directory
$ cd /var/www/html

# create index.html file with website content
$ sudo vi index.html

Note: To stop our webserver we can use below command
$ sudo service httpd stop


============================
What is systemctl in linux ?
============================

=> systemctl is a command-line utility in Linux systems. It is used to manage system services.

=> some common tasks we can perform using systemctl

=> Starting service
=> stopping service
=> restarting  service
=> reloading service
=> enabling / disabling services

=================================================
Here are some examples of how you can use systemctl
=================================================

# start a service
$ systemctl start service-name

Ex: systemctl start httpd

# stop the service
$ systemctl stop service-name

# re-start the service
$ systemctl restart service-name

# re-load service
$ systemctl reload service-name

# enable service
$ systemctl enable service-name

# disable service
$ systemctl disable service-name

# check the status of a service
$ systemctl status service-name


==================================
How to change hostname in linux vm ?
==================================

# set hostname
```

```
$ sudo hostname <new-name>

# re-start session
$ exit
```

Note: Connect with machine again using ssh command.

==========
Q) How to copy files from one linux machine to another linux machine ?
==========

==========
Linux Architecture
==========

1) Shell
2) Kernel
3) Hardware components : drivers, printers, disc....

=> Kernel is the core component is linux os. Kernel is responsible to communicate with Hardware components.

=> Kernel will act as mediator between shell and linux hardware components.

=> Shell is a mediator between users and kernel. Our commands will be processed by shell only.

==========
What is Scripting ?
==========

=> Scripting means set of commands we are keeping in a file for execution.

=> Scripting is used to automate our daily routine work.

=> For example, i want to execute below commands on daily basis

```
whoami
pwd
date
cal
ls
```

Note: instead of executing these commands one after other manually we can keep them inside a file and we can execute that file which is called as Scripting.

=> The process of executing script file using shell is called as Shell Scripting.

=> Shell scripting is used to automate our daily routine work in the project.

Note: Shell script files we will create using .sh extension.

==========
what is sha-bang in linux ?
==========

=> sha-bang is used to specify which shell we should use to process our script file.

syntax: #! /bin/bash

Note: Writing sha-bang is not mandatory but recommended.

========== 01 - Shell Script ==========

```
#! /bin/bash
```

```
echo "Enter Your Name"

read uname

echo "Hey $uname, welcome to ashokit..."

==================02 - Shell Script ===============

#! /bin/bash

echo  "Enter your firstname"

read fname

echo "Enter your lastname"

read lname

echo "Your Fullname : $fname $lname"

===========
Variables
===========

=> Variables are used to store the values

=> Variables will represent data in key-value format

        a = 10
        b = 20
        name = ashokit
        age = 30

Note: We don't have data types in shell scripting.

=> We have 2 types of variables

        1) System Variables / Environment Variable

        2) User Defined Variables

=> The variables which are already defined and using by our system are called as System variables.

        $ echo $SHELL

        $ echo $USER

        $ echo $PATH

Note: We can access all the environmental variables using below command

        $ env

=> The variables which we are creating for our requirement are called as 'User Defined Variables'.

            name = ashok
            id = 101
            age = 25
            gender = male

Note : To access value of variable we will use below syntax

            $ echo $VARIABLE_NAME
```

```
# create variable using terminal
$ export course=devops

# get variable value
$ echo $course

# unset variable
$ unset variable_name
```

Note: If we use export command in terminal for setting variables then those variables will be removed once we close our terminal. These are called temporary variables.

```
==================================
How to set variables permanently ?
==================================
```

=> We will use .bashrc file to set variables permanently for the user.

=> In user home directory, .bashrc file will be available.

```
        $ cat .bashrc

# open .bashrc file
$ vi .bashrc

# add variables at end of the file
course=devops
trainer=ashok

# apply .bashrc changes
$ source .bashrc

# Access variables
$ echo $course
$ echo $trainer
```

Note: In linux machine, every user will contain their own .bashrc file.

```
================================================
How to set variables for all users in linux ?
================================================

$ cat /etc/profile
```

Note: If we add variables in /etc/profile then those variables applicable for all users in linux vm.

```
================
Variables Rules
================
```

=> Variable names shouldn't start with digits

=> Variable names shouldn't contains special symbols

    Ex: - , @, #

Note: It is recommended to use UPPERCASE characters for variable names

```
              name ===> NAME

===========
Operators
===========
```

```
=> Operators are used to perform some operation on variables.

Ex:

a=10
b=20

a+b
a>b
a==b

========================
Arithematic Operations
========================

Addition : $((a + b))

Multiplication : $((a * b))

Substraction : $((a - b))

Division : $((a / b))

Modulas : $((a % b))


==================03 - Shell Script ===============

#! /bin/bash

echo "Enter first number"

read FNUM

echo "Enter second number"

read SNUM

echo "Result : $((FNUM+SNUM))"

========================
Comparision Operators
========================

Equals ( == )

Not equal ( != )

Greater Than ( > )

Less Than ( < )

========================
Conditional Statements
========================

=> Conditional statements are used to execute commands based on condition.

Syntax :

if [ condition-1 ]; then
   // stmt-1

elif [ condition-2 ]; then
   // stmt-2
```

```
else
  // stmt-3
fi
```

==========04 - Shell Script ==========

Requirement : Read two numbers and check weather those numbers are equal or not.

```
#! /bin/bash

echo "Enter First Num"
read N1

echo "Enter Second Num"
read N2

if [ $N1 -eq $N2 ]; then
  echo "Both are equal"
else
  echo "Both are not equal"
fi
```

==========05 - Shell Script ==========

Requirement : Read user age and determine weather user is eligible for vote or not.

```
#! /bin/bash

echo "Enter Your Age"
read AGE

if [ $AGE -gt 18 ]; then
 echo "Eligible For Vote"
else
 echo "Not Eligible for vote"
fi
```

==========06 - Shell Script ==========

Requirement : Read a number and check type of number (Positive or Negative or Zero)

```
#! /bin/bash

echo "Enter Number"
read N1

if [ $N1 -gt 0 ]; then
 echo "Positive Num"
elif [ $N1 -lt 0 ]; then
 echo "Negative Num"
else
 echo "It is zero"
fi
```

==========
Looping Statements
==========

=> Loops are used to execute statements multiple times.

=> We can use 2 types of Loops

1) Range Based Loop (ex: for)

```
      2) Conditional Based Loops (ex: while)

=================
For loop Syntax
=================

for(( intialization ; condition ; modification ))

do
   // stmts

done


================================================
For loop example - Print Numbers from 1 to 10
================================================

#! /bin/bash

for((i=1; i<=10; i++))
do
echo "$i"
done

================================================
For loop example - Print Numbers from 10 to 1
================================================

#! /bin/bash

for((i=10; i>=1; i--))
do
echo "$i"
done

==============
while loop
==============

=> While loop is used to execute statements until condition is true

==============================
Print Numbers from 1 to 10
==============================

#! /bin/bash

N=1

while [ $N -le 10 ]
do
echo "$N"

let N++

done

==============================
Print Numbers from 10 to 1
==============================

#! /bin/bash
```

```
N=10

while [ $N -ge 1 ]
do
echo "$N"

let N--

done
```

```
=============
Functions
=============

=> Functions are used to perform some action / task

=> Using functions we can divide big task into multiple small tasks.

=> Functions are used to divide our work logically

=> Functions are re-usable.

---------
syntax
---------

# creating function
function funcName ( ) {

    // function body
}

# call function
funcName

====================== Script with Function ==============

#! /bin/bash

function welcome(){
    echo "Welcome to ashok it"
    echo "welcome to linux"
    echo "welcome to devops"
}

welcome

================================================================================
Q) Write a function which will read filename from user and print content of that file.
================================================================================

#! /bin/bash

# creating function
function readFile(){
  echo "Enter filename"
  read FNAME

  echo "*********printing file data - start ************"
  cat $FNAME
  echo "*********printing file data - end *************"
}

#calling above function
readFile
```

```
================================================================================
Q) Write a function which will read filename from user and print content of that file.
================================================================================

#! /bin/bash

# creating function
function readFile(){
  echo "Enter filename"
  read FNAME

  if [ -f "$FNAME" ];then

  echo "*********printing file data - start ************"
  cat $FNAME
  echo "*********printing file data - end *************"

  else
   echo "********* - File Not Found - ************"

  fi
}

#calling above function
readFile


=========================
Command Line Arguments
=========================

script file : task.sh

execution : sh task.sh

=> cmd args are used to supply values to script file at the time of execution.

    $ sh task.sh 10 20 ashok it 20 30

=> cmd args are we can access in script file like below...

$# => Total no.of args passed

$0 => Get script file name

$1 => Read First cmd arg

$2 => Read second cmd arg

$* => Read all cmd args

============================

#! /bin/bash

echo "Total Args : $#"

echo "Script file name : $0"

echo "First cmd Arg : $1"

echo "Second cmd Arg : $2"

echo "==========="
```

```
echo "All cmd args : $*"

============================

================================================================
Q) Write shell script to perform sum of 2 numbers using cmd args
================================================================

#! /bin/bash

result=$(($1+$2))

echo "sum is : $result"


=========================================
Q) Write shell script to print system info
=========================================

# print host name
echo "Host Name : $(hostname)"

# print current date & time
echo "Date & Time : $(date)"

# print system uptime
echo "System Uptime : $(uptime)"

# print disc usage
echo "Disc usage"
df -h

# print memory usage
echo "Memoy usage"
free -h

================================================================================================
Q) Write shell script for log analysis. Identify how many ERRORS avalable in log file
================================================================================================

#! /bin/bash

logfile="/var/log/dnf.log"

#count error msg occurance
errCount=$(grep -c "ERROR" "$logfile")

echo "Number of errors : $errCount"


=========================================================================
Q) Write shell script to create ec2-user home directory files backup
=========================================================================

#! /bin/bash

backup_dir="/home/ubuntu"
source_dir="/home/ubuntu"

tar -czvf "$backup_dir/backup_$(date +%Y%m%d).tar.gz" "$source_dir"


==========================
Shell Scripts for practice
```

```
==========================

1) Write shell script to check given number is even or odd

2) Write shell script to check given number is prime number or not

3) Write shell script to check given string is palindrome or not

4) Write shell script to print table of given number like below

          2 * 1 = 2
          2 * 2 = 4
          ...
          2 * 10 = 20


====================
What is Scheduling ?
====================

=> Scheduling means configuring the tasks to be executed automatically.

    Ex: Setting alarm @6:00 AM in phone

=> Similar to alarm trigger, i want to schedule my script file execution.

=> In linux, we will use CRON to schedule jobs/scripts execution.

=> CRON is an utility in linux to schedule jobs execution.

=> In real-time we will use several jobs on daily/weekly/monthly/yearly basis to automate our work.

      - Delete temp files
      - Take backup of files
      - System health checks

=========
usecase
=========

=> Execute shell script for every 5 minutes.

Note: Instead of human executing script for every 5 minutes, we can automate script execution using
CRON job.

================
CRON Job syntax
================

Syntax :   * * * * * <script-file>

=> First * will represent minutes ( 0 - 59 )

=> Second * will represent hour ( 0 - 23 )

=> Third * will represent day of month ( 1 - 31 )

=> Fourth * will represent month of year ( 1 - 12 )

=> Fifth * will represent day of week (0 - 6 / sun-mon)

======================
Sample CRON Schedules
======================
```

```
Run for every minute => * * * * * <script-file-name>

Run for every 15 mins => */15 * * * * <script-file-name>

Run every day @5:00 AM => 0 5 * * * <script-file-name>

Run every day @8:00 PM => 0 20 * * * <script-file-name>

Run every month first day @9:00 AM => 0 9 1 * * <script-file-name>

# 20th May - 9 am
0 9 20 5 * <file-name>

# Dec 31st @11:55 PM
55 23 31 12 * <script-file-name>

### Cron Expression Generator : https://crontab.guru/

==================
What is CROND ?
==================

=> In linux machines, CROND is a deamon process (background process).

=> Every minute, CROND will be checking for CRON Jobs Schedule for the execution.

========================
what is crontab file ?
========================

=> Crontab file is used to configure cronjobs for execution.

# open crontab file
$ crontab -e

# Display cronjob schedules
$ crontab -l

# remove crontab file
$ crontab -r

====================
CRONJOB Practicals
====================

1) Launch Linux machine with UBUNTU AMI

2) Connect with Ubuntu VM using MobaXterm / Putty / GitBash

3) Create shell script file and keep below content

    $ vi task.sh

touch /home/ubuntu/f1.txt
touch /home/ubuntu/f2.txt

4) Provide execute permission for script file

    $ chmod +x task.sh

5) Open crontab file and configure job schedule

    $ crontab -e

Note: Add below job schedule info
```

```
   */1 * * * * /bin/bash /home/ubuntu/task.sh

  6) Save and close the crontab file (ctrl + x + y + enter)

  7) After 1 minute check files got created or not.

         $ ls -l

  ============================================================

  ============================
  Linux Commands we practiced
  ============================

  1) whoami
  2) date
  3) cal
  4) pwd
  5) cd
  6) ls
  7) touch
  8) rm
  9) mv
  10) cat
  11) tac
  12) rev
  13) cp
  14) diff
  15) cmp
  16) wc
  17) history
  18) head
  19) tail
  20) grep
  21) vi
  22) sed
  23) awk

  24) useradd
  25) passwd
  26) id
  27) userdel
  28) usermod
  29) groupadd
  30) groupdel

  31) chmod
  32) chown

  33) ping
  34) wget
  35) curl
  36) ifconfig

  37) zip
  38) unzip

  39) free
  40) top
  41) htop

  42) ln

  43) yum
```

44) systemctl

==========
Shell Scripting - Summary
==========

1) What is Shell
2) What is Kernel
3) What is Scripting
4) Shell Scripting
5) Why shell scripting
6) What is Sha-bang
7) Variables
8) System Variables vs user-defined variables
9) what is .bashrc ?
10) Commandline arguments
11) Operators
12) Conditional Stmts (if-elif-else-fi)
13) Looping stmts (for, while)
14) Functions
15) Scripts for practice
16) Printing system info using script
17) Taking files backup using script
18) CRON Jobs

==========
Linux FAQ's
==========

1) What is Kernel
2) What is Shell
3) How do you find files in linux
4) What is the purpose of grep command
5) explain chmod command
6) how to create link files
7) how to check system resource usage in linux
8) Explain the role of /etc/passwd file
9) How to check network connectivity in linux
10) how to deal with services in linux
11) What is CRON in linux
12) How to manage packages in linux
13) How to create backup in linux
14) Explain about ssh command
15) How to monitor linux system performance
16) How to kill process in linux
17) What is sudo in linux
18) How to check IP addr of linux machine
19) How to manage users and groups in linux
20) How to check linux system information
21) How to check running process in linux vm
22) How to change file permissions in linux
23) how to change file ownership
24) How to search content in file
25) how to check system execution time
26) How to check listening ports in linux
27) What is linux distribution
28) What is vi in linux
29) How to delete 2 months old files in linux
30) Which linux distribution used in your project ? (RHEL)
31) grep vs awk vs sed