LINUX

Linux is the open source operating system.It is based on unix system.first unix has came which uses C programing language for developing unix system.Open source means data and source are available to the user.So advantage is developer can use or enhance the code to wok on and re- distribution.

Security falws are are detected and cover.All viruses are on data and not on os in linux.So that if viruses came it eill dleete the files and not affect the os so secure.

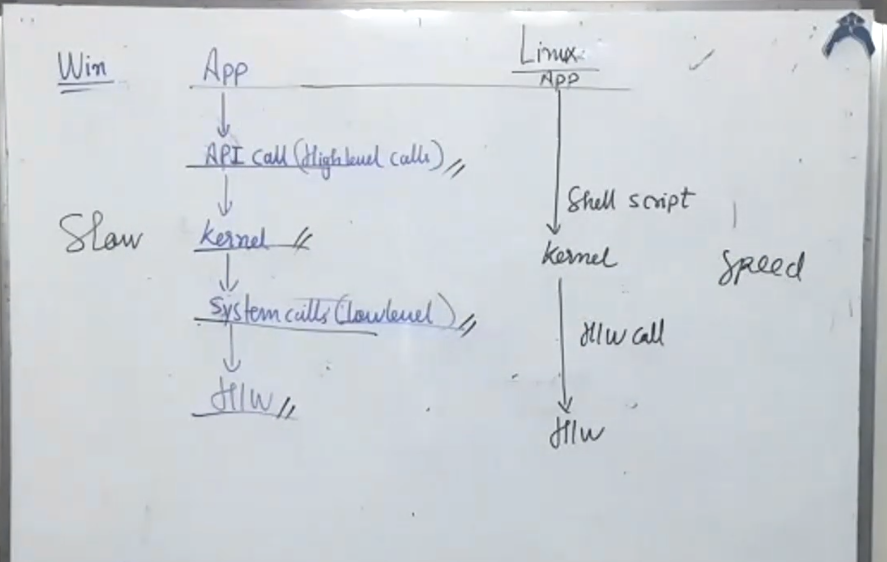
It has built in multi user and multi tasking.Kernel is designed such a way that it supports both.

High preformance.Hardware performance wise it is 10 timesbetter than windows.

Unix internally have command line interface(CLI) initially.e.g. cmd.Then linux are made which has a 70% and 30%.30% is GUI and 70% CLI.Then it has made ubundu which is today has 90-95% of GUI but 100% CLI is also there.

Linux have a ability that means the android is made on linux and MAC also is on linux.

**In terms of windows and linux:-**



In windows we go multiple phases to communicate with the hardware but in linux we write shell script and then pass to kernel and kernel will take the hardware call.Soworking wise windows are slower and linux is faster as it supports shell scripting.

**Terminal:-**

Terminal are the medium where we write the commands.

$ - Basic commands

Cd 🡪 Change Directory

Pwd 🡪 It will give the current working directory path.

Ls 🡪 list directory.

--help 🡪 It will give the documentation of whole commands.

mkdir 🡪 Make directory.

clear🡪 Clear screen.

cp 🡪Copy file.

mv 🡪 Move file

rm 🡪 Remove file.

Syntax:- <command> <filename> <destination>

grep 🡪 If we want the file content then we have grep command.i.e. to read conent in file

touch <file.extension> 🡪 It will create the file but not allow u to enter.

nano <file.extension> 🡪It will create the file and open the file for writing.

man 🡪 man command is used for help file similar to --help command.

What is the type of interface linux follow?

🡪It follows CLI i.e. command line interface.GUI is introduced later.

**SHELL SCRIPTING**

we are using Baurn shell which is buildin shell in linux.

To create file we do

|->nano \_\_.sh

>#!/bin/sh 🡪This says that we are writing the file in baurn shell.

>line of code

>line of code

|->ctrl+s 🡪For saving the file.

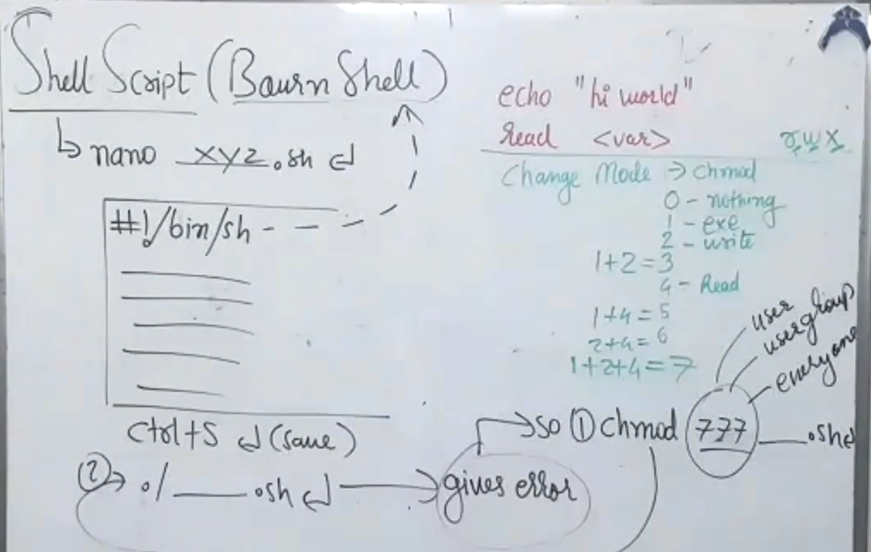
|->ctrl+x 🡪For exiting from file.

|->./\_\_.sh 🡪It will give error. We want premission to access the file for that we use chmod.

echo “hi world” 🡪 echo allows you to print anything on the screen.

read <variable> 🡪 which allows you to reaqd data from user prompt and store it.

**CHMOD COMMAND:-**



It is used to change mode of the file i.e. used to grant permissions to the file.

When we talk about change mode there are 3 rights to be given to file they are read,write,execute.

0 🡪nothing no access.

1 🡪execute

2 🡪write

3 🡪1+2 🡪 execute+write

4 🡪Read

5 🡪1+4 🡪 execute+read

6 🡪2+4 🡪 write+read

7 🡪4+2+1 🡪 write+read+execute.

Chmod 777 \_\_\_.sh

Why 3 times 7?

One is for user rights,second for user groups rights and third for everyone.

**Types of Variables:-**

Envorinmental variable 🡪env 🡪Whenever we load the filethe envoirnmental variables are set.

Local variable 🡪var 🡪to access variable we use 🡪$var.

|-> echo “\_\_\_\_ $<var>” 🡪In this $<var> is taken as content.

|-> echo ‘\_\_\_\_ $<var>’ 🡪 It will be treated as the general message itself i.e print msg only at it is.

For eg:-

|-> nano demovar.sh

>#!/bin/sh

>no=6

>echo “value set is $no”

>echo ‘value set is $no’

|->chmod 777 demovar.sh

|->./demovar.sh

Global variable 🡪global 🡪 this variables used when we want variable globally that is to share vriable within multiple files then it is used.

For eg:-

|-> env

Special variables in global which is given by CLI+DATA

$0 🡪 this gives the filename

$1 🡪 number of arguments passed

$2 🡪 number of arguments passed.

$# 🡪 gives number of arguments.

$@ 🡪 list of arguments which are given

For eg:- use demovar file 1st all lines make comments by using #then

>echo “File name: $0”

>echo “Argument 1: $1”

>echo “Argument 2: $2”

>echo “Total arguments are : $#”

>echo “Arguments are : $@

|->./demovar.sh mayur chaudhari sidya sagar pritam monu.

Note:-Argument number starts from 1 to n. indexing is from 1 to n.

When we write |-> readonly <var> it will kept the variable constant that is it will not change.

When we write |-> unset <var> this will Clear the content.

|-> nano adder.sh

>#!/bin/sh

>total = $1 + $2

>echo “Total is $total”

|->chmod 777 adder.sh

|->./adder.sh 10 5

|->nano adder.sh

>echo “data available is $1 and $2”

>total = $1 + $2

>echo “Total is $total”

|->/.adder.sh 10 5

|->nano adder.sh

>echo “data available is $1 and $2”

>total = `expr $1 + $2`

>echo “Total is $total”

|->./adder.sh

**Operators:-**

Arithmetic operators:-+,-,\*,/,%

Relational operators:-

< -lt

<= -le

> -gt

>= -ge

== -eq

!= -ne

Logical operators:-

and -a

or -o

Note:- Any conditional relational operators are to be in square brackets [ ]

Conditional statements:-

**IF statemnt:-**

If [ ]

then

code1

code2

code3

elif [ ]

then

code1

code2

else

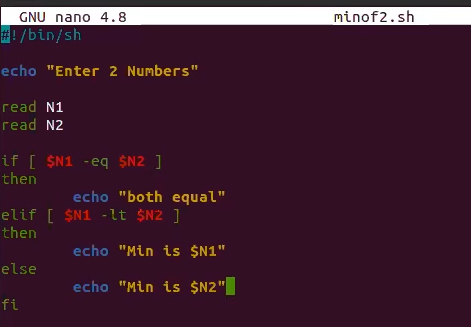
code of else

fi

Note:- When we end any any conditional staements then we write oposite of statement for eg for if we write fi in end to end the statement.

For eg:-Get min of two numbers program

|->nano min.sh



|->chmod 777 min.sh

|->./min.sh 33 4

**Case Statement:- (Switch case)**

Case $var

In

Value1)\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_

;;

Value2)\_\_\_\_\_\_\_

\_\_\_\_\_\_\_

;;

Value3)\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_

;;

\*)\_\_\_\_\_\_

\_\_\_\_\_\_\_

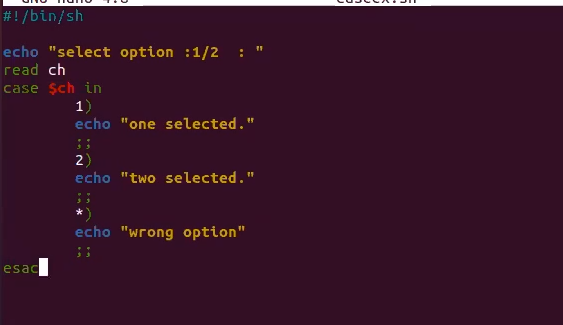
;;

esac

Note :- \* is used for default value.

For eg:-

|->nano case.sh



|->chmod 777 case.sh

|->./case.sh

**Loops in linux:-**

For loop

Syntax:-

for <var> in list

do

code $<var>

done

for eg:-

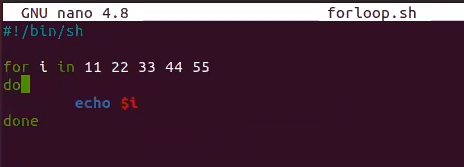
for i in 1 2 3 4 5 6

do

echo “value is $i”

done

|->nano forloop.sh



|->chmod 777 forloop.sh

|->./forloop.sh

|->nano forloop.sh



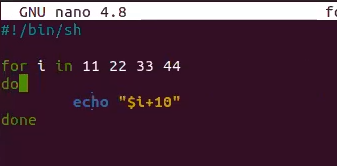
|->./forloop.sh

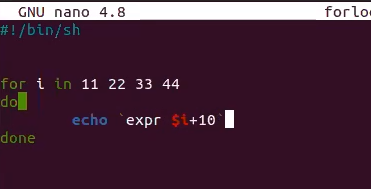
|->nano forloop.sh



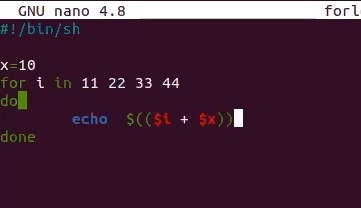
|->./forloop.sh











While loop:-

Syntax:

While [ ]

do

code

done

for eg:-1)

while [$i -lt 5]

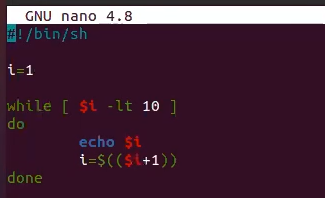
do

echo $i

done

2)

|->nano whileloop.sh



|->chmod 777 whileloop.sh

|->./whileloop.sh

Functions:-

Syntax:

function <name>

{

}

Note:-We donot pass any parameters in linux functions we write within the functions.

For eg:-

function adderf()

{

Total = ‘expr $1 +$2’

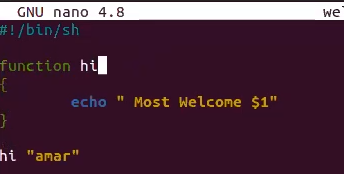
echo “total is $total”

}

adderf 10 20

for eg:-

|->nano hello.sh



|->chmod 777 hello.sh

|->./hello.sh

|->nano func1.sh



|->chmod 777 func1.sh

|->./func1.sh