Bash scripts can be used for various purposes,

such as executing a shell command, running multiple commands together, customizing administrative tasks, performing task automation etc.

So knowledge of bash programming basics is important for every Linux user.

cat /etc/shells

which bash

------------------------------------------------------------------------------------------------------------------------------

#! /bin/bash

echo "raham" > file.txt

------------------------------------------------------------------------------------------------------------------------------

#! /bin/bash

count=10

if [$count -eq 10]

then

echo "true"

fi

#!/bin/bash

n=10

if [ $n -lt 10 ];

then

echo "It is a one digit number"

else

echo "It is a two digit number"

fi

#!/bin/bash

echo "Printing text with newline"

echo -e "\nRemoving \t backslash \t characters\n"

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

# Add two numeric value

((sum=25+35))

#Print the result

echo $sum

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

: '

The following script calculates

the square value of the number, 5.

'

Or

<< ///

The following script calculates

the square value of the number, 5

///

((area=10\*5))

echo $area

------------------------------------------------------------------------------------------------------------------------------=

#!/bin/bash

valid=true

count=1

while [ $valid ]

do

echo $count

if [ $count -eq 5 ];

then

break

fi

((count++))

done

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Enter Your Name"

read name

echo "Welcome $name"

------------------------------------------------------------------------------------------------------------------------------

!/bin/bash

echo "Enter username"

read username

echo "Enter password"

read password

if [[ ( $username == "raham" && $password == "1234" ) ]]; then

echo "valid user"

else

echo "invalid user"

fi

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Enter your lucky number"

read n

if [ $n -eq 101 ];

then

echo "You got 1st prize"

elif [ $n -eq 510 ];

then

echo "You got 2nd prize"

elif [ $n -eq 999 ];

then

echo "You got 3rd prize"

else

echo "Sorry, try for the next time"

fi

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Enter your lucky number"

read n

if [ $n -eq 101 ];

then

echo "You got 1st prize"

elif [ $n -eq 510 ];

then

echo "You got 2nd prize"

elif [ $n -eq 999 ];

then

echo "You got 3rd prize"

else

echo "Sorry, try for the next time"

fi

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Total arguments : $#"

echo "1st Argument = $1"

echo "2nd argument = $2"

./filename Redhat ubuntu fedora centos

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

string1="Linux"

string2="unix"

echo "$string1$string2"

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Enter directory name"

read newdir

mkdir $newdir

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Enter filename"

read newfile

touch $newfile

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Enter directory name"

read ndir

if [ -d "$ndir" ]

then

echo "Directory exist"

else

`mkdir $ndir`

echo "Directory created"

fi

------------------------------------------------------------------------------------------------------------------------------

#!/bin/bash

echo "Enter filename to remove"

read fn

rm -i $fn

CRONTAB EXAMPLES:

\* \* \* \* \* lscpu >> /tmp/cpuinfo

\* \* \* \* \* lsmem >> /tmp/abc.txt

Script to delete files with are older than 60 days:

#! /bin/bash

find /opt/ -type f -mtime +30

echo "im deleting the files"

find /opt/ -type f -mtime +30 -delete

delete only .txt files on abc folder

command to modify the file times: touch -t 202208311530 \* (The “-t” option modifies the time stamp of the file and the format is YYYYMMDDHHMM.)

find /root/abc -name "\*.txt" -type f -mtime +30

echo "im deleting files with are older than 30 days"

find /root/abc -name "\*.txt" -type f -mtime +30 -delete