**class 8 : 25-10-2022**

1. core part of linux is kernel. kernel makes the system understand your language. shell is an interpreter
2. shell interprets to kernel by converting the input to binary language and kernel talks to hardware.
3. different types of shell are c shell, k shell and bash shell.
4. scripting : combination of logically connected commands is called scripting.
5. the first line inside script is called shebang. it invokes bash shell. if i don't write. it uses default shell.

#!/bin/csh ; #!/bin/ksh

1. vi filename . sh
2. #!/bin/bash --- > shebang
3. echo "hi this is abc and i am from blore"
4. echo "below is the list of current dirs"
5. ls –lrt (if this doesn’t work, try giving an empty line previous to this command)
6. how to execute shell script

./filename.sh

or

sh filename.sh

or

bash filename.sh

1. incase permission denied to execute, change permission of file using chmod.
2. try to execute again now
3. echo $SHELL ---- > to know which shell you are using.

note: .sh after giving filename of shell script is not compulsory.

------------------------------------example2.sh---------------------------------------

1. variables : that changes during the execution of a program. need to define it in shell script by assigning values to variables. declaration of variables is not needed in shell script.
2. if u don’t use “ “ in echo command while defining the variable, it will only consider the first word and not the second. in below example it will consider only xyz if u don’t use “ “.
3. $ prints the variable, if u don’t mention then it will consider as sentence.
4. instead of var , var1 etc , you can type anything man can tan dan etc etc.
5. to view the file cat filename .sh

#!/bin/bash

var=abc

var1=blore

var2="xyz 1234"

echo "this is $var and i am from $var1"

echo “$var2”

eg :

data=blore

tata=jamshedpur

kata=home $ data $ tata

echo “i love $ kata”

-------------------------------example3.sh-----------------------------------------------

eg :

man=manasa

man1=mlore

man2="$man1"

man3="9.1 & also 10 pointer in my final sem"

man4="$man bhat"

man=mansa

echo "my name is $man"

echo "i am from $man1"

echo "i completed my puc in $man2"

echo "i scored $man3 during my graduation"

echo "$man4 is not my full name"

ls -lrt

in this case it will overwrite manasa as mansa in man=manasa and not in formula used below i.e man4=$man as u wrote the formula before defining the new command.

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

#!/bin/bash

echo "this is $1 and i am from $2" ---- > this is called command line and $1 , $2 are called arguments.

execute this ./filename . sh manasa mlore --- >(no need to add .sh after the filename, its not compulsory)

it will print as this is manasa and i am from mlore.

here ./filename . sh is considered as $0

manasa as $1

mlore as $2

eg : echo "this is $1 and i am from $2

execute this ./filename “manasa bhat” “mlore & mysore”

it will print as this is manasa bhat and i am from mlore & mysore.

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

how do you read arguments in shell script?

we can pass arguments to shell script separated by spaces

we can read them using $1 $2 $3... inside shell script

$1 ---> 1st argument

$2----> 2nd argument

${10} ---> 10th argument

${12} --- > 12th argument

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

System defined arguments are

$# --> total number of args passed to script

$\* --> all args passed to script

$? --> used to check status of last executed command, 0 means success and non-zero is failure

$@ --> all args passed to script but stored in array format

$! ---> PID of last command went into background

$$ --> PID of current running process

$0 --> name of the script itself

eg : ./filename . sh a b c

no .of args passed is 3.

eg :try defining var1= $\*

$? eg : type ls , then type echo $? answer is 0 as ls was executed.

now grep for a string that does not exists in ur file , say grep “newyear” filename

then type echo $? answer is 1 as grep was failed.

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

how do you run script in background?

use & symbol at end of script.

./script1 &

when the script is big, you run it in background. if you close the session it will close.

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

ps -ef | grep "script name"

how do you bring it back to foreground?

fg pid ---> for small scripts it gav msg saying no such job

difference between ./script & and nohup ./scrpt &

./filename & ---> script will run in background and stops its execution once you close session or terminal

nohup ./filename .sh & (sir used this)

nohup (no hang up)--> run script in background and will complete execution even after closing session or terminal

note:

1. you cannot use nohup without & whereas u can use & without nohup.
2. output of scripting can be stored in file using >

**class 9 : 27-10-2022**

**If statement**: The if statement is used to check a condition and if the condition is true, we run a block of statements (called the if-block), else we process another block of statements (called the else-block).

simple if statement

if [ condition ] ---> if true

then

statement --->then print this statement.

fi

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

write a script to check given number is 5 or not

#!/bin/bash

if [ $1 -eq 5 ]

then

echo "$1 is five"

fi

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

numeric comparison

-eq --> equals

-ne --> not equal to

-gt --> greater than

-ge --> greater than or equal to

-lt --> less than

-le --> less than or equal to

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

string comparison

== --> equal to

>= --> greater than or equal to

<= --> less than or equal to

!= --> not equal to

> ---> greater than

< ---> less than

&& --> and

|| ---> or

! --> not

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

if.... else....

if [ condition ]

then

statment1

else

statement2

fi

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

#!/bin/bash

if [ $1 -eq 5 ]

then

echo "$1 is five"

else

echo "$1 is not five"

fi

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

write a script to find biggest of two numbers

#!/bin/bash

If [ $1 -gt $2 ];then

echo "$1 is big" (# 1st argument)

else

echo "$2 is big"

fi

here if i pass equal no.s it wont work. also note that it will print second statement as 1st statement is false.

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

restrict users to pass only two args --->when user gives more args, it should give error msg.

#!/bin/bash

if[$# -eq 2];then

type above script

else

echo “pass only 2 arguments”

fi

in this case the entire script will run before typing “pass only 2 arguments” which will take longer time. so we use below command where it will exit from the script.

or

#!/bin/bash

if [ $# -ne 2 ];then

echo "pass 2 args only"

exit 1

fi

if [ $1 -gt $2 ];then

echo "$1 is big" # 1st argument

else

echo "$2 is big"

fi

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

to find biggest of 3 numbers and restrict to only 3 args.

#!/bin/bash

if [ $# -ne 3 ];then

echo "pass 3 args"

exit 1

if [$1 –gt $2] && [$1 –gt$3];then

echo “$1 is big”

elif [$2 –gt $3];then

echo “$2 is big”

else

echo “$3 is big”

fi

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

if [ condition ];then

statement

elif [ condition1 ];then

statement

elif [ condtion2];then

statement

else

default statement

fi

To find 2nd biggest of 3 numbers

#!/bin/bash

If [ $1 > $2 ] && [ $1<$3 ];then

echo “ $1 is the 2nd biggest number”

elif [ $2 > $3 ];then

echo “$3 is the 2nd biggest number”

else

echo “$2 is the 2nd biggest number”

fi

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

note:

1) environment variable --> variable which can be exported to subshell. Also called as global variable.

2) very important environment variable is path.

3) all environment variable are defined in capital letters and local variables are defined in small letters.

4) local variables are defined inside a script and their usage is restricted to the script. environment variable is available for whole system.

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

which ls ----> gives location/full path of command

which pwd --- > gives location of command

type env ---> to get list of all environment commands present in my system.

Note : bin is present before “home” folder everything. i.e do cd ..when u r inside home folder. Then do ls to find bin , etc , var…

wget ---> download any file/ directory from url(browser) on linux system.

echo $PATH , echo $SHELL , echo $USER ---> are examples of envi var. To check the content of the variable.

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

To convert a script to command :

1. cd dir/ ----> enter the folder that has all the shellscript
2. pwd
3. copy the entire path of the dir by clicking on left cursor key.
4. PATH="$PATH:/home/ec2-user/SHELLSCRIPT"

Now files inside shell script will work like command.

Eg “filename 4 5 6”

This will give biggest of 3 numbers.

Note:

1. When u log out and login the path of shell script will not be there.
2. So u will need to use bash command.
3. Open vi .bashrc in home folder
4. At the end type the below line

PATH=”$PATH :/home/ec2-user/dirname”

1. Save and exit
2. Open duplicate session
3. Now when u just type the name of the script, it will execute. Don’t have to type sh scriptname.

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

Defining envi variable and use it in shell script

On regular command page type the foll

VAR1=abc

echo $VAR1

This will give result as abc

export VAR

Now enter the script and type

Echo “$VAR”

Save and exit

This will display abc

Now try to write script in home folder

VAR2=sh xyz.sh

Echo $VAR2 ----> Should give output of script

Vi .bashrc

Type sh.xyz.sh in end of .bashrc

Save and exit

Open duplicate session. It should display output of script.

NOTE : when u logout and login the command wont work again.

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

**while loop :** A "While" Loop is used to repeat a specific block of code an unknown number of times, until a condition is met.(or is it until the condition remains true)

It executes the command in iterations. When the output becomes false, it stops the iterations.

while [ condition ]; do

statement

done

**Until loop : Until** loop is used to execute a block of code until the expression is evaluated to be false. This is exactly the opposite of a [while loop](https://www.tecmint.com/different-ways-to-read-file-in-bash-script/). While loop runs the code block while the expression is true and **until** loop does the opposite.

------------------------print5.sh---------------------------

To type ascending order of number.

#!/bin/bash

num=$1

while [ $num -gt 0 ];do

echo $num

num=`expr $num - 1`

done

NOTE:

1. To collect output of a command to a file we use >
2. To collect output of a command to a variable we use ---> Expr in above formula means mathematical expression used to add,multiply,subtract and divide.

Eg:

var=`ls-lrt`

echo $var

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

assignment :

1. modify the script to find equal numbers in greatest of 3 numbers

#!/bin/bash

if [ $1 –eq $2 ] && [ $1 –eq $3 ];then

echo “ $1 $2 $3 are equal”

elif [ $1 –gt $2 ] && [ $1 –gt $3 ];then

echo “$1 is the greatest”

elif [ $2 –gt $3 ];then

echo “$2 is the greatest”

else

echo “$3 is the greatest”

fi

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

how to set environment variable and export it to subshell.

VAR=abc

Echo $VAR ---> abc

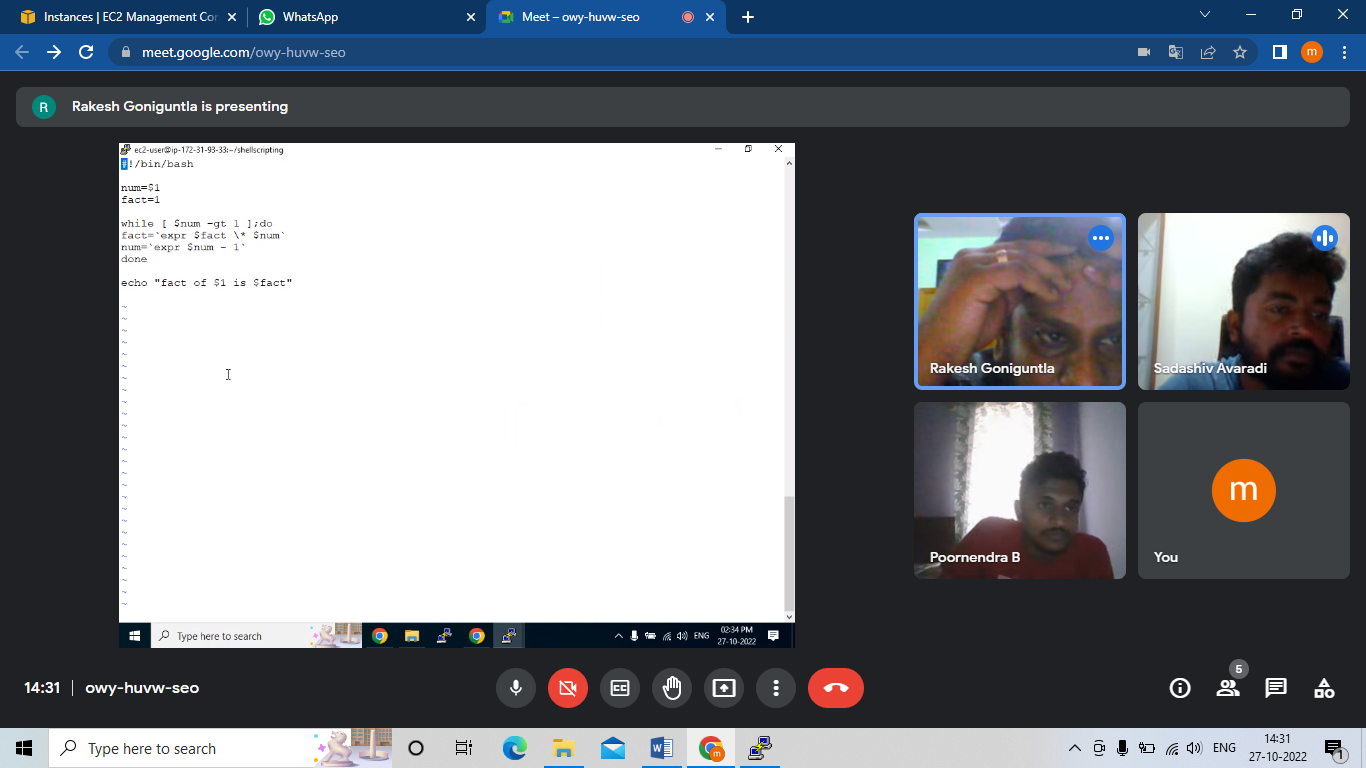
export VAR

open file in vi

type echo “$VAR”

execute the file

write a script to find factorial of a number



**class 10 : 28-10-2022**

special characters ---> special characters have special meaning to shell.. special characters are \* $\* $$ etc...

escape character (\): escape special meaning to shell

echo "\$?"

echo "\$$"

\*-----> means “all” as per linux

\ ----> it wil ESCAPE the special meaning

Eg1: \\* -----> multiplication

Eg2: Say I want to print echo” Hi how r u” ---> will give results Hi how r u

Now try to print echo “$0” ----> it will give results of the special charater i.e script name.

To escape the special meaning type echo “\ $?” ---> output will be \?

Eg3: Replace space with : in a content ----> sed ‘s/ /:/g’ filename

Now try to replace space with / ---- > sed ‘s/ /\/ /g’ filename

Eg4: To print \ on console ----> echo “\ \”

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

write a script to check given name is a file or dir or link

#!/bin/bash

echo "enter name to check"

read name

if [ -f $name ];then

echo "$name is a file"

elif [ -d $name ];then

echo "$name is a dir"

elif [ -L $name ];then

echo "$name is a link"

else

echo "$name does not exist"

fi

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

-f --> file exists

-d --> directory exists

-L ---> link

-e --> exist or not

-r --> read permission

-w --> write permission

-x --> execute permission

-z ---> length of string is 0, empty string or empty word.

-s ----> if lines exists/ its not an empty line

how to file empty or not

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

how do read file line by line??

while read line

do

echo $line

done < filename

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

To print all the lines of a file

While read line

do

echo $ line

done < $1

Note:

1. if u r not inside the folder that contains the file while executing the command, u will have to type the entire path of the file while giving input(passing argument) to the script

Eg: sh filename.sh /home/ec2-user/dir1/file1

1. can type the filename here instead of $1. Will directly execute output of the file.

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

To add line numbers to the content of the file and display it.

Num=1

While read line

Do

Echo “$num $ line”

Num=`expr $num + 1`

Done < $1

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

write a script to find number of words in each line of a file

#!/bin/bash

number=1

while read line

do

words=`echo "$line" | wc -w`

echo "$number : $words"

number=`expr $number + 1`

done < $1

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

To count number of words in each line and print line numbers

#!/bin/bash

Num=1

Echo “S.no no. of words”

While read line

Do

Words=`echo “$line” | wc –w`

Echo “$num $words”

Num=`expr $num +1`

Done < $1

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

write script to add line number to each line of a file

#!/bin/bash

line\_num=1

while read line;do

echo "$line\_num. $line"

line\_num=`expr $line\_num + 1`

done < $1

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

write a script to segregate only odd lines and store in file called log\_odd and even lines in a fille called log\_even

#!/bin/bash

number=1

while read line;do

status=`expr $number % 2`

if [ $status -eq 0 ];then

echo "$line" >> log\_even

else

echo "$line" >> log\_odd

fi

number=`expr $number + 1`

done < $1

echo "==============odd lines============"

cat log\_odd

echo "====================even lines============"

cat log\_even

rm log\_odd log\_even ----> this is req coz as u execute the script multiple times, the executed data will be appending to the log\_odd and log\_even every time.

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

write a script to find employees if thier age is more than 50 from given data file

NAME EMPID AGE

ABCD 20005 036

XYZA 20006 056

AJAY 20007 034

VIJI 20008 054

GAAA 20009 025

DDDD 20010 042

#!/bin/bash

sed '1d' $1 > temp ---> u store the data without the heading like and use that.

while read line;do

age=`echo "$line" | awk -F " " '{print $3}'`

if [ $age -gt 50 ];then

echo "$line" | awk -F " " '{print $1}'

fi

done < temp ----> very imp step. Use the data without header and not the original file.

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

To debug the shell script: Debug means to identify the reasons for the error.

Just go inside any script and type the following at the start of the script below shebang. **Set -x**

Assignment:

write a script to find employees if thier age is between 30 and 50

#!/bin/bash

sed "1d" $1 > abc

while read line;do

age=`echo "$line" | awk -F " " '{print $3}'`

if [ $age -gt 30 ]&& [ $age -lt 50 ];then

echo "$line" | awk -F " " '{print $1}'

fi

done < abc

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

write a script to find employees for given age

#!/bin/bash

sed "1d" $1 > abc

echo " enter the age "

read age

while read line;do

data=`echo "$line" | awk -F " " '{print $3}'`

if [ $data -eq $age ];then

echo "$line" | awk -F " " '{print $1}'

fi

done < abc

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

To check whether file is empty or not

#!/bin/bash

echo "enter the name of the file to be checked if empty or not"

read name

if [ -s $name ];then

echo "its not an empty file"

else

echo "its an empty file"

fi

done < $1

**class 11 : 29-10-2022**

write a script to reverse a file, 1st line should be printed as last line, 2nd line as 2nd last line and so on...

using head and tail command:

#!/bin/bash

line=`cat $1 | wc -l`

while [ $line -gt 0 ]

do

head -$line $1 | tail -1 >> rev

line=`expr $line - 1`

done

cat rev

rm rev

using sed command:

#!/bin/bash

set -x

line=`cat $1 | wc -l`

while [ $line -gt 0 ]

do

sed -n "$line""p" $1 >> revsed

line=`expr $line - 1`

done

cat revsed

rm revsed

To cut character in a word:

**Echo “vijay” | cut –c2 ----> cuts and prints i**

**Echo “manasa” | cut –c5 ----> cuts and prints s**

To cut word from a line:

**Echo “ my name is manasa” | awk –F “ “ ‘{ print$NF }’**

To reverse a string

#!/bin/bash

echo "Enter the word"

read word

c=`echo $word | wc -c`

while [ $c -gt 0 ];do

echo "$word" | cut -c$c >> log1

c=`expr $c - 1`

done

cat log1 |xargs

rm log1

**for loop:** it reads set of values one by one

for i in val1 val2 val3

do

statement

done

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

-----------------------------------------add-number.sh-----------------------------------------------

Write a script to add numbers

#!/bin/bash

sum=0

for i in 2 4 6 7

do

sum=`expr $sum + $i`

done

echo "sum is $sum"

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

Write a script to add numbers

#!/bin/bash

var="2 5 7 9 10"

sum=0

for i in $var

do

sum=`expr $sum + $i`

done

echo "sum is $sum"

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

Write a script to add numbers when numbers have to be entered in command line

sum=0

for i in $\*

do

sum=`expr $sum + $i`

done

echo "sum is $sum"

how do you debug shell script

set -x

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

Write a script to add defined numbers to a series of number

#!/bin/bash

Echo “Enter the number to be added”

Read num

for i in $\*

do

i=`expr $i + $num`

done

echo “ Result of addition is $i ”

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

write a script to find factorial for given set of numbers

#!/bin/bash

for i in $\*

do

num=$i

fact=1

while [ $num -gt 0 ];do

fact=`expr $fact \\* $num`

num=`expr $num - 1`

done

echo "fact of $i is $fact"

done

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

execute command or script on remote server

**ssh -i key.pem user@server2 "/home/ec2-user/scripts/big2.sh 3 4"**

**ssh -i key.pem user@server2 "ls -lrt"** -----> if u need to execute home folder

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

Realistic script:

interview script 1)

I have written a script to check disk size. if disk size is more than 90%, send email notification saying disk size is 90%.

please take approriate action

Note: This script wont work as of now. Just practise template.

-----------------------checksize.sh--------------------------------------

#!/bin/bash

size=`df -h . | tail -1 | awk -F " " '{print $(NF - 1)}' | sed 's/%//g'`

echo –e “Hi \n Disk size is $size \n Please take appropriate actions \n \n Thank you “ > file

if [ $size -gt 90 ];then

cat file | mail -s "memory reached 90%" -c "abc@gmail.com" xyz@gmail.com

fi

Here cat file ----> body of mail

“memory reached 90%” ---> subject of mail

-c [abc@gmail.com](mailto:abc@gmail.com) ----> cc person’s mail ID

-b [123@gmail.cim](mailto:123@gmail.cim) ---> bcc person’s mail ID

[xyz@gmail.com](mailto:xyz@gmail.com) ---> Recievers mail ID

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

Crontab : This is a scheduler. Defines when the mail has to be sent to the receiver.

Can schedule multiple scripts inside one crontab.

**crontab -e** --> edit

**crontab -l** ---> list cron jobs

Type **sudo crontab –e** incase u don’t hav permission

day of week

00 --> sunday

01 --> monday

02 --> Tuesday

03 --> wed

04 --> Thur

05 --> Fri

06 --> Sat

schedule script to run at 11.30am on 4th july

\*\*\*\*\* ---> min hr date month day of week

30 11 04 07 01 /home/ec2-user/scripts.sh ---> full path of the file location.

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

schedule to run at 6.15pm today

15 18 04 07 01 /home/ec2-user/scripts.sh

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

schedule script to run at 7.30pm only on tuesday

30 19 \* \* 02 /home/ec2-user/scripts.sh

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

schedule script to run at 5.15pm every day

15 17 \* \* \* /home/ec2-user/scripts.sh

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

schedule script to run at 4.30pm on monday and friday

30 16 \* \* 01,05 /home/ec2-user/scripts.sh

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

schedule script to run at 4.30pm on monday to friday

30 16 \* \* 01-05 /home/ec2-user/scripts.sh

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

schedule script to run every 30mins on Wednesday -----> this will consider 30th min from the time we save the file.

\*/30 \* \* \* 03 /home/ec2-user/scripts.sh

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

every 5 mins on monday

\*/5 \* \* \* 01 /home/ec2-user/scripts.sh

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

every min every day

\* \* \* \* \* /home/ec2-user/scripts.sh

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

schedule script to run at 1st of every month at 11.30am

30 11 01 \* \* /home/ec2-user/scripts.sh

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

schedule script to run last day of every month

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

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

mail or mailx

cat filename | mail -s "subject" -c "abc@gmail.com" xyz@gmail.com,ajay@gmail.com

or

mail -s "subject" -c "abc@gmail.com" xyz@gmail.com,ajay@gmail.com < filename

or

echo -e "Hi,\n body of the mail" | mail -s "subject" -c "abc@gmail.com" [xyz@gmail.com,ajay@gmail.com](mailto:xyz@gmail.com,ajay@gmail.com)

Assignment:

Write a script to find biggest of n numbers

#!/bin/bash

set -x

num=1

for i in $\*

do

if [ $i -gt $num ];then

num=$i

else

num=$num

fi

done

echo "$num is greatest"

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

Write a script to find smallest of n numbers:

#!/bin/bash

num=$1

for i in $\*

do

if [ $i -lt $num ];then

num=$i

else

num=$num

fi

done

echo "$num is smallest"

Write script to add specific number to the given numbers

#!/bin/bash

echo "type the number to be added to above numbers"

read num

for i in $\*

do

sum=`expr $i + $num`

echo " $sum " >> log

done

cat log

rm log

Write a script to find 2nd biggest of n numbers

#!/bin/bash

set -x

num=1

for i in $\*

do

if [ $i -gt $num ];then

num=$i

else

num=$num

fi

done

big2n=`echo "$\*" | sed "s/${num}//g"`

x=1

for i in $big2n

do

if [ $i -gt $x ];then

x=$i

else

x=$x

fi

done

echo "2nd biggest number is $x"

Write a script to find 2nd smallest of n numbers

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

Write a script to find prime numbers.

#!/bin/bash

echo -e "\n Enter the number to check if it is a prime number or not \n"

read n

for((i=2;i<=$n-1;i++))

do

num=`expr $n % $i`

if [ $num -eq 0 ];then

echo " $n is not a prime number"

exit 0

fi

done

echo " $n is a prime number"

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

**class 12 : 30-10-2022**

if any of these services stoppped, we should get email notification saying tomcat service stopped, pls take appropriate action

#!/bin/bash

services="ser1 ser2 ser3....."

for i in $services;do

ps -ef | grep "$i"

if [ $? -ne 0 ];then

echo "$i" >> stopped-services

fi

done

if [ -s stopped-services ];then

cat stopped-services| mail -s "service stopped" [abc@gmail.com,xyz@gmail.com](mailto:abc@gmail.com,xyz@gmail.com)

fi

rm stopped-services

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

**SWITCH**: Just like if and else.

case $var in

value1) statement1

;;

value2) statement2

;;

value3) statement3

;;

\*) statement4

;;

esac

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

To find out if given numbers are odd or even numbers.

#!/bin/bash

case $1 in

1|5|7|9) echo "this is odd number"

;;

2|4|6|8|10) echo "this is even number"

;;

16) echo "this is sixteen"

;;

\*) echo "Invalid number"

;;

esac

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

#!/bin/bash

case $1 in

1|5|7|9) echo "this is odd number"

;;

2|4|6|8|10) echo "this is even number"

;;

mon) echo "this is monday"

;;

'tue'|'wed') echo "this is tuesday or wednesday"

;;

\*) echo "Invalid number"

;;

esac

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

Odd and even number

odd=$1 % 1

even=$1 % 2

case $1 in

odd) echo “this is odd no” check if $ is req before odd and even

;;

even)echo “this is even no”

;;

\*) echo “Give valid input”

;;

Esac

write a menu based script to do followings

1) monday, create a file log1 and log2

2) Tuesday, rename files log1 to log\_1 and log2 to log\_2

3) wednesday, copy file log\_1 to log1\_backup and log\_2 to log2\_backup

4) Thursday, redirect output of l-lrt to log-list

5) friday, delete log1\_backup and log1\_backup

6) sat and sun, holiday

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

#!/bin/bash

case $1 in

'mon'|'monday') echo "creating files"

touch log1 log2

;;

'tue'| 'tuesday') echo "rename files"

mv log1 log\_1

mv log2 log\_2

;;

'wed'|'wednesday') echo "take backup of files"

mv log\_1 log1-backup

mv log\_2 log2-backup

;;

'thu'|'thursday') echo "redirect list files to file "

ls -lrt > log-list

;;

'fri'|'friday') echo "remove files"

rm log1-backup log2-backup

;;

\*) echo "today is holiday"

;;

esac

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

write menu based script for followings

./menu.sh

1) searchword

2) checkname

3) find file

4) create link

5) edit file

6) exit

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

#!/bin/bash

echo "below is the menu"

echo -e "\n1) searchword\n2) chekcname\n3) findfile\n4) creatlink\n5) edit file\n6) exit\n"

echo "select options from the above menu"

read opt

case $opt in

1) /home/ec2-user/shellscripts/searchword.sh

;;

2)/home/ec2-user/shellscripts/checkname.sh checkname.sh is filename

;;

3) /home/ec2-user/shellscripts/findfile.sh

;;

4) /home/ec2-user/shellscripts/createlink.sh

;;

5)/home/ec2-user/shellscripts/editfile.sh

;;

\*)echo "you have selected option to exit from the script"

;;

esac

--------------------------searchword.sh-----------------------------------------------

#!/bin/bash

echo "enter word to search"

read word

grep -R -l "$word" \* > log\_word

if [ $? -eq 0 ];then

echo "$word is found in below files"

cat log\_word; rm log\_word

else

echo "$word is not found in any files"

fi

------------------------------------checkname.sh---------------------------------------------

#!/bin/bash

echo "enter name to check"

read name

if [ -f $name ];then

echo "$name is a file"

elif [ -d $name ];then

echo "$name is a dir"

elif [ -L $name ];then

echo "$name is a link"

else

echo "$name does not exist"

fi

-----------------------------------findfile.sh--------------------------------------------

#!/bin/bash

echo "enter file to find its location"

read file

find /home/ec2-user -iname "$file" > log\_file

if [ -s log\_file ];then or if [ $? –eq 0 ];then

echo "$file is found in below location"

cat log\_file;rm log\_file

else

echo "$file is not found"

fi

------------------------------------------------createlink.sh---------------------------------

#!/bin/bash

echo "enter file name to create softlink"

read file

if [ ! -f $file ];then

echo "$file is incorrect. please enter correct filename"

exit 1

fi

echo "enter link name "

read link

if [ -L $link ];then

echo "$link already exists"

exit 1

fi

ln -s $file $link

if [ $? -eq 0 ];then

echo "softlink $link created successfully for file $file"

ls -lrt $link

fi

Note:

1. complete path of the file while giving input , Eg : cd /home/ec2-user/shellscripts/temp/temp1
2. Exit 1 means exit if the script fails. Exit 0 means exit if the script succeeds.
3. Read link ---> It will check if the link already exists in present directory only
4. Now say I want to create the link in different folders and not the present folder,
   1. While giving the input link type the entire path before the link ,where the link has to be created.
   2. Use mv inside the script after the link has been created.
   3. Did not note this down. It was inside the script to use cd/home/ec2..

--------------------------------editfile------------------------------

#!/bin/bash

echo "enter filename to edit"

read file

if [ -f $file ];then

vim $file

else

echo "$file doe not exist"

fi

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

this assignment is mandatory

1. if choose any option, script should not exit after executing that particular option.

it should keep display menu again and again till I select exit button explicitly

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

1. hotel menu display

hotel.sh

menu should be displayed

1) idli

2) dosa

3) samosa

4) Pallav

5) coffee/tea

6) exit

each option should have sub-menu with rate. As u select each option it should return back to main menu.

and display all items selected and total price (bill to be generated) when exit from script

Refer the folder hotel menu

1. install tomcat on ubuntu and download calendar.war file and copy it to /webapps folder under tomcat

Use wget to download tomcat

Check security --> 80,443,is available

Public server 8080

Note : Public server for Ubuntu is Ubuntu.

To check if tomcat process is running

Ps –ef | grep “tomcat”

1. script to create 3 three users to be added 3 different groups

#!/bin/bash

echo "Enter the two user name and two group name"

read $1 $2 $3 $4

for i in $1 $2

do

sudo useradd $i

echo "User name created with name of $i"

done

for j in $3 $4

do

sudo groupadd $j

echo "New groups created with name of $j"

done

sudo usermod -G $3,$4 $1

sudo usermod -G $3,$4 $2

cat /etc/group | grep -e "$3" -e "$4"

1. script to move lines contains "linux" word to x folder & move lines contains "windows" word to y folder

#!/bin/bash

echo "Enter the file that contains the word"

read line

grep -i "linux" $line >>linux

grep -i "unix" $line >>unix

grep -i "windows" $line >>windows

cat linux

cat unix

cat windows

rm linux unix windows

1. Script to find a particular word in any file.

#!/bin/bash

echo "Enter the word to be searched"

read word

echo "Enter the file that contains the word"

read line

grep -i "$word" $line >>abc

cat abc

rm abc

1. script to create directory called folder in different servers using ip address

#!/bin/bash

ssh ec2-user@172.31.38.131 "mkdir Folder1"

ssh ubuntu@172.31.46.150 "mkdir Folder1"

scp smalln.sh ec2-user@172.31.38.131:/home/ec2-user/Folder

scp smalln.sh ubuntu@172.31.46.150:/home/ubuntu/Folder

rsync -avzp /home/ec2-user/hotel ec2-user@172.31.38.131:/home/ec2-user/Folder

rsync -avzp /home/ec2-user/hotel [ubuntu@172.31.46.150:/home/ubuntu/Folder](mailto:ubuntu@172.31.46.150:/home/ubuntu/Folder)

1. script to login to different user of 2nd server
2. script to print odd numbers b/w 20 to 40

*OPTION 1:*

#!/bin/bash

for ((i=20;i<=40;i++))

do

num=`expr $i % 2`

if [ $num -eq 0 ];then

echo " $i is even " >>even

else

echo "$i is odd" >>odd

fi

done

cat even

cat odd

rm even odd

*OPTION 2:*

#!/bin/bash

echo " Print 2 numbers to define the range of numbers "

read s e

for ((i=$s; i<=$e; i++))

do

num=`expr $i % 2`

if [ $num -eq 0 ];then

echo " $i is even " >>even

else

echo "$i is odd" >>odd

fi

done

cat even

cat odd

rm even odd

1. script to print prime number between 1 to 20.
2. Question sent to rakesh on read write permission
3. Command to print extension also write is as script.

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

To check if server mentioned in a file are up and running - IP address mentioned row wise

while read line

do

ping -c 1 $line

if [ $? -eq 0 ];then

echo " $line " >> ip\_running

else

echo " $line " >> ip\_notrunning

fi

done < $1

echo " IP that are UP IP that are down "

paste ip\_running ip\_notrunning

rm ip\_running

rm ip\_notrunning

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

To check if server mentioned in a file are up and running - IP address mentioned column wise

echo " Enter the file which has IP adress"

read file

var=`cat $file`

for i in $var

do

ping -c 1 $i

if [ $? -eq 0 ];then

echo " $i " >> ip\_running

else

echo " $i " >> ip\_notrunning

fi

done

echo " IP that are UP IP that are down "

paste ip\_running ip\_notrunning

rm ip\_running

rm ip\_notrunning

Say u hav a script. It should if another script has found a particular word. It should keep checking this in loop. If it finds the word, then your script should exit from the loop.

#!/bin/bash

echo "Enter the file name"

read file

while read line

do

cat "$file" | grep "abc"

if [ $? -eq 0 ];then

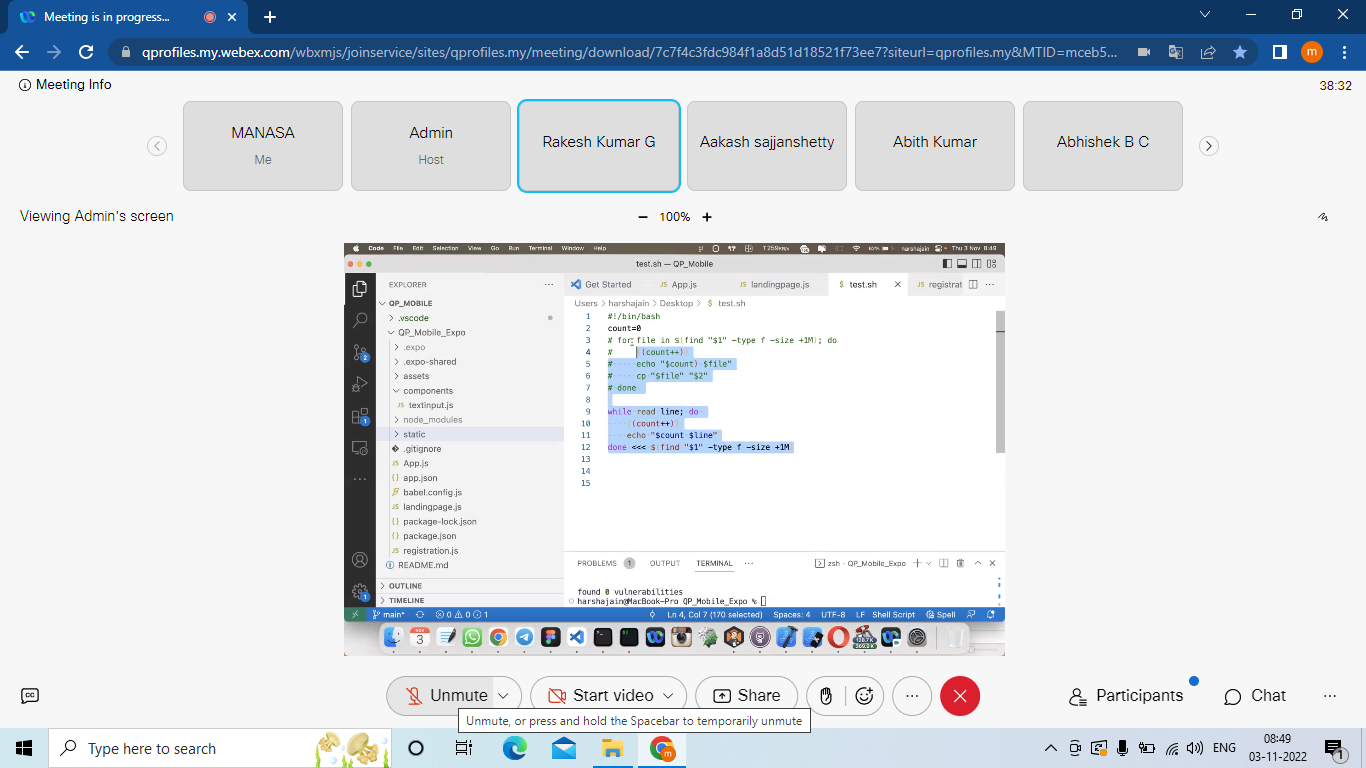
echo " abc found "

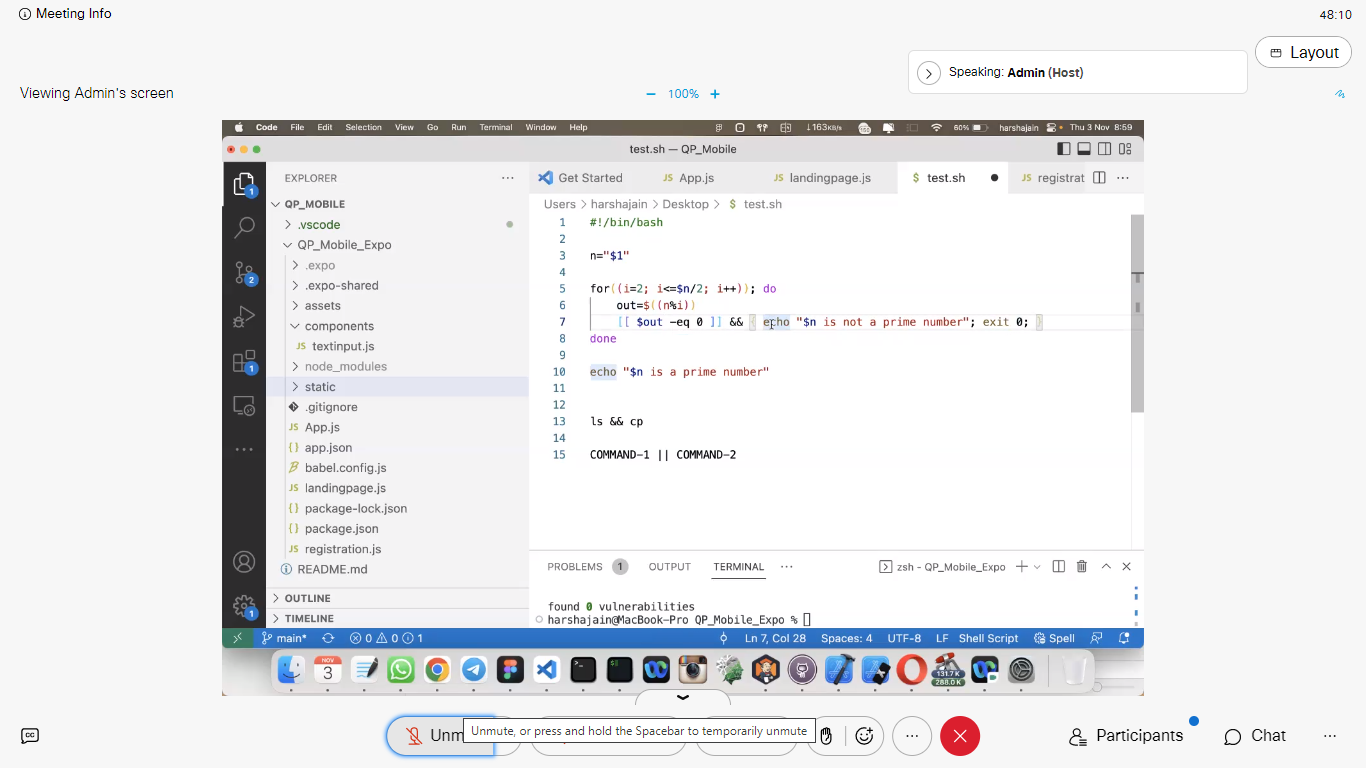
break

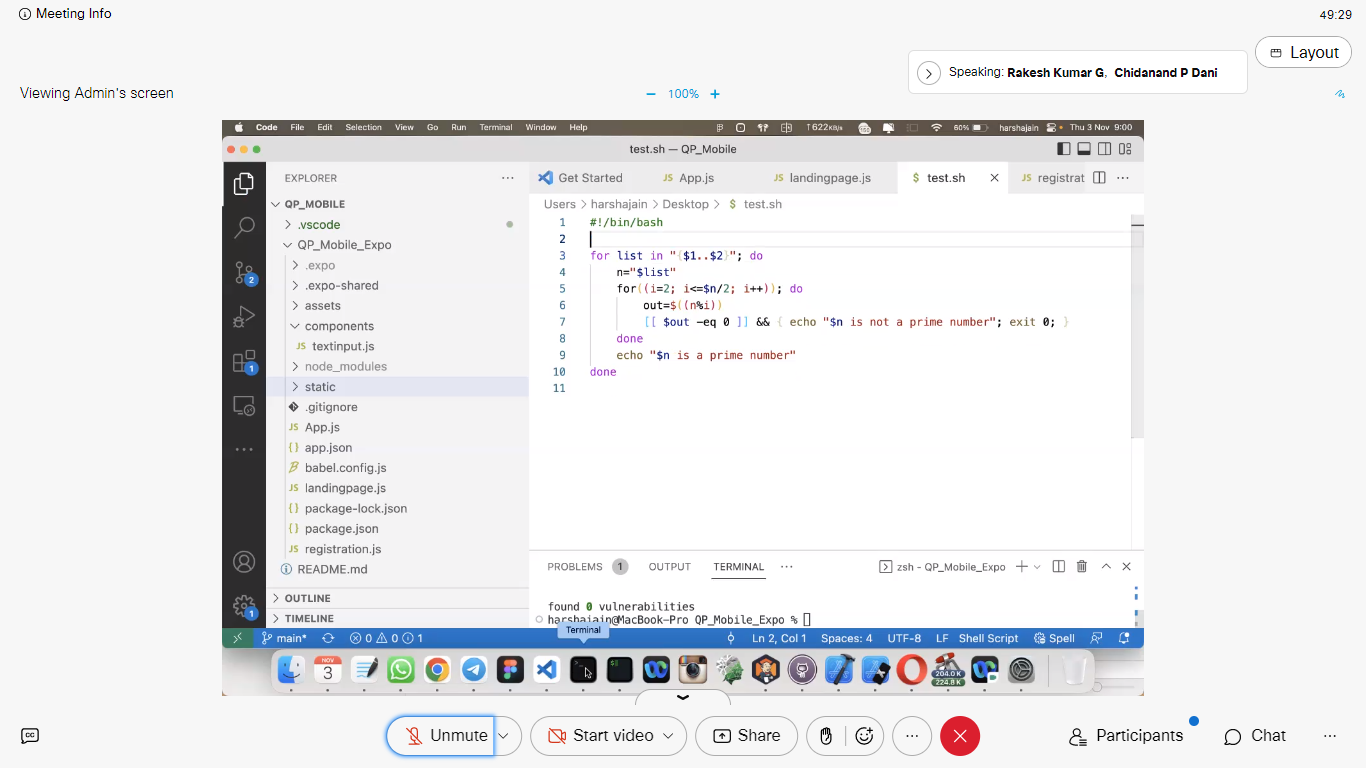
fi

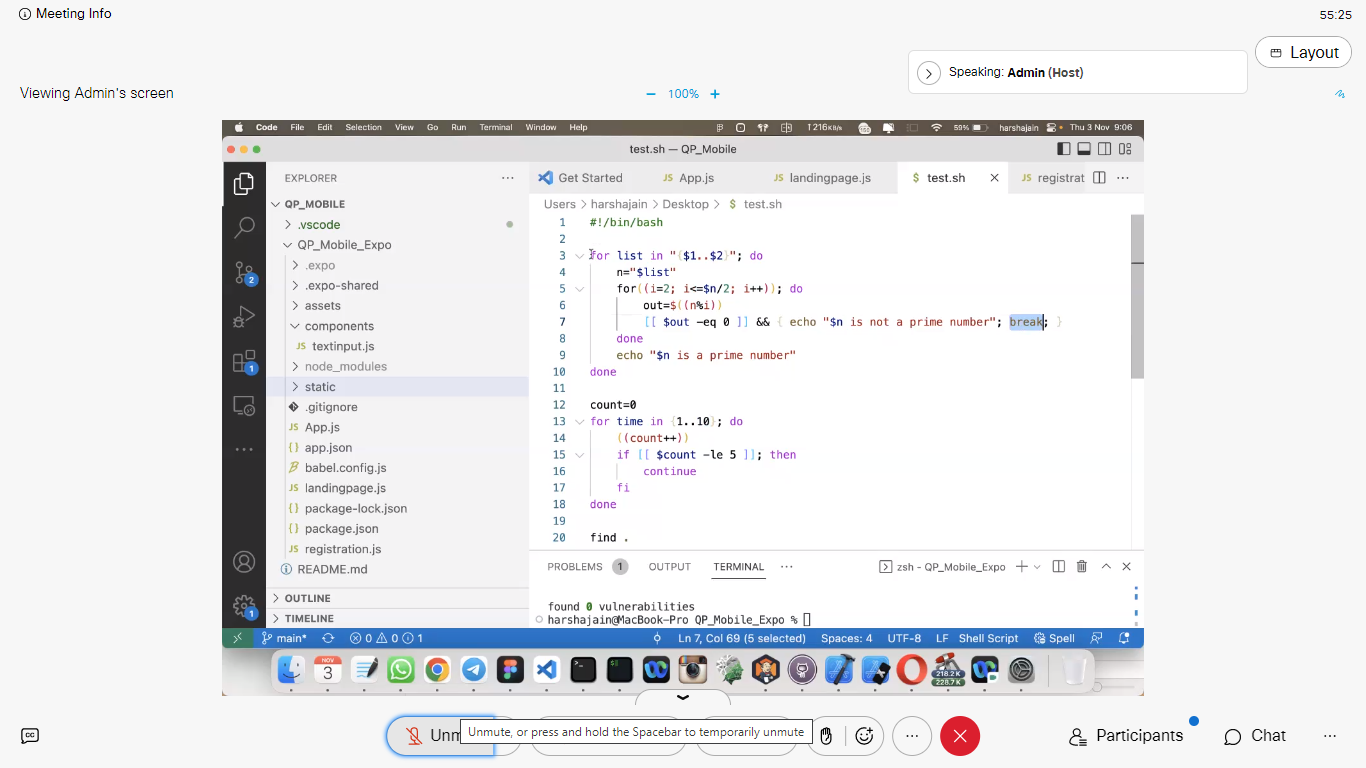
done

**class 13 : 01-11-2022**









To search if path exists. Relative path and absolute path

