Linux Assignment Lab 3 (Solutions)

1.Display the Primary and Secondary prompt. Change the primary prompt to your name: temporarily

->PS1="Rahul" , sh

2: As soon as you login, the prompt should be changed to your name:

also the name of the home directory should be automatically displayed.

->In .bashrc ---> PS1="Rahul" and echo"$HOME"

3: Check the content of the Environmental variable SHELL.

->echo $SHELL

4:Try the below exercise and check the output.

->$continent=”Africa”

$echo “$continent”

------------→ Africa

$sh

$echo “$continent”

------------→ No Response

$continent=”Asia”

$echo “$continent”

------------→ Asia

$ctrl + d

$echo “$continent”

------------→ Africa

$sh

$echo “$continent”

------------→ No Response

$ctrl + d

5:Try the below exercise and check the output. (Export variables)

->export continent in parent shell if changed in child shell changes but remain constant in parent shell.

$continent=”Africa” export continent

$echo “$continent”

------------→ Africa

$sh

$echo “$continent”

------------→ Africa

$continent=”Asia”

$echo “$continent”

------------→ Asia

$ctrl + d

$echo “$continent”

------------→ Africa

6. Write a shell script that takes the user name as input and reports whether he / she has logged in or not.

-> echo enter user name

read username

if users | grep $username>/dev/null

then

echo username logged in

else

echo username not logged in

fi

7: Write a shell script to display the file name and its contents of all the files that is there in the current directory.

->for i in \*.\*

do

echo "$i"

cat $i

done

8: Write a shell script, which will take a file name as argument and check whether the file exists and display its access permissions for user.

->echo enter the filename

read file

if [ -f $file ]

then

echo file found

echo `ls -l seven.sh | cut -d" " -f1|cut -b2,3,4`

else

echo file not found

fi

9: Pass three numbers as command line arguments and display the largest number in the given three numbers.

->if [ $1 -gt $2 -a $1 -gt $3 ]

then

echo maximum: $1

elif [ $2 -gt $1 -a $2 -gt $3 ]

then

echo maximum: $2

else

echo maximum: $3

fi

10: Write a shell script which will accept a pattern and a file name. The pattern will be searched in the file provided. Display appropriate messages and perform necessary validations on file.

->echo enter the filename

read file

echo enter the pattern

read pat

if grep $pat $file >/dev/null

then

echo pattern found

else

echo pattern not found

fi

11: To create a menu program for a) creating a file, b) Creating a directory, c) copying a file,

d) moving a file. (use functions)

a. If the file exists already give the appropriate message

b. If the dir exists already give the appropriate error message

c. Source file should exist if not give a message, It should have read permission if not another message, Destination file either there or not, if not there then create it and copy it. If there, then ask whether to overwrite or not, if yes then overwrite it or else give a message file exists already and not overwritten.

->crtfl ()

{

if [ -f $1 ]

then echo file exist

elif [ -d $1 ]

then echo its a directory

else touch $1

fi

}

crtdr ()

{

if [ -d $1 ]

then echo its exist

else mkdir $1

fi

}

cpy ()

{

if [ -r $1 ]

then

echo $1 have read permission

else

echo no read permission

fi

if [ -f $1 -a -d $2 ]

then

if find $2 -name $1

then

echo Do u want to overwrite y/n

read w

if [ $w = "y" ]

then cp $1 $2

echo file is overwritten

else

echo Not interested

fi

else

cp $1 $2

fi

elif [ -f $1 -o -d $2 -a -f $1 ]

then

if [ -d $2 ]

then cp $1 $2

else mkdir $2

cp $1 $2

fi

else echo file or dir not exist

fi

}

mv ()

{

if [ -f $1 -a -d $2 ]

then

if find $1 `ls $2`

then

echo Do u want to Replace y/n

read w

if [ $w = "y" ]

then mv $1 $2

else

echo Not interested

fi

else

mv $1 $2

fi

elif [ -f $1 -o -d $2 -a -f $1 ]

then

if [ -d $2 ]

then mv $1 $2

else mkdir $2

mv $1 $2

fi

else echo Both does not exist

fi

}

ch=0

while [ $ch -ne 5 ]

do

echo Enter options :

echo 1. Create file

echo 2. create Directory

echo 3. copy file

echo 4. move file

echo 5. exit 0

read ch

case $ch in

1) echo enter the file name

read fn

crtfl $fn ;;

2) echo enter the dirr name

read dir

crtdr $dir ;;

3) echo enter file name

read fn

echo enter the destination or path

read path

cpy $fn $path ;;

4) echo enter file name

read fn

echo enter the destination or path

read path

mv $fn $path ;;

\*) exit 0

esac

done

12: Write a function yesno() to display question to user and accept answer as y/n. If answer to the question is y the function should return 0 otherwise 1.

Use yesno functions for asking different questions. Question will be passed as parameter to the function.

Accept filename from user check whether it is file or directory. Use yesno() function to display question do you really want to delete file? If the ans is y, then delete the file or directory.

->yesno(){

if [ $1 = "y" ]

then

return 0

else

return 1

fi

}

echo enter the file:

read fn

if [ -f $fn ]

then

echo It is a file

echo "do you want to delete the file(y/n):"

read opi

if yesno $opi

then

rm $fn

echo file deleted

else

echo you chose not to delete the file

fi

else

echo file not found

exit 0

fi

if [ -d $fn ]

then

echo it is a directory

exit 0

fi

13: Write a shell script to store names of four employees and check whether those employees are currently logged in or not. Display appropriate message.

->log(){

user=who|tail -1|cut -d" " -f1

if [ $1 = `who|tail -1|cut -d" " -f1` ]

then

echo $1 logged in

else

echo $1 not logged in

fi

}

echo enter the names of four employees:

read e1

read e2

read e3

read e4

log $e1

log $e2

log $e3

log $e4

14: Accept the user's first and last name and the echo the entire name along with some suitable comment.

->echo enter the first name

read n1

echo enter the last name

read n2

echo "The full name of yours is $n1 $n2"

15: List all files that have been modified today.

->find \* -type f -mtime 1 -ls

16: Display long listing of only the regular files in the current directory.

->for i in \*.\*

do

if [ -f $i ]

then

echo `ls -l $i`

fi

done

17: Display details of all files in the 2 “paths” accepted from user. The display should be screen by screen.

->echo "enter the path 1"

read dir1

echo "enter the path 2"

read dir2

ls -l $dir1 > sev.txt

ls -l $dir2 >> sev.txt

more sev.txt

18: Let the script display its name and its PID.

->echo $0

echo "the pid is `ps ax| grep $0 |head -1|cut -b3,4,5`"

19: Get the concatenated o/p of 2 files into a third file: Take 3 command line arguments:

The first argument is the name of a destination file, and the other two arguments are names of files whose contents are to be placed in the destination file.

->cat $2 $3 > $1

20: Write a menu driven shell program to:

a.Display calendar of current month

b.Search for a pattern in all the files/subdirectories from current directory.

c.Count the no. of directories / sub directories in current directory

->echo "menu

a. display calender of current month

b. search for a pattren

c. count the no of direc"

read opt

case $opt in

a) cal ;;

b) echo "enter the pattern"

read p

ls -R | grep $p ;;

c) find \*/ -type d|wc -l ;;

esac

21: Display day of week for a given date. (ddmmyyyy)

If day is Monday, display message “Monday Blues” Friday display message “yeh! It’s week end.”

Similarly display different messages for each day of the week.

->echo "enter the date in format (ddmmyyyy):"

read d

day=`echo $d|cut -c1-2`

month=`echo $d|cut -c3-4`

year=`echo $d|cut -c5-8`

dy=`date -d $year-$month-$day +%A`

echo $dy

if [ $dy = "Monday" ]

then

echo "Monday Blues"

elif [ $dy = "Tuesday" ]

then

echo "Tuesday Tulips"

elif [ $dy = "Wednesday" ]

then

echo "Wednesday Vibes"

elif [ $dy = "Thursday" ]

then

echo "Thursday Thunders"

elif [ $dy = "Friday" ]

then

echo "Friday one day to weekend"

elif [ $dy = "Saturday" ]

then

echo "Saturday night fever" ]

else

echo "Yay!! Holiday"

fi

22: Display the contents of all .lst files in the current directory.

->ls \*.lst

23: Design a simple calculator, which will add/subtract/multiply/divide 2 numbers. eg. cal 10 20 + will give o/p as 30.

->if [ $3 = "+" ]

then

echo `expr $1 + $2 `

elif [ $3 = "-" ]

then

echo `expr $1 - $2 `

elif [ $3 = "x" ]

then

echo `expr $1 \\* $2 `

else

echo `expr $1 / $2 `

fi

24: For a student file with the following fields, rollno, name, marks, Generate 2 files ‘Pass’ and ‘Fail’ containing records of student who have passed or failed. Also count the number of students who have passed or failed.

->

25: Accept a date string from terminal and display employees born after the input date.

->