**DAY 3 ASSIGNMENT**

Question 1 :- Get user info from */*etc*/*passwd and change ownership of user's home directory (select userid higher than 1000)

Answer :-

#!/bin/bash -x

for id in `cat /etc/passwd | awk -F: '{print $3}'`

do

if [ $id -gt 1000 ]

then

path=`cat /etc/passwd | grep $id | awk -F: '{print $6}'`

chown $id $path

fi

done

Question 2 :- Move files from one folder to the respective folders.

Answer :-

#!/bin/bash -x

for file in `ls \*.txt`

do

foldername=`echo $file | awk -F . '{print $1}'`

if [ -d $foldername ]

then

echo $foldername already exists, hence file not copied.

else

mkdir $foldername && cp $file $foldername

echo $file copied to $foldername

fi

done

Question 3 :- Append current date to all log files name which has extension .log. 1 from a folder

Answer :-

#!/bin/bash -x

for file in `ls \*.log.1`

do

filename=`echo $file | awk -F. '{print $1}'`

date=`date -I | awk -F- '{print $3$2$1}'`

filename=`echo $filename-$date.log`

mv $file $filename

done

Question 4 :- Archive the files from */*var*/*log folder which have modified 7 days ago and move it to your backup folder

Answer :-

#!/bin/bash -x

for file in /var/log/\*

do

diff=`echo $(($(($(date +%s) - $(date +%s -r $file)))/86400))`

if [ $diff -ge 7 ]

then

cp -r $file backup/

fi

done

Question 5 :- Check if a folder exists or not. If it's not present, create it

Answer :-

#!/bin/bash

read -p "Enter a name of the folder you want to make: " folder

if [ -d $folder ]

then

echo Folder exists already

else

mkdir $folder

echo $folder created

fi

Question 6 :- Execute command "hello" and "Is" and check its execution status and print whether command executed successfully or not.

Answer :-

#!/bin/bash -x

path=`whereis hello | awk '{print $2}'`

echo -e "\nExecuting hello....."

if [ "$path" == "" ]

then

echo Execution failed as hello is not found

else

hello

fi

path=`whereis ls | awk '{print $2}'`

echo -e "\nExecuting ls....."

if [ "$path" == "" ]

then

echo Execution failed as ls is not found

else

ls

fi

Question 7 :- Set environment usersecret="dH34xJaa23" if its already not set

Answer :-

#!/bin/bash -x

# Needs to be run using "source ./myscript.sh" to run it in the current

# shell as running it normally using ./ opens a new bash subshell and

# the environment variable gets assigned in the child shell

if [ "`echo $usersecret`" != "" ]

then

echo "usersecret already set"

else

export usersecret=dH34xJaa23

fi

Question 8 :- Find a word "systemd" from all log files in the folder */v*ar*/*log and print number of occurrence more than 0 against each file.

Answer :-

#!/bin/bash

for file in /var/log/\*

do

number=`grep systemd $file | awk '{print NR}' | tail -1`

if [ "$number" == "" ]

then

number=0

fi

echo ${file##\*/} : $number

done

Question 9 :- Create process list table displays process id, parent process id, command name, % of memory consumption, % of cpu utilization

Answer :- ps -e -o pid,ppid,cmd,%mem,%cpu | less

Question 10 :- Print last 4 frequently access urls count in sorted order from */v*ar*/*log/httpd*/*access.log

Answer :- cat access.log | sort | tail -4

Question 11 :- Print list of last 4 frequently access unique urls at particular hours from */var/*log/httpd*/*access.log

Answer :- cat access.log | awk -F: '{if($2==12) print $0}' | tail -4

Question 12 :- Print list of web response code count in the unique sorted order at specific hours

Answer :- cat access.log | sort -u | awk -F: '{if($2==12) print $0}' | awk -F"\"" '{print $3}' | tail -4

Question 13 :- Print list of last 10 unique sorted client IP from */var/*log/httpd*/a*ccess.log

Answer :- cat access.log | sort -u | awk '{print $1}' | tail -10

Question 14 :- Data analysis */* manipulation (Awk)

Answer :-

i) Print EmployeeName and TotalPay who has BasePay greater than 10000

cat data.csv | awk '{if($4>10000) print $2, $7}'

ii) What is the aggregate TotalPay of employees whose jobtitle is 'CAPTAIN'

cat data.csv | grep -i captain | awk '{sum+=$7} END {print sum}'

iii) Print JobTitle and Overtimepay who has Overtimepay is between 7000

cat data.csv | awk '{if($5>7000 && $5<10000) print $3, $5}'

iv) Print average BasePay

cat data.csv | awk '{sum+=$4} END {print sum/(NR-1)}'

Question 15 :- Find the difference between original file and the updated file. Apply changes to the original file.

Answer :-

#!/bin/bash +x

original=`cat original/original-file.sh`

updated=`cat updated/updated-file.sh`

if [ "$original" != "$updated" ]

then

cp -r original original-backup/

cp updated/updated-file.sh original/original-file.sh

else

echo Already Updated.

fi