**Quiz 5**

All Shell scripts should display usage function, they should also validate input arguments to be correct, use functions and recursion as much as possible.

1. Write a shell script to find the number of lines in a list of files using sed.

You should write a for loop to go through all files in a directory and then count the number of lines in each file, display it as:

./script <Full path to directory>

File1 has 45 lines

File2 has 20 lines

2 Files in total, 65 lines in total

A)#/bin/bash

echo " please enter the directory name to count the number of lines in a file "

read directoryname

cd $directoryname/

echo " Total number of files in a directory $(find . -maxdepth 1 -type f -printf . | wc -c) "

find $directoryname | xargs wc -l

2. Write a shell script to substitute one pattern for another in a text file.

./script.sh oldpattern newpattern

a)while read a ; do echo ${a//abc/XYZ} ; done < /tmp/file.txt > /tmp/file.txt.t ; mv /tmp/file.txt{.t,}

3. Write a shell script to print complete pathname associated with pid. User has to pass the PID from command line.

./script.sh PID

a)#/bin/bash

echo "Please enter the processor name"

read processname

ps -ef | grep $processname | grep -v grep | awk '{print $2}'

4. Write a shell script to print all users on system using awk.

Hint: Learn what is /etc/passwd file in Linux

a)#!/bin/bash

VAR1=$(cut -d: -f1 /etc/passwd)

echo -e “The users in the system as fallows..n $VAR1 n Completed listing users..!”

echo “Total number of users are $(cat /etc/passwd | wc -l)”

5. Write a shell script to list the frequency of words used in a file.

Hint: Sort and uniq commands will help

a)#/bin/bash

echo " Please enter the file name "

read filename

tr -c '[:alnum:]' '[\n\*]' < $filename | sort | uniq -c | sort -nr | head -10

6. Write a script to take backup of files changed in last 24 hours and archive them.

Hint: Read the Find command tutorial in Linux folder. We typically take backups of a folder by “tar”-ring the entire folders.

a)#/bin/bash

echo " Please enter the directory name "

read directoryname

find $directoryname -mtime -1 -ls

oldfile=$1

newfile=$2

month=`date +%B`

year=`date +%Y`

prefix="frozenskys"

archivefile=$prefix.$month.$year.tar

# Check for existence of a compressed archive matching the naming convention

if [ -e $archivefile.gz ]

then

echo "Archive file $archivefile already exists..."

echo "Adding file '$oldfile' to existing tar archive..."

# Uncompress the archive, because you can't add a file to a

# compressed archive

gunzip $archivefile.gz

# Add the file to the archive

tar --append --file=$archivefile $oldfile

# Recompress the archive

gzip $archivefile

# No existing archive - create a new one and add the file

else

echo "Creating new archive file '$archivefile'..."

tar --create --file=$archivefile $oldfile

gzip $archivefile

fi

7. Write a shell script to determine if a particular service is active or not. For eg: if SSH service is active it should display yes and vice versa. Use netstat , ps commands etc

a)#/bin/bash

echo " Enter the processor name"

read process

systemctl status $process

ps -ef |grep $process

8. Write a shell script to remove spaces from filenames and replace it with underscore

Hint: you can use mv command to re-name files

a)find /tmp/ -depth -name "\* \*" -execdir rename 's/ /\_/g' "{}" \;

9. Write a shell script which prints the df output in more formatted way as below

Filesystem Size Used Avail Capacity Mounted

/dev/sda1 446.71G 18.11G 405.88G 5% /

udev 10M 0 10M 0% /dev

tmpfs 1.14G 9.16M 1.13G 1% /run

a)df -h

10. Write a shell script to summarize available disk space and present in a logical and readable fashion

Ans)#/binbash

free -h | grep -v + > /tmp/ramcache

echo -e '\E[32m'"Ram Usages :" $tecreset

cat /tmp/ramcache | grep -v "Swap"

echo -e '\E[32m'"Swap Usages :" $tecreset

cat /tmp/ramcache | grep -v "Mem"

# Check Disk Usages

df -h| grep 'Filesystem\|/dev/sda\*' > /tmp/diskusage

echo -e '\E[32m'"Disk Usages :" $tecreset

cat /tmp/diskusage

# Check Load Average

loadaverage=$(top -n 1 -b | grep "load average:" | awk '{print $10 $11 $12}')

echo -e '\E[32m'"Load Average :" $tecreset $loadaverage

11. Write a shell function to rename .txt files to .text

Ans. for noteFile in \*\*

do

# Check if is a notes file (even if UPPERCASE or lowercase), and not already edited.

if [[ ${noteFile,,} == \*".text"\* ]] && [[ ${noteFile,,} != \*".txt" ]] && [[ ! -d "$noteFile" ]]

then

# If so, move and rename the file to your save directory.

mv "$noteFile" "$movePath/$noteFile.txt"

fi

done