All Shell scripts should display usage function, they should also validate input arguments to be correct, use fuctions 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

Ans:

#!/bin/bash

# Counting the number of lines in a list of files

# for loop over arguments

if [ $# -lt 1 ]

then

echo "Usage: $0 file ..."

exit 1

fi

echo "$0 counts the lines of code"

l=0

n=0

s=0

for f in $\*

do

l=`wc -l $f | sed 's/^\([0-9]\*\).\*$/\1/'`

echo "$f: $l"

n=$[ $n + 1 ]

s=$[ $s + $l ]

done

echo "$n files in total, with $s lines in total"

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

./script.sh oldpattern newpattern

#!/bin/bash

# Substitute one pattern for another

sed -e '/oldpattern/ { '-e' r newpattern' -e 'd' -e}' -i devel.txt

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

./script.sh PID

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

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

Ans

#!/bin/bash

#Purpose: to list all system users

\_l = "/etc/login.defs"

\_p = "/etc/passwd"

l =$(grep "^UID\_MIN" $\_l)

l1 =$(grep "^UID-MAX" $\_l)

echo "system users"

awk -F ':' -v "min=${1##UID\_MIN}" -v "max=${l1##UID\_MAX}" '{ if ( !($3 >= min && $3 <= max && $7 != "/sbin/nologin")) print $0 }' "$\_p"

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

Hint: Sort and uniq commands will help

#!/bin/sh

#Word Frequency Ascending

cat test.txt |tr -d '[:punct:]'|sed 's/ /\

/g'|tr '[A-Z]' '[a-z]'|sort -n|uniq -c|sort -n

while read count name

do

if [ ${count} -gt 0 ]

then

echo "${name} ${count}"

fi

echo "${name} is unreadable or not exist"

done

exit -1

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.

find /directory\_path -mtime -1 –ls

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

Ans)

Ans: #!/bin/bash

ps aux | grep –i abc

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

find .-type f -name "\* \*.xml"-exec bash -c 'mv "$0" "${0// /\_}"'{} \;

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

#!/bin/sh

# newdf - a friendlier version of df

sedscript="/tmp/newdf.$$"

trap "rm -f $sedscript" EXIT

cat << 'EOF' > $sedscript

function showunit(size)

{ mb = size / 1024; prettymb=(int(mb \* 100)) / 100;

gb = mb / 1024; prettygb=(int(gb \* 100)) / 100;

if ( substr(size,1,1) !~ "[0-9]" ||

substr(size,2,1) !~ "[0-9]" ) { return size }

else if ( mb < 1) { return size "K" }

else if ( gb < 1) { return prettymb "M" }

else { return prettygb "G" }

}

BEGIN {

printf "%-27s %7s %7s %7s %8s %-s\n",

"Filesystem", "Size", "Used", "Avail", "Capacity", "Mounted"

}

!/Filesystem/ {

size=showunit($2);

used=showunit($3);

avail=showunit($4);

printf "%-27s %7s %7s %7s %8s %-s\n",

$1, size, used, avail, $5, $6

}

EOF

df -k | awk -f $sedscript

exit 0

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

#!/bin/sh

# diskspace – Summarizes available disk space and presents it in a logical and readable fashion.

tempfile=”/tmp/available.$$”  
trap “rm -f $tempfile” EXIT

cat << ‘EOF’ > $tempfile  
{ sum += $4 }  
END { mb = sum / 1024  
gb = mb / 1024  
printf “%.0f MB (%.2fGB) of available disk space\n”, mb, gb  
}  
EOF

df -k | awk -f $tempfile

exit 0

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

rename (){

for i in `ls -1`

do

mv $i `echo $i | sed 's/.\*\.txt$/.text/'`

done

}