**2] Move files from one folder to the respective folders.**

**Solution-**

#!/bin/bash -x

for file in `ls \*.txt`

do

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

if [ -d $folderName ]

then

rm -R $folderName;

fi

mkdir $folderName;

cp $file $folderName

done

**Output-**

$ ./moveFileToFolder.sh

++ ls abc.txt

+ for file in `ls \*.txt`

++ echo abc.txt

++ awk -F. '{print $1}'

+ folderName=abc

+ '[' -d abc ']'

+ mkdir abc

+ cp abc.txt abc

**3] append current date to all log file**

**Solution-**

#!/bin/bash -x

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

do

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

newfileName=$fileName"-"$(date '+%d%m%Y')".log"

mv $file $newfileName

done

**Output-**

++ ls abc.log.1

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

++ echo abc.log.1

++ awk -F. '{print $1}'

+ fileName=abc

++ date +%d%m%Y

+ newfileName=abc-19062020.log

+ mv abc.log.1 abc-19062020.log

**4] Archive the files from folder which have modified 7 days ago and move it to your backup folder**

**Solution-**

#!/bin/bash -x

mkdir backUpFolder

oldFiles=`find -atime +7 -type f`

mv $oldFiles backUpFolder

**Output-**

$ ./moveFiles.sh

+ mkdir backUpFolder

++ find -atime +2 -type f

+ oldFiles='./backUpFolder1/assignCurrentDate.sh

./statusCheck.sh'

+ cp ./backUpFolder1/assignCurrentDate.sh ./statusCheck.sh backUpFolder

+ echo ./backUpFolder1/assignCurrentDate.sh ./statusCheck.sh Moved to backup folder

./backUpFolder1/assignCurrentDate.sh ./statusCheck.sh Moved to backup folder

**5] Print last 4 frequently access urls count in sorted order from access.log**

**Solution-**

cat access.log | awk '{print $11}' | sort | uniq -c | sort -r | head -4

**Output-**

1475 "https://fundoopush-dev.bridgelabz.com/login"

1141 "https://fundoopush-dev.bridgelabz.com/dashboard/article"

377 "-"

176 "https://fundoopush-dev.bridgelabz.com/add-post"

**6] Print list of last 4 frequently access unique urls at particular hours**

**Solution-**

cat access.log | awk '{print $4 $11}' | `grep 06:34:27` | sort | uniq -c | sort -r | head -4

**Output-**

5 [20/Sep/2019:06:34:27"https://fundoopush-dev.bridgelabz.com/login"

1 [20/Sep/2019:06:34:27"-"

**7] Print list of web response code count in the unique sorted order at specific hours**

**Solution-**

$ cat access.log | awk '{print $4 $9 $10}' | grep 12:09:03 | sort | uniq -c

**Output-**

1 [30/Sep/2019:12:09:032001710

1 [30/Sep/2019:12:09:0320045365

1 [30/Sep/2019:12:09:0320049526

1 [30/Sep/2019:12:09:0320079309

**8] Print list of last 10 unique sorted client IP from access.log**

**Solution-**

cat access.log | awk '{print $22}' | sort | uniq -c | sort -r | head -10

**Output-**

2348 "114.79.180.62"

474 Gecko)

181

168 Chrome/77.0.3865.90

16 Google

7 Firefox/3.6.8"

7 Chrome/77.0.3865.120

7 "182.48.221.222"

1 Silk/47.1.79

1 Chrome/74.0.3729.131

9**] Check if a folder exists or not. If it’s not present, create it**

**Solution-**

#!/bin/bash -x

dir="xyz"

if [ -d "$dir" ]

then

echo folder already exists

else

mkdir $dir

echo Folder $dir created

fi

**Output-**

$ ./folderExistCheck.sh

+ dir=xyz

+ '[' -d xyz ']'

+ mkdir xyz

+ echo Folder xyz created

Folder xyz created

**10] Find a word "systemd” from all log files in the folder /var/log and print number of occurrence more than 0 against each file.**

**Solution-**

grep -o systemd access.log | wc -l

**11] Set environment usersecret="d H34xJaa23” if its alread y not set**

**Solution-**

#!/bin/bash -x

if [[ ! -v usersecret ]]

then

export usersecret=dH23xJaa23

echo updated

else

echo already present

fi

**output-**

+ [[ ! -v userSecret ]]

+ export userSecret=dH23xJaa23

+ userSecret=dH23xJaa23

+ echo updated

updated

**12] Execute command "hello" and “Is" and check its execution status and print whether command executed successful or not.**

**Solution-**

#!/bin/bash -x

hello

echo $?

ls

echo $?

**Output-**

+ hello

./statusCheck.sh: line 2: hello: command not found

+ echo 127

127

+ ls

abc backUpFolder moveOldFiles.sh statusCheck.sh

abc.txt backUpFolder1 original updated

abc-19062020.log def-19062020.log original-file.sh updated-file.sh

addTwoDice.sh folderExistCheck.sh randomdice.sh

assignCurrentDate.sh moveFileToFolder.sh setEnv.sh

+ echo 0

0

**13] Create process list table displays process id, parent process id, command name, % of memory consumption, % of cpu utilization**

**Solution-**

Ps –elf

**14] Data analysis / manipulation (Awk)**

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

cat data.csv | awk '{if($4>100000){print $2" "$7}}'

**Output-**

EmployeeName TotalPay

NATHANIEL 567595

GARY 538909

ALBERT 335279

PATRICK 326373

DAVID 316285

DAVID 307899

JOANNE 302377

EDWARD 294580

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

cat data.csv | grep CAPTAIN | awk '{print sum+=$4}'

**Output-**

155966

368705

468427

**iiï) Print JobTitle and Overtimepay who has Overtimepay is between 7000 and 10000**

cat data.csv | awk '$5>700 && $5<10000{print $3" "$5}'

**Output-**

DEPUTYCHIEF 9737

ASSTDEPUTY 8601

**iv) Print average BasePay**

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

**Output-**

157972

**15] Find the difference between original file and the updated file. Apply changes to the original file.**

Gauri@DESKTOP8U1QG93MINGW64~/codingClub/terminalCommand/assignments

$ touch original-file.sh updated-file.sh

Gauri@DESKTOP8U1QG93MINGW64~/codingClub/terminalCommand/assignments

$ mkdir original/ updated/

Gauri@DESKTOP8U1QG93MINGW64~/codingClub/terminalCommand/assignments

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

Gauri@DESKTOP-8U1QG93 MINGW64 ~/codingClub/terminalCommand/assignments

$ mkdir backUpFolder/original-backup/

Gauri@DESKTOP-8U1QG93 MINGW64 ~/codingClub/terminalCommand/assignments

$ cp original/original-file.sh backupFolder/original-backup/

Gauri@DESKTOP-8U1QG93 MINGW64 ~/codingClub/terminalCommand/assignments

$ diff -q updated/ backUpFolder/original-backup/

Only in backUpFolder/original-backup/: original-file.sh

Only in updated/: updated-file.sh