**LINUX COMMANDS CODINCLUB ASSIGNMENT-1**

**BY: PRAMOD KUMAR**

**QUES:- Print last 4 frequently access urls count in sorted order from /var/log/httpd/access.log**

1. cat access.log
2. cat access.log | awk '{print$7}'
3. cat access.log | awk '{print$7}' | sort | uniq -c
4. cat access.log | awk '{print$7}' | sort | uniq -c | sort -nr | head -4

**OP:-**

199 /vendor.js

199 /scripts.js

199 /runtime.js

199 /favicon.ico

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

**Checking at hour – 04**

cat access.log | awk '{print$4" "$7}' | awk -F: '{if($2==04) {print$4}}' | awk -F" " '{print$2}' | sort | uniq -c | sort -nr | head -4

**OP:-**

5 /favicon.ico

4 /vendor.js

4 /styles.js

4 /scripts.js

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

**Checking at hour – 06**

cat access.log | awk '{print$4" "$9}' | awk -F: '{if($2==06) {print$4}}' | awk -F" " '{print$2}' | sort | uniq -c | sort -nr | head -4

**OP:-**

368 200

4 206

## There are only two codes at hour 06.

**QUES:- Print list of last 10 unique sorted client IP from access.log**

1. less access.log
2. awk '{print$1}' access.log
3. awk '{print$1}' access.log | sort | uniq -c
4. awk '{print$1}' access.log | sort | uniq -c | sort -nr | head -4

**OP:-**

383 10.56.2.2

345 10.56.19.3

323 10.56.21.2

285 10.56.6.4

**Last 10 unique IP’s:-**

awk '{print$1}' access.log | sort | uniq -c | tail -10

**OP-**

323 10.56.21.2

251 10.56.22.3

214 10.56.3.4

9 10.56.34.4

242 10.56.4.2

7 10.56.44.4

40 10.56.46.2

168 10.56.5.2

285 10.56.6.4

186 10.56.9.3

**QUES:- Data Analysis / Manipulation**

**1.Print Emp name and TotalPay who has BasePay > 10000**

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

**OP-**

EmployeeName : TotalPay

NATHANIEL : 567595

GARY : 538909

ALBERT : 335279

CHRISTOPHER : 332343

PATRICK : 326373

DAVID : 316285

ALSON : 315981

DAVID : 307899

JOANNE : 302377

PATRICIA : 297608

EDWARD : 294580

**2.Aggregate Total Pay whose jobtitle is “CAPTAIN”:**

cat data.csv | awk '{if($3=="CAPTAIN") sum+=$7} END {print "Aggregate = ",sum}'

**OP-**

Aggregate = 1171796

**3.Print jobtitle and overtime pay who has overtime pay b/w 7000 and 10000 :**

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

**OP-**

DEPUTYCHIEF 9737

ASSTDEPUTY 8601

**4.Print avg BasePay:**

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

**OP-**

Average = 172333

**QUES:- Find the difference bw original and updated file.Apply changes to the original file**

mkdir original && cd original && touch original-file.sh

mkdir updated && cd updated && touch updated-file.sh

diff -ruN original updated

diff -ruN original updated > patchfile.patch

cp -r original original-backup

patch -d original-backup/ < patchfile.patch

**SHELL SCRIPT QUESTIONS**

**Ques: Get user info from /etc/passwd and change ownership of user’s home directory**

#! /bin/bash

data=$(cat /etc/passwd | awk -F: '{if($4>1000) {print$6":"$7}}')

for i in $data

do

user=$(echo $i | cut -d: -f1)

dir=$(echo $i | cut -d: -f2)

chown : $dir

echo $user

ls -l $dir

done

**OP:-**

sync

-rwxr-xr-x 1 root root 35488 leq 7 2019 /bin/sync

nobody

-rwxr-xr-x 1 root root 14488 qad 16 2019 /usr/sbin/nologin

\_apt

-rwxr-xr-x 1 root root 14488 qad 16 2019 /usr/sbin/nologin

rwhod

-rwxr-xr-x 1 root root 14488 qad 16 2019 /usr/sbin/nologin

iodine

-rwxr-xr-x 1 root root 14488 qad 16 2019 /usr/sbin/nologin

**QUES:- Move files from one folder to the respective folders.**

a. touch abc.txt def.txt ghi.txt

b. ls

**SHELL CODE:-**

#! /bin/bash

for filename in `ls \*.txt`

do

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

if [ -d $foldername ]

then

rm -r $foldername

fi

mkdir $foldername

cp $filename $foldername

echo $filename "copied to" $foldername

done

**QUES:-Append current date to all log files**

#! /bin/bash

for file in `ls \*log.1`

do

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

ext=`echo $file | awk -F. '{print$2}'`

DATE=`date +%d%m%Y`

newfile=$filename-$DATE.$ext

mv $file $newfile

echo $newfile

done

**OP:-**

a-04072020.log b-04072020.log c-04072020.log

abc-04072020.log bcd-04072020.log

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

#! /bin/bash

find . -mtime -7 | awk -F/ '{print$2}' | xargs tar cf day7.rar

if [ -d "backup" ]

then

mv day7.rar backup/

else

mkdir backup

mv day7.rar backup/

fi

**QUES: Check if a folder exists or not : -**

#! /bin/bash

read -p "Enter folder name: " dir

echo $dir

for folder in `ls`

do

if [ ! -d $dir ]

then

mkdir $dir

break

else

echo "FOLDER ALREADY EXISTS"

break

fi

done

**Ques:Execute command “hello” and “ls” and check its execution status and print if its successful or not.**

#! /bin/bash

status=`ls`

if [ $? -eq 0 ]

then

echo "Command ls Executed Successfully"

else

echo "Command ls Execution Failed"

fi

status=`hello`

if [ $? -eq 0 ]

then

echo "Command hello Executed Successfully"

else

echo "Command hello Execution Failed"

fi

**OP:-**

Command ls Executed Successfully

./cmdstatus.sh: line 11: hello: command not found

Command hello Execution Failed

**Ques:Set Environment usersecret=”dH34xJaa23” if its already not set**

#! /bin/bash

var1=`printenv usersecret`

len=`echo ${#var1}`

if [ $len -gt 0 ]

then

echo "UserSecret Exists"

else

export usersecret="dH34xJaa23"

echo "UserSecret has been set"

echo `printenv usersecret`

fi

**Ques: Find a word “systemd” from all log files and print no. of occurrence more than 0 against each file.**

#! /bin/bash

for file in `ls \*.log`

do

count=`grep -c "systemd" $file`

if [ $count -gt 0 ]

then

echo "No. of occurrence of systemd in "$file "is "$count

fi

done

**OP:-**

No. of occurrence of systemd in a.log is 3

No. of occurrence of systemd in b.log is 1

**Ques:Create process list table display pid,ppid,cmd,%mem,%cpu**

#! /bin/bash

printf "%-10s%-15s%-15s%-15s%s\n" "PID" "PPID" "CMD" "%MEM" "%CPU"

data=$(ps -o pid,ppid,comm,pmem,pcpu | grep -v PID | awk '/[0-9]\*/{print$1":"$2":"$3":"$4":"$5}')

for i in $data

do

PID=$(echo $i | cut -d: -f1)

PD=$(echo $i |cut -d: -f2)

CMD=$(echo $i | cut -d: -f3)

MEM=$(echo $i | cut -d: -f4)

CPU=$(echo $i | cut -d: -f5)

printf "%-10s%-15s%-15s%-15s%s\n" "$PID" "$PD" "$CMD" "$MEM" "$CPU"

done

**OP:-**

PID PPID CMD %MEM %CPU

2769 2762 bash 0.1 0.2

2791 2769 pid.sh 0.0 0.0

2792 2791 pid.sh 0.0 0.0

2793 2792 ps 0.0 0.0

2794 2792 grep 0.0 0.0

2795 2792 awk 0.0 0.0