NAME: CHELUVESHA. B

CODINCAMP BOOTCAMP

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

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

a) Vi*ew /*etc*/*passwd file b) Print the 1st field from /etc*/*passwd file c) Print all userids > 1000 d) Print the 2nd field to get home directory e) Use command substitution to get user list and home directory f) Change ownership of above home directory with user which is retrieve above g) Iterate above steps for all userid > 1000

a) View /var*/*log/httpd/access.log b) Print field which has urls data. c) Sort extracted urls and count it d) Print 4 unique urls

Expect sample output -

3458 /index.html

300 /api/swagger-ui.html 100 /favi.ico

20 /robots.txt

E.g */*etc/passwd =>

ashishv:x:1002:1002::/home*/*ashishv:/bin/bash extract values mark in bold - user and home directory Expected output -

/home/ashishv: drwx------ 6 ashishvashishy 4096 Aug 6 12:48 ashishv

#! /bin/bash

for file in `ls passwd`

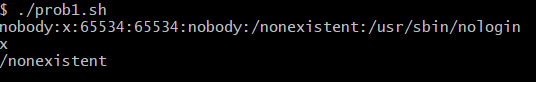
do

cat passwd | awk -F: '{if($3>1000) print $0}'

cat passwd | awk -F: '{if($3>1000) print $0}' | awk -F: '{print $2}'

cat passwd | awk -F: '{if($3>1000) print $0}' | awk -F: '{print $6}'

done



1. Move files from one folder to the respective folders.

 E.g current folder have files abc.txt, def.txt, ghi.txt, jkl.txt

You have to move these files to the folder like abc.txt => abc/ , def.txt => def/ .*..* Expected outcome -

abc/abc.txt def/def.txt ghi*/*ghi.txt jkl/jkl.txt

#! /bin/bash

for filename in `ls \*.txt`

do

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

if [ -d $foldername ]

then

rm -r $foldername

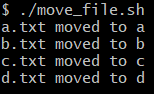
fi

mkdir $foldername

mv $filename $foldername

echo $filename "moved to" $foldername

done



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

Expected sample output -

E.g. original file - access.log.1

New updated file name - access-20102019.log

#! /bin/bash

for filename in `ls \*.log`

do

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

echo $foldername"-"$(date +"%Y%m%d")

done

$ ./logo.sh

access-20200629

4) Check if a folder exists or not. If it's not present, create it

#! /bin/bash

if [ -f xyz ]

then

echo folder already exist

else

mkdir xyz

echo folder created

fi





5) Execute command "hello" and "Is" and check its execution status and print whether command executed successful or not.

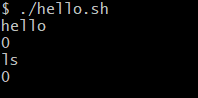
#! /bin/bash

echo hello

echo $?

echo ls

echo $?



6) Create process list table displays process id, parent process id, command name, % of memory consumption, % of cpu utilization

#! /bin/bash

#! /bin/bash

ps -o pid,ppid,cmd,%mem,pcpu

7) Print list of last 10 unique sorted client IP from */var/*log/httpd*/a*ccess.log

a) *V*iew access.log without opening it using editor. b) Print client ip field from access log c) Sort extracted client IP and count it d) Print 4 unique client IPs

Expect sample output -

3635 107.181.177.135

423 27.62.203.44

45 157.4*4*.195.138

4 157.39.158.225

# /bin/bash

cat access.log | awk '{if (NR!=1) {print substr($16, 2, length($16)-2)}}' | sort -nr | head -10

8) Print list of last 4 frequently access unique urls at particular hours from */var/*log/httpd*/*access.log

a) View access.log without opening it using editor. b) Print urls which has given timestamp. c) Sort extracted urls and count it d) Print 4 unique urls

#! /bin/bash

echo -e |cat access.log | awk '{print $4"["$11}' | sort | uniq -c | sort -r |head -4 | awk -F[ '{print $1 $2"--"$3}'

9) Print list of last 4 frequently access unique urls at particular hours from */var/*log/httpd*/*access.log

a) View access.log without opening it using editor. b) Print urls which has given timestamp. c) Sort extracted urls and count it d) Print 4 unique urls

cat access.log | awk -F" " '{print $6" "$7}'| sort | uniq –u| sort | tail -4

1 "HEAD /vb/

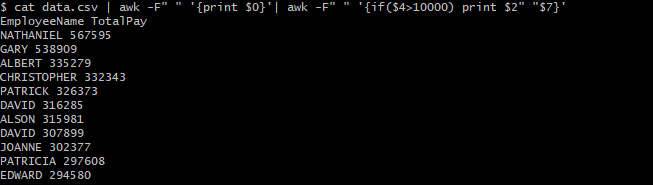
1 "HEAD /vb5/

1 "HEAD /vbulletin/

1 "HEAD /vbulletin5/

**10) Print Employee Name and Total Pay who has Base Pay greater than 10000**

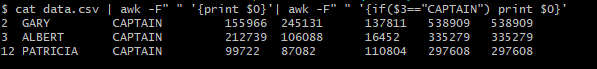
a) Read data file 'data.csv' from command line and extract rows which have Base Pay > 10000 b) Print only Employee Name and Total Pay

****

ii) What is the aggregate Total Pay of employees whose job title is 'CAPTAIN'



1. Read data file 'data.csv' from command line and extract rows which have 'CAPTAIN' in the column 'job title'



1. b) Extract Total Pay and calculate sum. Print the result on terminal



iii) Print Job Title and Overtime pay who has Overtime pay is between 7000 and 10000



iv) Print average Base Pay

****

11) Find the difference between original file and the updated file. Apply changes to the original file.

#! /bin/bash

diff=`diff -iEZ ~/Desktop/TerminalCommands/linux-content/stage1/original/original-file.sh ~/Desktop/TerminalCommands/linux-content/stage1/updated/updated-file.sh`

echo ${#diff}

if [ ${#diff} -gt 0 ]

then

echo "There is Difference in Files"

cp ~/Desktop/TerminalCommands/linux-content/stage1/original/original-file.sh ~/Desktop/TerminalCommands/linux-content/stage1/updated/updated-file.sh

echo "applied changes to UpdatdeFolder"

else

echo "No difference!! "

fi

if [ -d 'original-backup' ]

then

cp ~/Desktop/TerminalCommands/linux-content/stage1/original/original-file.sh ~/Desktop/TerminalCommands/linux-content/stage1/original-backup

else

mkdir original-backup

cp ~/Desktop/TerminalCommands/linux-content/stage1/original/original-file.sh ~/Desktop/TerminalCommands/linux-content/stage1/original-backup

fi

diff2=`diff -iEZ ~/Desktop/TerminalCommands/linux-content/stage1/original-backup/original-file.sh ~/Desktop/TerminalCommands/linux-content/stage1/updated/updated-file.sh`

if [ ${#diff2} == 0 ]

then

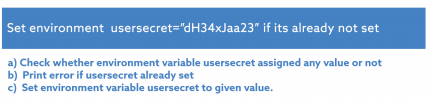
echo "Changes Applied Successfully.... Backup Created!"

else

echo "differ in updated - backup"

fi

**12)**

****

#! /bin/bash

len=`echo $usersecret`

lenu=`echo ${#len}`

if [ ${#len}==0 ]

then

value="dH34xJaa23"

export usersecret=$value

echo "all set"

else

echo "error : env already set"

fi

dH34xJaa23

13) Archive the files from /var/log folder which have modified 7 days ago and move it to your backup folder

#! /bin/bash

destination="/home/varad/Desktop/TerminalCommands/linux-content/stage1/olderbkup"

for file in `find /var/log -type f -mtime +7`

do

cp $file $destination

done

14) Find a word "systemd" from all log files in the folder /var/log and print number of occurence more than 0 against each file.

#! /bin/bash

for log in `ls /var/log/\*.log`

do

count=`grep -c 'systemd' $log`

if [ $count -gt 0 ]

then

echo $count

fi

done