SHIVARAJU K

4shivaraj.gowda@gmail.com

**TERMINAL COMMANDS**

DAY -3 Assignment

1. Get user info from /etc/passwd and change ownership of user's home directory

(select userid higher than 1000)

View /etc/passwd file

Print the 1st field from /etc/passwd file

Print all userids > 1000

Print the 2nd field to get home directory

Use command substitution to get user list and home directory

Change ownership of above home directory with user which is retrieve above

Iterate above steps for all userid > 1000

E.g /etc/passwd =>

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

/home/ashishv:

drwx------ 6 ashishv ashishv 4096 Aug 6 12:48 ashishv

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

shivaraj@LAPTOP-2TP24VFJ:~$ nano 1Problem.sh

shivaraj@LAPTOP-2TP24VFJ:~$ cat 1Problem.sh

#!/bin/bash -x

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

shivaraj@LAPTOP-2TP24VFJ:~$ ./1Problem.sh

++ ls passwd

+ for file in `ls passwd`

+ cat passwd

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

nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin

+ cat passwd

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

+ awk -F: '{ print $2}'

x

+ cat passwd

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

+ awk -F: '{ print $6}'

/nonexistent

2. 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

Create files in current directory or any temporary directory - abc.txt, def.txt, ghi.txt, jkl.txt

Print list of files to move.

Segregate basename and extension of a file.

Create folder using basename.

Move file to newly created folder.

Iterate above steps for all files

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment

$ nano 2Problem.sh

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment

$ cat 2Problem.sh

#!/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

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment

$ chmod u+x 2Problem.sh

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment

$ ./2Problem.sh

++ ls abc.txt def.txt ghi.txt jkl.txt

+ for file in `ls \*.txt`

++ echo abc.txt

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

+ foldername=abc

+ '[' -d abc ']'

+ rm -r abc

+ mkdir abc

+ cp abc.txt abc

+ for file in `ls \*.txt`

++ echo def.txt

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

+ foldername=def

+ '[' -d def ']'

+ rm -r def

+ mkdir def

+ cp def.txt def

+ for file in `ls \*.txt`

++ echo ghi.txt

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

+ foldername=ghi

+ '[' -d ghi ']'

+ rm -r ghi

+ mkdir ghi

+ cp ghi.txt ghi

+ for file in `ls \*.txt`

++ echo jkl.txt

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

+ foldername=jkl

+ '[' -d jkl ']'

+ rm -r jkl

+ mkdir jkl

+ cp jkl.txt jkl

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment

$ ls -p

2Problem.sh abc/ abc.txt def/ def.txt ghi/ ghi.txt jkl/ jkl.txt

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

E.g original file - access.log.1

New updated file name - access-20102019.log

Create files with name abc.log.1, def.log.1 , ghi.log.1, jkl.log.1, mno.log.1

Print list of files to rename.

Segregate basename and extension of a file

Print Date Command to show in ddmmyy

Append Date to the log file name

Iterate above steps for all files which has extension .log.1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/3rd

$ nano 3Problem.sh

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/3rd

$ cat 3Problem.sh

#!/bin/bash -x

for filename in `ls \*.log\*.`

do

foldername=`echo $filename | awk -v var="$(date)" -F. '{ print $1 " " var }'

done

4. Archive the files from /var/log folder which have modified 7 days ago and move it

to your backup folder

Identify files which have modified time greater than 7 days

b)Move these files to the backup folder

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

shivaraj@LAPTOP-2TP24VFJ:~$ nano 4Problem.sh

shivaraj@LAPTOP-2TP24VFJ:~$ cat 4Problem.sh

#!/bin/bash -x

destination="Backup\_Folder/"

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

do

cp $file $destination

done

shivaraj@LAPTOP-2TP24VFJ:~$ ./4Problem.sh

+ destination=Backup\_Folder/

++ find /var/log -type f -mtime +7

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

+ cp /var/log/alternatives.log Backup\_Folder/

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

+ cp /var/log/apt/eipp.log.xz Backup\_Folder/

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

+ cp /var/log/apt/history.log Backup\_Folder/

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

+ cp /var/log/apt/term.log Backup\_Folder/

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

+ cp /var/log/btmp Backup\_Folder/

cp: cannot create regular file 'Backup\_Folder/btmp': Permission denied

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

+ cp /var/log/dpkg.log Backup\_Folder/

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

+ cp /var/log/wtmp Backup\_Folder/

shivaraj@LAPTOP-2TP24VFJ:~$ ls -p Backup\_Folder/

alternatives.log btmp dpkg.log eipp.log.xz history.log term.log wtmp

5. Print last 4 frequently access urls count in sorted order from /var/log/httpd/access.log

View /var/log/httpd/access.log

Print field which has urls data.

Sort extracted urls and count it

Print 4 unique urls

Expect sample output -

3458 /index.html

300 /api/swagger-ui.html

100 /favi.ico

20 /robots.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/5th

$ cat Day3Probs01access.log | awk '{print $11}' | sort | uniq -c | sort -r | head -4 | awk '{print $1 $2}'

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 from /var/log/httpd/access.log

View access.log without opening it using editor.

Print urls which has given timestamp.

Sort extracted urls and count it

Print 4 unique urls

Expect sample output -

3458 /index.html

300 /api/swagger-ui.html

100 /favi.ico

20 /robots.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$ cat Day3Probs01access.log | awk '{print $4"["$11}' | sort | uniq -c | sort -r | head -4 | awk -F[ '{print $1 $2 $3}'

8 01/Oct/2019:05:55:53"https://fundoopush-dev.bridgelabz.com/dashboard/article"

6 30/Sep/2019:09:28:37"https://fundoopush-dev.bridgelabz.com/dashboard/article"

6 30/Sep/2019:06:20:47"https://fundoopush-dev.bridgelabz.com/dashboard/article"

6 30/Sep/2019:06:01:17<https://fundoopush-dev.bridgelabz.com/dashboard/article>

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

View access.log without opening it using editor.

Print web response code field which has given timestamp

Sort extracted response code and count it

Print 4 unique response code count

Expected sample output -

1000 200

100 304

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$ cat Day3Probs01access.log | awk '{print $1}' | sort | uniq -c | sort -r | head -4 | awk '{print $1}'

383

345

323

285

8. Print list of last 10 unique sorted client IP from /var/log/httpd/access.log

View access.log without opening it using editor.

Print client ip field from access log

Sort extracted client IP and count it

Print 4 unique client IPs

Expect sample output -

3635 107.181.177.135

423 27.62.203.44

45 157.44.195.138

4157.39.158.225

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/5th

$ cat Day3Probs01access.log | awk '{if (NR!=1) {print substr($16,2,length($16)-2)}}' | sort -nr |head -10

209.97.150.153

209.17.96.90

209.17.96.250

209.17.96.250

209.17.96.18

209.17.96.18

209.17.96.138

159.65.250.185

86\_64

86\_64

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

a) Test if particular folder exists in current directory or not

b) If its doesn't exists then create it else print "folder already exists.."

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/9th

$ nano 9Problem.sh

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/9th

$ cat 9Problem.sh

#!/bin/bash

if [ -f ABC ]

then

echo "folder already exist"

else

mkdir ABC

echo "folder created"

fi

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/9th

$ ./9Problem.sh

mkdir: cannot create directory ‘ABC’: File exists

folder created

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

a) Execute "hello" command at command prompt

b) Check execution status of "hello" command

c) Execute "Is" command at command prompt

d) Check execution status of "Is" command

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/9th

$ nano 10Problem.sh

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/9th

$ cat 10Problem.sh

#!/bin/bash -x

echo hello

echo $?

echo ls

echo $?

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/9th

$ ./10Problem.sh

+ echo hello

hello

+ echo 0

0

+ echo ls

ls

+ echo 0

0

11. Set environment usersecret="dH34 xJaa23" if its already not set

a) Check whether environment variable usersecret assigned any value or not

b) Print error if usersecret already set

c) Set environment variable usersecret to given value.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/11

$ nano 11Problem.sh

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/11

$ cat 11Problem.sh

#!/bin/bash -x

length=`echo $usersecret`

lengthnumber=`echo ${#length}`

if [ ${#length}==0 ]

then

value="dH34xJaa23"

export usersecret=$value

echo "all set"

else

echo "error : env already set"

fi

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/Day-3/Assignment Final/11

$ ./11Problem.sh

++ echo

+ length=

++ echo 0

+ lengthnumber=0

+ '[' 0==0 ']'

+ value=dH34xJaa23

+ export usersecret=dH34xJaa23

+ usersecret=dH34xJaa23

+ echo 'all set'

all set

12. Find a word "systemd" from all log files in the folder /var/log andprint number of occurrence more than O against each file.

Use linux command to search word and print occurrence

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/5th

$ nano 12Problem.sh

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/5th

$ cat 12Problem.sh

#!/bin/bash -x

for file in `ls \*.log`

do

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

if [ $count -gt 0 ]

then

echo $count

fi

done

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/5th

$ ./12Problem.sh

++ ls Day3Probs01access.log

+ for file in `ls \*.log`

++ grep -c systemd Day3Probs01access.log

+ count=0

+ '[' 0 -gt 0 ']'

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PID | **PPID** | CMD | **%MEM** | %CPU |
| 760 |  | /usr/bin/dockerd -H unix:// | 3.5 | 0.0 |
| 776 |  | /usr/bin/containerd | 0.7 | 0.1 |
| 7266 | 757 | sshd: root@pts/0 | 0.6 | 0.0 |
| 759 |  | /usr/sbin/rsyslogd -n | 0.5 | 0.0 |
| 347 |  | /usr/lib/systemd/systemd-jo | 0.3 | 0.0 |
| 484 |  | /usr/sbin/NetworkManager | 0.3 | 0.0 |
|  | 0 | /usr/lib/systemd/systemd | 0.2 | 0.0 |
| 7268 | 7266 | -bash | 0.2 | 0.0 |
| 758 |  | /usr/bin/python -Es /usr/sb | 0.1 | 0.0 |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

shivaraj@LAPTOP-2TP24VFJ:~$ ps -o pid,ppid,cmd,%mem,pcpu

PID PPID CMD %MEM %CPU

9 8 -bash 0.0 0.3

72 9 ps -o pid,ppid,cmd,%mem,pcp 0.0 0.0

14. Data analysis/ manipulation (Awk)

Id Employee Name Job Title **Base Pay** Overtime Pay Other Pay Total Pay TotalPayBenefits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 NATHANIEL | | GM | 167411 | 0 | 400184 | 567595 | 567595 |
| 2 GARY | | CAPTAIN | 155966 | 245131 | 137811 | 538909 | 538909 |
| 3 ALBERT | | CAPTAIN | 212739 | 106088 | 16452 | 335279 | 335279 |
| 4 | CHRISTOPHER | MECHANIC | 77916 | 56120 | 198306 | 332343 | 32343 |
| 5 PATRICK | | DEPUTY CHIEF | 134401 | 9737 | 182234 | 326373 | 326373 |
| 6 DAVID | | ASST DEPUTY | 118602 | 8601 | 189082 | 316285 | 316285 |
| 7 ALSON | | BATTALION CHIEF | 92492 | 89062 | 134426 | 315981 | 315981 |
| 8 | DAVID | DEPUTY DIRECTOR | 256576 | 0 | 51322 | 307899 | 307899 |
| 10 JOANNE | | CHIEF | 285262 | 0 | 17115 | 302377 | 302377 |
| 12 PATRICIA | | CAPTAIN | 99722 | 87082 | 110804 | 297608 | 297608 |
| 13 EDWARD | | EXECUTIVE | 294580 | 0 | 0 | 294580 | 294S80 |

i) Print EmployeeName and Total Pay who has BasePay greater than 10000

a) Read data file 'data.csv' from command line and extract rows which have BasePay > 10000

b) Print only EmployeeName and TotalPay

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$ cat data.csv | awk -F" " '{print $0}' | awk -F" " '{if($4>10000) print $2" "$7}'

EmployeeName TotalPay

NATHANIEL 567595

GARY 538909

ALBERT 335279

CHRISTOPHER 332343

PATRICK 326373

DAVID 316285

ALSON 315981

DAVID 307899

JOANNE 302377

PATRICIA 297608

EDWARD 294580

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

a) Read data file 'data.csv' from command line and extract rows which have 'CAPTAIN' in the column 'jobtitle'

b) Extract TotalPay and calculate sum. Print the result on terminal.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/CIC-517/linux-content (master)

$ cat data.csv | awk -F" " '{print $0}' | awk -F" " '{if($3=="CAPTAIN") print $0}' | awk '{sum+=$7} END {print sum/3}'

390599

iii) Print JobTitle and Overtimepay who has Overtime pay is between 7000 and 10000

a) Read data file 'data.csv' from command line and extract jobtitle and overtimepay for column value range between 7000-10000

b) Print the result on terminal.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/CIC-517/linux-content (master)

$ cat data.csv | awk -F" " '{print $0}' | awk -F" " '{if($5>7000 && $5<10000) print $3" "$5}'

DEPUTYCHIEF 9737

ASSTDEPUTY 8601

iv) Print average BasePay

a) Read data file 'data.csv' from command line and extract BasePay values and calculate its average

b) Print the result on terminal.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THE BEAST@LAPTOP-2TP24VFJ MINGW64 ~/Desktop/Bridge lab/CIC-517/linux-content (master)

$ cat data.csv | awk -F" " '{print $0}' | awk '{sum+=$4} END {print sum/11}'

172333

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

a) Create two directories as "original' and "updated"

b) Copy given file 'original-file.sh' to the folder "original" and "updated-file.sh" to the folder "updated"

c) Find the difference between these directories using linux command

d) Make copy of folder "original" to some other directory as "original-backup" and apply changes to 'original-file.sh' file

e) Verify that both folders "updated" and "original-backup" have no difference.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*