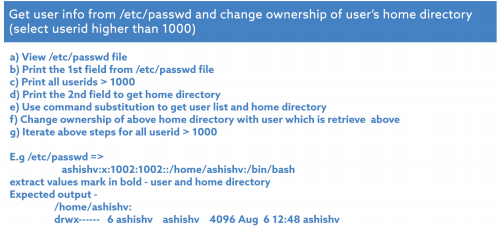
**Assignment 1**

**Left side question**

**Q1.**



nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ cd passwd/

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content/passwd (master)

$ cat etcpasswd.txt

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content/passwd (master)

$ cat etcpasswd.txt | awk'{print $1}'

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content/passwd (master)

$ cat etcpasswd.txt | awk -F: '{if($3 > 200) print $1 " " $3}'

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content/passwd (master)

$ cat etcpasswd.txt | awk -F: '{print $1}'

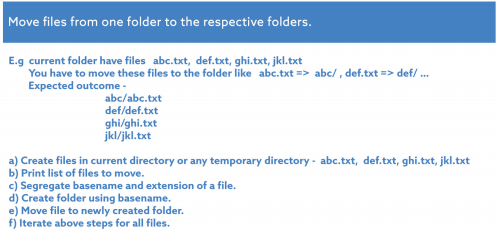
nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content/passwd (master)

$ home\_directory=`cat etcpasswd.txt |awk -F: '{print $6}'`

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content/passwd (master)

$ chown -c -R username home\_directory

Q2.



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 " is copied in " $foldername

done

Output

nida@nida-PC MINGW64 ~

$ ./Q2.sh

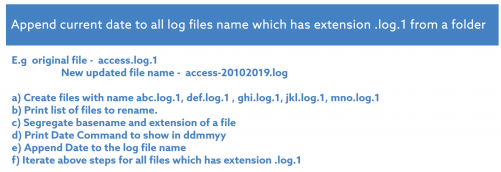
abc.txt is copied in abc

def.txt is copied in def

ghi.txt is copied in ghi

jkl.txt is copied in jkl

Q3.



Code:

#! /bin/bash

n=1

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

do

mv $filename access\_$n-`date "+%d%m%Y"`.log.l

echo $filename "is renamed"

((n=n+1))

done

ls \*.log.l

Output:

nida@nida-PC MINGW64 ~

$ ./Q3.sh

abc.log.l is renamed

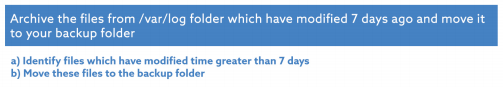
def.log.l is renamed

ghi.log.l is renamed

jkl.log.l is renamed

access\_1-22102020.log.l access\_2-22102020.log.l access\_3-22102020.log.l access\_4-22102020.log.l

Q4



Code:

#! /bin/bash

files=`find -mtime -7 | grep -v git `

tar cf backup.tar [--files-from=$files]

mkdir backup

mv backup.tar backup

Output:

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ cd backup/

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content/backup (master)

$ tar tf backup.tar

./a.txt

./access.log

./copyfile/

./copyfile/a/

./copyfile/a/a.txt

./copyfile/a.txt

./copyfile/b/

./copyfile/b/b.txt

./copyfile/b.txt

./copyfile/c/

./copyfile/c/c.txt

./copyfile/c.txt

./copyfile/copyfile.sh

./copyfile/d.pdf

./copyfile/a/

./copyfile/a/a.txt

./copyfile/a/a.txt

./copyfile/a.txt

./copyfile/b/

./copyfile/b/b.txt

./copyfile/b/b.txt

./copyfile/b.txt

./copyfile/c/

./copyfile/c/c.txt

./copyfile/c/c.txt

./copyfile/c.txt

./copyfile/copyfile.sh

./copyfile/d.pdf

./copyfolder/

./data.csv

./enter/

./foldername/

./linux\_chit\_sheet.pdf

./linux\_problem\_sheet.pdf

./Main.class

./Main.java

./Q4.sh

./Q6.sh

./Q8.sh

./QuickLink/

./QuickLink/language.txt

./QuickLink/linux\_chit\_sheet.pdf

./QuickLink/language.txt

./QuickLink/linux\_chit\_sheet.pdf

./README.md

./Test1/

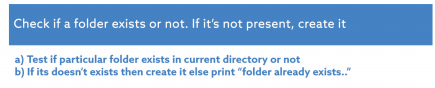
./Test1/language.txt

./Test1/language.txt

./Test2/

./Test3/

Q5.



Code:

#! /bin/bash

read -p "enter foldername to search: " fname

for foldername in `ls -d \*/`

do

exist\_folder=`echo $foldername | awk -F/ '{print $1}'`

done

if [ -d $fname ]

then

echo $fname "already exists"

else

mkdir $fname

echo $fname "folder is created"

fi

Output:

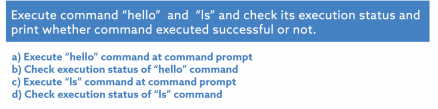
nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ ./Q4.sh

enter foldername to search: Test3

Test3 folder is created

Q6.



Code:

#! /bin/bash

read -p "Enter the command u want to execute: " Command

$Command

echo "returned code $?"

Output:

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ ./Q6.sh

Enter the command u want to execute: ls

a.txt  foldername  'New Text Document.txt'  Test2

access.log linux\_chit\_sheet.pdf Q4.sh    Test3

copyfile linux\_problem\_sheet.pdf Q6.sh    TestFile1.sh

copyfolder Main.class  QuickLink

data.csv Main.java  README.md

enter  mytarfile.tgz  Test1

returned code 0

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ ./Q6.sh

Enter the command u want to execute: hello

./Q6.sh: line 3: hello: command not found

returned code 127

Q7.



Code:

#! /bin/bash

read -p "Enter the var u want to set: " var

var\_pres=`printenv $var`

if [ -d $var\_pres ]

then

echo "The Variable is Not SET"

read -p "Enter the value you want it set to: " value

export $var=$value

echo "The value is Set"

printenv

else

echo $var\_pres

fi

Output:

nida@nida-PC MINGW64 ~

$ ./Q7.sh

Enter the var u want to set: OS

Windows\_NT

nida@nida-PC MINGW64 ~

$ ./Q7.sh

Enter the var u want to set: usersecret

The Variable is Not SET

Enter the value you want it set to: dH34xJaa23

The value is Set

n/vendor\_perl:/usr/bin/core\_perl

HOMEDRIVE=C:

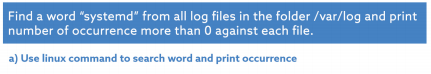
usersecret=dH34xJaa23

PKG\_CONFIG\_PATH=/mingw64/lib/pkgconfig:/mingw64/share/pkgconfig

INFOPATH=/usr/local/info:/usr/share/info:/usr/info:/share/info

HOMEPATH=\Users\nida

Q8.



Code:

#! /bin/bash

read -p "Enter the word u want to search " word

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

do

word\_occ=`cat $filename | sort | uniq -c | grep $word`

if [ -z "$word\_occ" ]

then

echo $word "not found in" $filename

else

echo $word\_occ "found in" $filename

fi

done

Output:

nida@nida-PC MINGW64 ~

$ ./Q8.sh

Enter the word u want to search systemd

1 systemd found in abc.log.l

systemd not found in access\_1-22102020.log.l

systemd not found in access\_2-22102020.log.l

systemd not found in access\_3-22102020.log.l

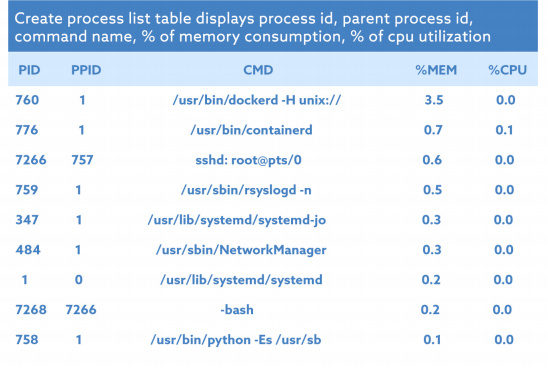
systemd not found in access\_4-22102020.log.l

2 systemd found in def.log.l

systemd not found in hij.log.l

3 systemd found in klm.log.l

**Q9.**



nida@nida-PC MINGW64 ~

$ ps -elf

UID PID PPID TTY STIME COMMAND

nida 2699 2102 pty1 17:56:01 /usr/bin/ps

nida 2102 2101 pty1 14:19:03 /usr/bin/bash

nida 1355 1 pty0 Oct 23 /usr/bin/nano

nida 2101 1 ? 14:19:02 /usr/bin/mintty

**Right Side**

**Q10**



**Code**

**Output**

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

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

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

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

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

28 "https://fundoopush-dev.bridgelabz.com/"

5 "https://fundoopush-dev.bridgelabz.com/dashboard/jobs"

4 "http://fundoopush-dev.bridgelabz.com/wp-login.php"

2 "http://fundoopush-dev.bridgelabz.com/.well-known/acme-challenge/4xM-Y1899BrBIJ76P5Er2sj2VhEtyi\_DFvfo6xiMXXw"

1 "https://fundoopush-dev.bridgelabz.com/dashboard/hashtags/animals"

1 "https://fundoopush-dev.bridgelabz.com/dashboard/archive"

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ cat access.log | awk -F" " '{print $11}' | sort -r | uniq -c | sort -r | head -5

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

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

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

28 https://fundoopush-dev.bridgelabz.com/

**For tail**

**Output:**

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

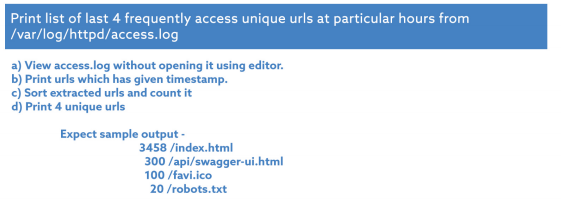
4 "http://fundoopush-dev.bridgelabz.com/wp-login.php"

2 "http://fundoopush-dev.bridgelabz.com/.well-known/acme-challenge/4xM- Y1899BrBIJ76P5Er2sj2VhEtyi\_DFvfo6xiMXXw"

1 "https://fundoopush-dev.bridgelabz.com/dashboard/hashtags/animals"

1 "https://fundoopush-dev.bridgelabz.com/dashboard/archive"

**Q11.**



**Code:**

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ cat access.log | awk -F" " '{print $4 " " $11}' | grep "21/Sep/2019" |awk -F" " '{print $2}' | sort -r | uniq -c

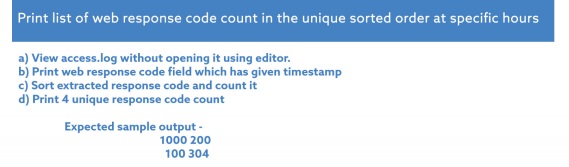
222 "https://fundoopush-dev.bridgelabz.com/login"

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

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

42 "-"

**Q12.**



**Code:**

**Output:**

$ cat access.log | awk -F" " '{print $4 " " $9 " " $10}' | grep "21/Sep/2019" |awk -F" " '{print $2 " " $3}' | sort -r | uniq -c | sort | head -4

1 200 12000023

1 200 1442432

1 200 1514578

1 200 2019077

$ cat access.log | awk -F" " '{print $4 " " $9 " " $10}' | grep "21/Sep/2019" |awk -F" " '{print $2 " " $3}' | sort -r | uniq -c | sort |tail -4

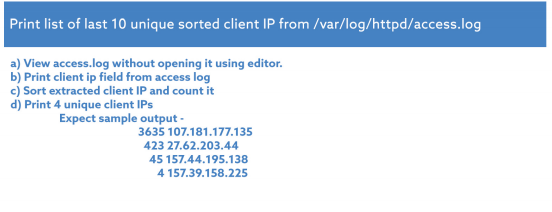
28 200 49526

28 200 79309

29 200 1150

29 200 554

**Q13.**



**Code:**

**Output:**

$ cat access.log | awk -F" " '{print $1}' | sort | uniq -c |sort|head -4

7 10.56.44.4

9 10.56.34.4

40 10.56.46.2

168 10.56.5.2

$ cat access.log | awk -F" " '{print $1}' | sort | uniq -c |sort|tail -4

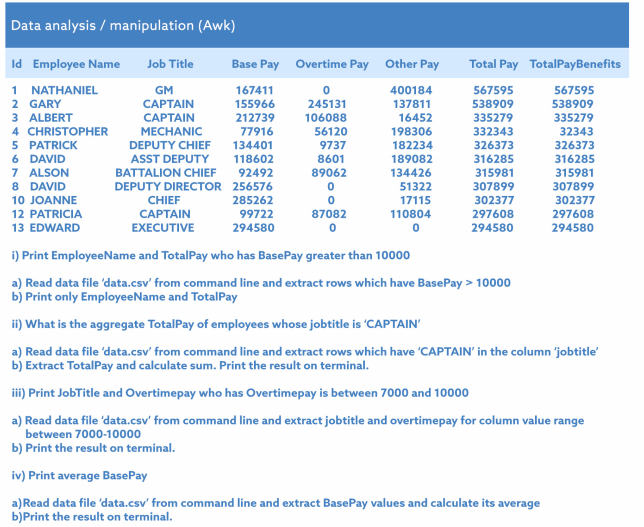
285 10.56.6.4

323 10.56.21.2

345 10.56.19.3

383 10.56.2.2

**Q14.**



i)

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

$ cat data.csv | awk '{if($4>10000) print $0}'

Id EmployeeName JobTitle BasePay OvertimePay OtherPay TotalPay 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 332343

5 PATRICK DEPUTYCHIEF 134401 9737 182234 326373 326373

6 DAVID ASSTDEPUTY 118602 8601 189082 316285 316285

7 ALSON BATTALIONCHIEF 92492 89062 134426 315981 315981

8 DAVID DEPUTYDIRECTOR 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 294580

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

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

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

EmployeeName | TotalPay

NATHANIEL | 567595

GARY | 538909

ALBERT | 335279

CHRISTOPHER | 332343

PATRICK | 326373

DAVID | 316285

ALSON | 315981

DAVID | 307899

JOANNE | 302377

PATRICIA | 297608

EDWARD | 294580

**Print EmployeeName and TotalPay**

$ awk '{print $2" | "$7}' data.csv

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'**

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ awk '{if($3=="CAPTAIN") print $7}' data.csv

538909

335279

297608

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

$ awk '{if($3=="CAPTAIN") print $0}' data.csv

2 GARY CAPTAIN 155966 245131 137811 538909 538909

3 ALBERT CAPTAIN 212739 106088 16452 335279 335279

12 PATRICIA CAPTAIN 99722 87082 110804 297608 297608

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

$ awk '{ sum +=$7 } END {print sum}' data.csv

3935229

iii)

1. **Read data file 'data.csv' from command line and extract jobtitle and overtimepay for column value range between 7000-10000**
2. **Print the result on terminal.**

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

DEPUTYCHIEF|9737

ASSTDEPUTY|8601

iv) **Print average BasePay**

1. **Read data file 'data.csv' from command line and extract BasePay values and**

**calculate its average**

1. **Print the result on terminal.**

$ awk '{ sum += $4; n++ } END { if (n > 0) print sum / n }' data.csv

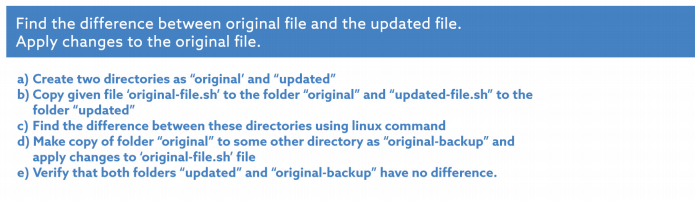
157972

OR

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

157972

**Q15.**



Code:

Output:

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ mkdir original

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ mkdir updated

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ touch original-file.sh

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ touch updated-file.sh

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ mv original-file.sh original

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ mv updated-file.sh updated

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ diff -q original/ updated/

Only in original/: original-file.sh

Only in updated/: updated-file.sh

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ mkdir original\_backup

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ cp original -r original\_backup

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ diff -q original\_backup/ updated/

Only in original\_backup/: original

Only in updated/: updated-file.sh

nida@nida-PC MINGW64 ~/TerminalCommnads/linux-content (master)

$ diff -qr original\_backup/original updated/

Only in original\_backup/original: original-file.sh

Only in updated/: updated-file.sh