**Assignment No.01**

**Name: -** Omprakash Khawshi

**Batch: -** 10 AM To 12 PM

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

**Q1.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

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

b) Print list of files to move.

c) Segregate basename and extension of a file.

d) Create folder using basename.

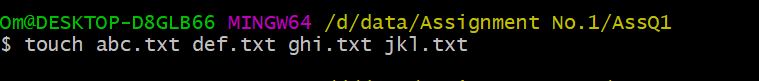
e) Move file to newly created folder.

f) Iterate above steps for all files.

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

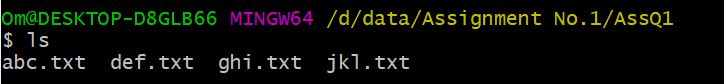
$ touch abc.txt def.txt ghi.txt jkl.txt

****

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

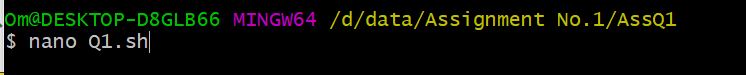
$ ls

abc.txt def.txt ghi.txt jkl.txt

****

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

$ nano Q1.sh

****

for file in `ls \*.txt`

do

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

echo $FolderName

if [ -d $FolderName ]

then

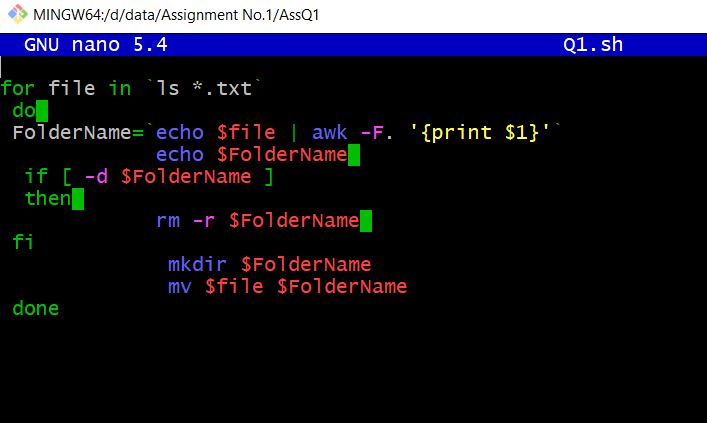
rm -r $FolderName

fi

mkdir $FolderName

mv $file $FolderName

done

****

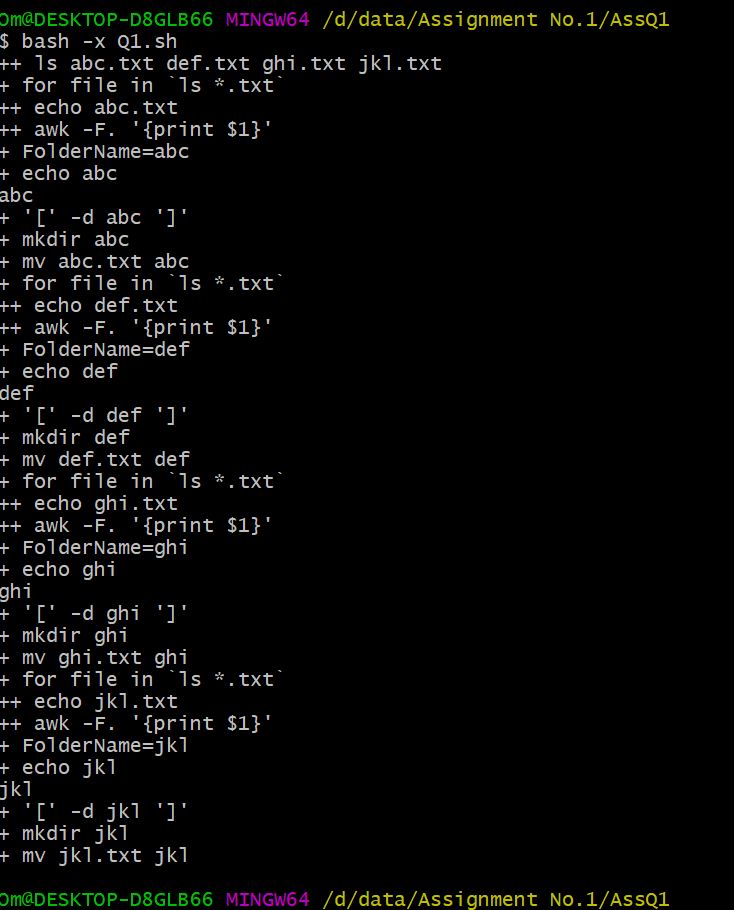
Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

$ bash -x Q1.sh

****

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

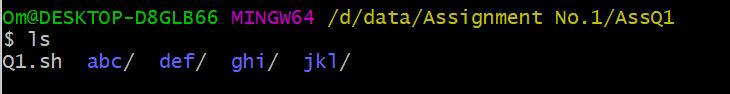
$ bash -x Q1.sh



Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

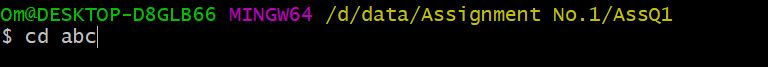
$ ls

Q1.sh abc/ def/ ghi/ jkl/



Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

$ cd abc



Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

$ cd abc

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1/abc

$ ls

abc.txt

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1/abc

$ cd ..

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

$ cd def

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1/def

$ ls

def.txt

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1/def

$ cd ..

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

$ cd ghi

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1/ghi

$ ls

ghi.txt

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1/ghi

$ cd ..

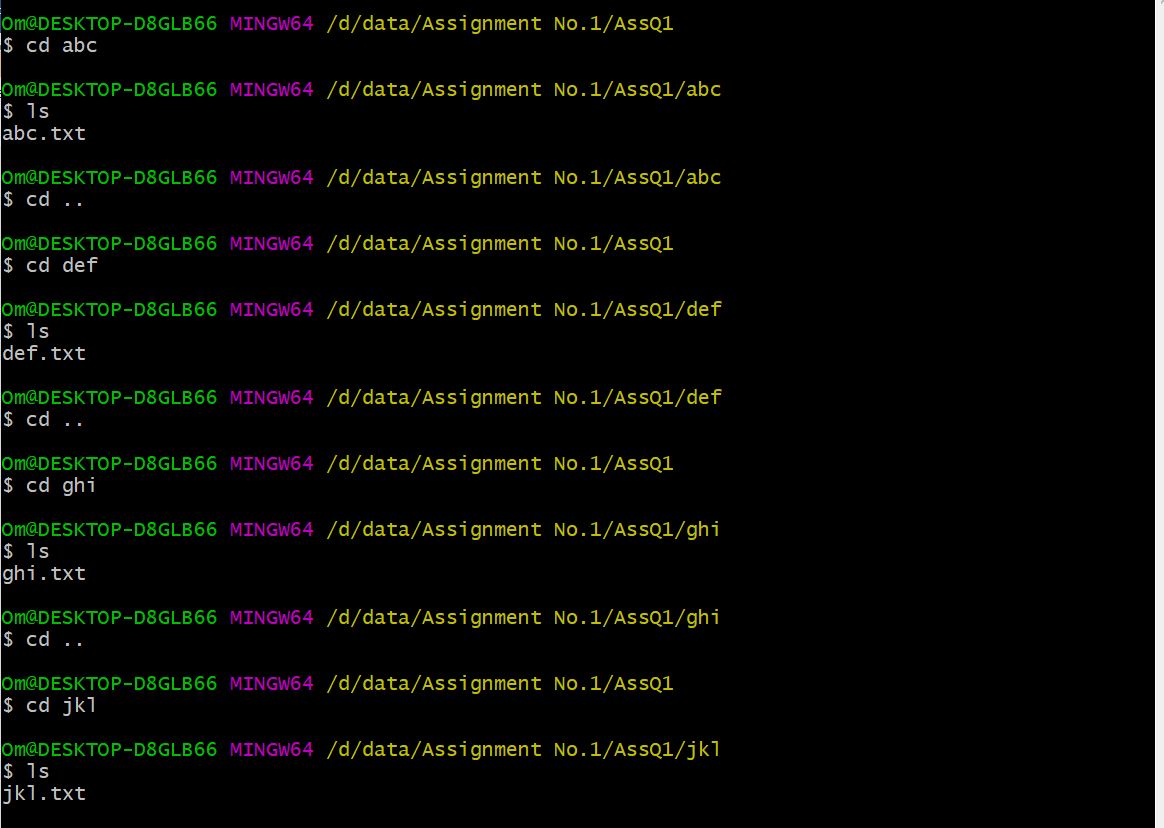
Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1

$ cd jkl

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ1/jkl

$ ls

jkl.txt



Q.2 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

a) Create files with name abc.log.1, def.log.1, ghi.log.1, jkl.log.1, mno.log.1 b) Print list of files to rename.

c) Segregate basename and extension of a file

d) Print Date Command to show in ddmmy

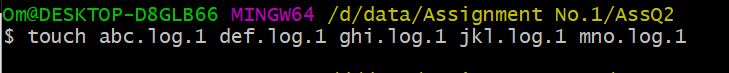
e) Append Date to the log file name

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

**Output:-**

Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ2

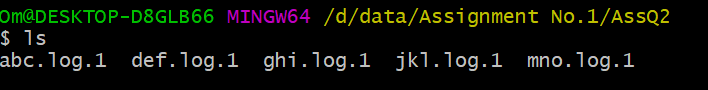
$ touch abc.log.1 def.log.1 ghi.log.1 jkl.log.1 mno.log.1



Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ2

$ ls

abc.log.1 def.log.1 ghi.log.1 jkl.log.1 mno.log.1



Om@DESKTOP-D8GLB66 MINGW64 /d/data/Assignment No.1/AssQ2

$ nano Q2.sh

#!/bin/bash -x

date=$(date +%d%m%y)

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

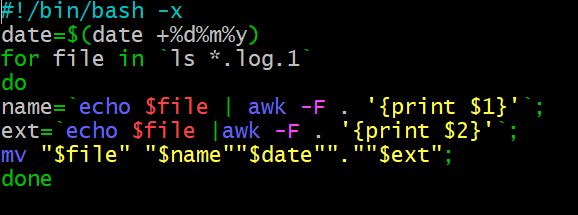
do

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

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

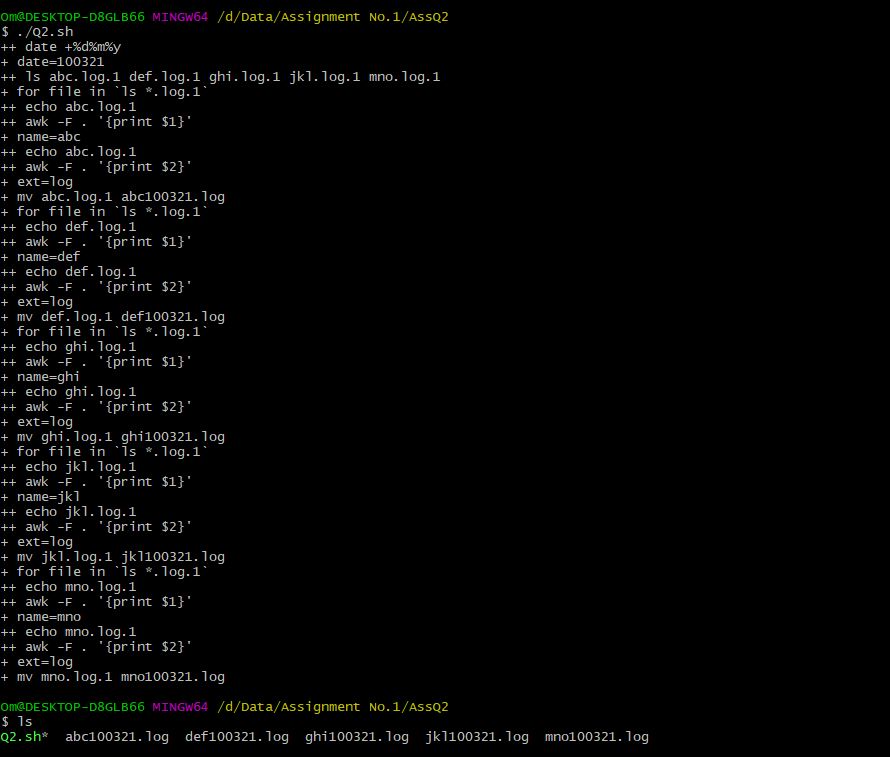
mv "$file" "$name""$date"".""$ext";

done



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ2

$ ./Q2.sh



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ2

$ ls

Q2.sh\* abc100321.log def100321.log ghi100321.log jkl100321.log mno100321.log

Q.3. Archive the files from /var/log folder which have modified 7 days ago and move it to your backup folder

a) Identify files which have modified time greater than 7 days

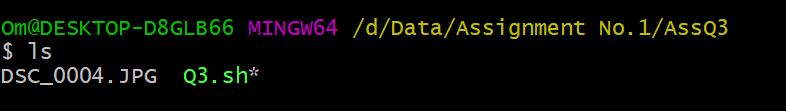
b) Move these files to the backup folder

**Output:-**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ3

$ ls

DSC\_0004.JPG Q3.sh\*



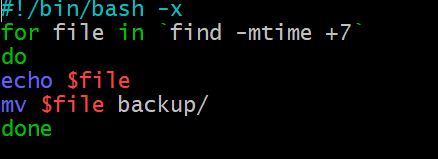
Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ3

$ mkdir Backup



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ3

$ nano Q3.sh



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ3

$ ./Q3.sh

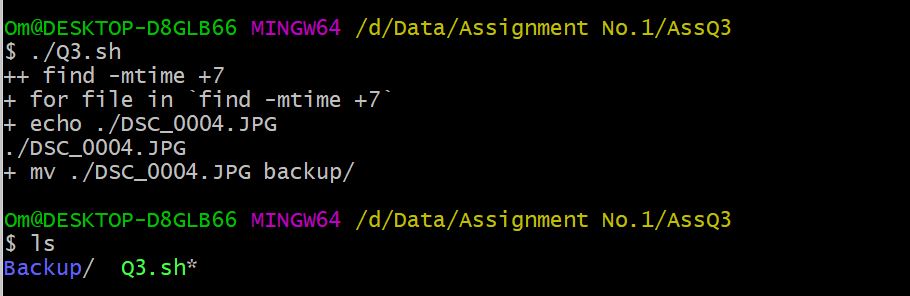
++ find -mtime +7

+ for file in `find -mtime +7`

+ echo ./DSC\_0004.JPG

./DSC\_0004.JPG

+ mv ./DSC\_0004.JPG backup/



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ3

$ ls

Backup/ Q3.sh\*

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ3

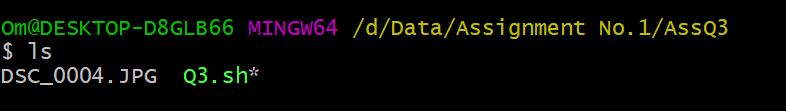
$ cd backup

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ3/backup

$ ls

DSC\_0004.JPG

ls



Q.4 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..."

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ ls

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ nano Q4.sh

#!/bin/bash -x

read -p "Enter Folder Name :-" foldername

if [ -d "${foldername}" ]

then

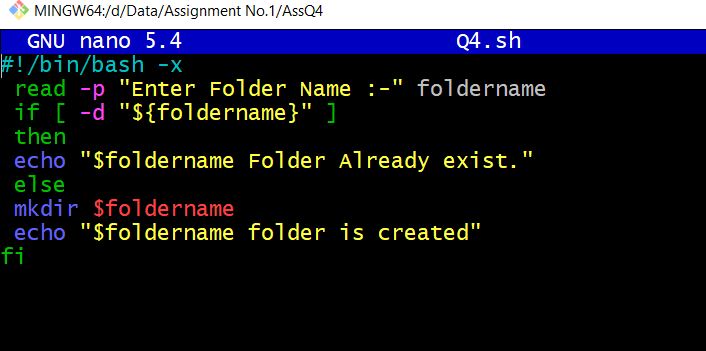
echo "$foldername Folder Already exist."

else

mkdir $foldername

echo "$foldername folder is created"

fi



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ ./Q4.sh

Enter Folder Name :-Omprakash

Omprakash folder is created

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ ./Q4.sh

Enter Folder Name :-Omprakash

Omprakash Folder Already exist.

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ nano Q4.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ ./Q4.sh

+ read -p 'Enter Folder Name :-' foldername

Enter Folder Name :-Om

+ '[' -d Om ']'

+ mkdir Om

+ echo 'Om folder is created'

Om folder is created

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ ./Q4.sh

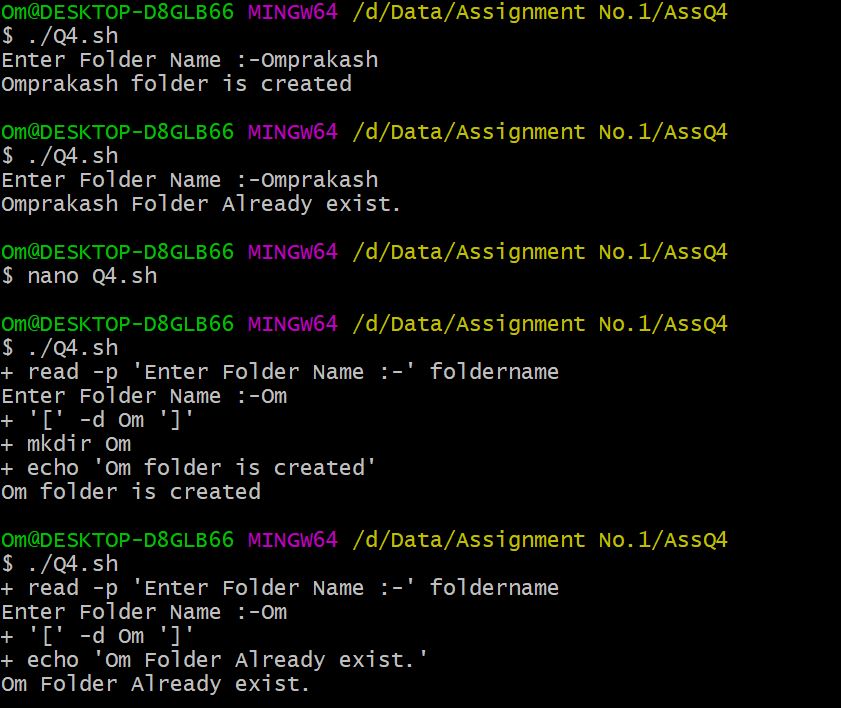
+ read -p 'Enter Folder Name :-' foldername

Enter Folder Name :-Om

+ '[' -d Om ']'

+ echo 'Om Folder Already exist.'

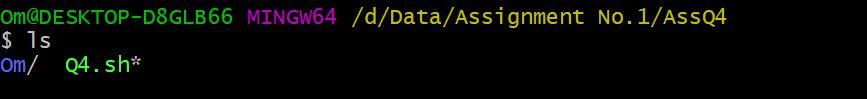
Om Folder Already exist.



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ4

$ ls

Om/ Q4.sh\*



Q.5 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

**Command**: -

**echo $?:-** Execute echo $? command to check the status of executed command as shown below. Here we get exit status as zero which means the “ls” command executed successfully.

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ hello

bash: hello: command not found

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ echo $?

127

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ ls

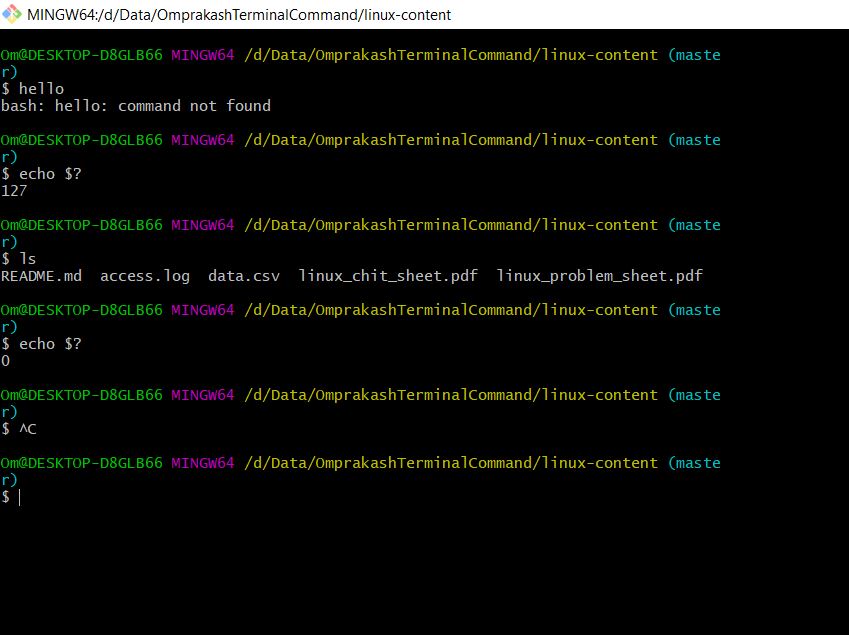
README.md access.log data.csv linux\_chit\_sheet.pdf linux\_problem\_sheet.pdf

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ echo $?

0

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)



**Or**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ5

$ nano Q5.sh

#! /bin/bash

abc=`hello`

if [ $? == 0 ]

then

echo "Command executed successfull"

else

echo -e "hello"

echo "Command failed to execute"

fi

xyz=`ls`

if [ $? == 0 ]

then

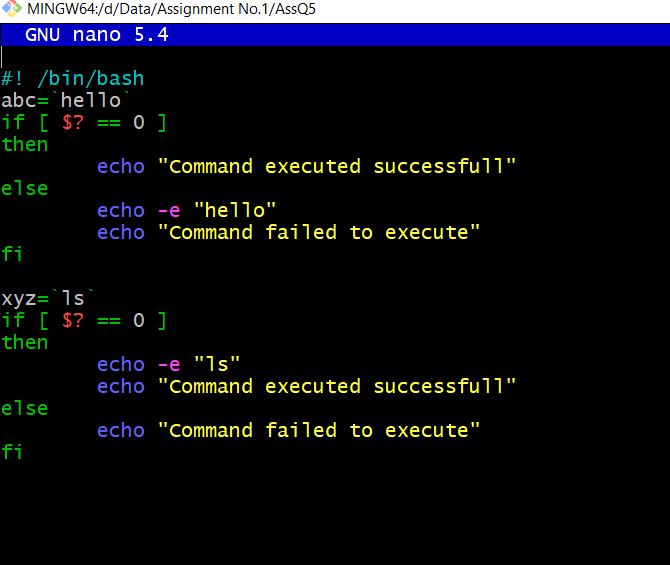
echo -e "ls"

echo "Command executed successfull"

else

echo "Command failed to execute"

fi



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ5

$ ./Q5.sh

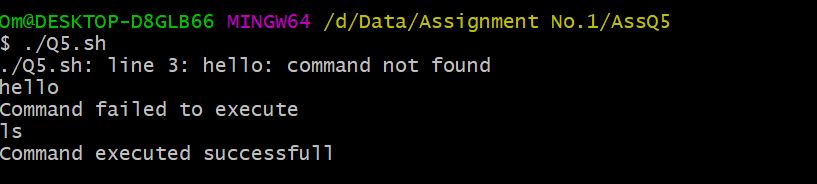
./Q5.sh: line 2: hello: command not found

hello

Command failed to execute

ls

Command executed successful



Q.6. Set environment usersecret="dH34xJaa23" 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.

**Commands: -**

**printenv**

The most used command to displays the environment variables is printenv. If the name of the variable is passed as an argument to the command, only the value of that variable is displayed. If no argument is specified, printenv prints a list of all environment variables, one variable per line.

**export: -**

The **export command** is a built-in utility of **Linux** Bash shell. It is used to ensure the environment variables and functions to be passed to child processes. It does not affect the existing environment variable. Environment variables are set when we open a new shell session.

**Env**:- The command allows you to run another program in a custom environment without modifying the current one. When used without an argument it will print a list of the current environment variables.

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTermr)

$ printenv usersecret

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ export usersecret

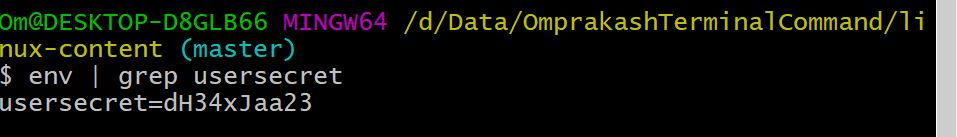
Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ export usersecret=$(echo "dH34xJaa23")

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ env | grep usersecret + Enter button

usersecret=dH34xJaa23



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ env

AppData/Local/Microsoft/WindowsApps:/c/Users/Om/AppData/Local/Programs/Microsoft VS Code/bin:/c/Program Files/Java/jdk-15.0.2/bin:/c/MinGW/bin:/usr/bin/vendor\_perl:/usr/bin/core\_perl

PS1=\[\033]0;$TITLEPREFIX:$PWD\007\]\n\[\033[32m\]\u@\h \[\033[35m\]$MSYSTEM \[\033[33m\]\w\[\033[36m\]`\_\_git\_ps1`\[\033[0m\]\n$

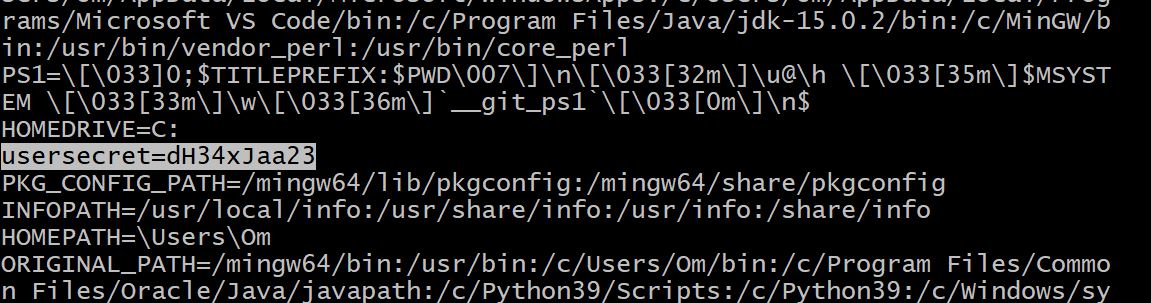
HOMEDRIVE=C:

usersecret=dH34xJaa23

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

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

HOMEPATH=\Users\Om

****

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ export -n usersecret

**Or**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ6

$ nano Q6.sh

#! /bin/bash

len=`echo $usersecret`

lenu=`echo ${#len}`

if [ ${#len}==0 ]

then

value="dH34xJaa23"

export usersecret=$value

echo "env set"

else

echo "error : env already set"

fi



Q.7 Find a word "systemd" from all log files in the folder /var/log and print number of occurrence more than 0 against each file.

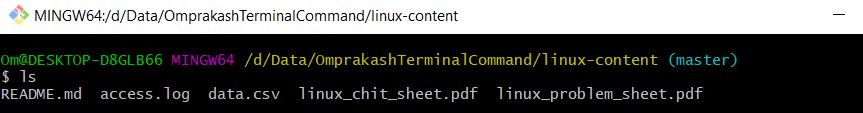
a) Use linux command to search word and print occurrence

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ ls

README.md access.log data.csv linux\_chit\_sheet.pdf linux\_problem\_sheet.pdf



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ grep -c systemd access.log

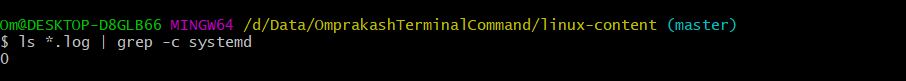
0



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ ls \*.log | grep -c systemd

0



Q.8 Create process list table displays process id, parent process id, command name, % of memory consumption, % of cpu utilization.

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ ps - aux

PID PPID PGID WINPID TTY UID STIME COMMAND

1724 1 1724 11892 ? 197609 01:35:34 /usr/bin/mintty

1767 1725 1767 132 pty0 197609 01:43:33 /usr/bin/ps

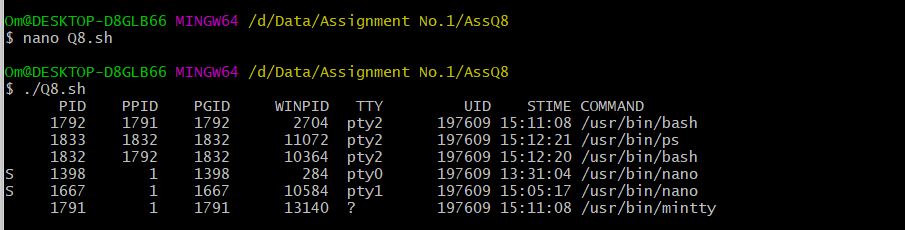
1725 1724 1725 10536 pty0 197609 01:35:34 /usr/bin/bash



**Or**

#! /bin/bash

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



Q.9 Print last 4 frequently access urls count in sorted order from /var/log/httpd/access.log

a) View /var/log/httpd/access.log

b) Print field which has url data.

c) sort extracted urls and count it

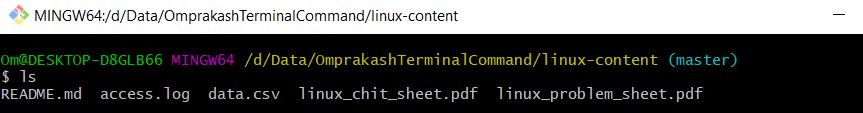
d) Print 4 unique urls

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ ls

README.md access.log data.csv linux\_chit\_sheet.pdf linux\_problem\_sheet.pdf



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

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

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

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

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



**Or**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ9

nano Q9.sh

cat Q9.sh

# /bin/bash

cat access.log | awk '{print $11}' | grep -v '"-"' | sort | uniq -c | sort -nr | head -4

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ9

$ ls

Q9.sh access.log

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ9

$ ./Q9.sh

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/"



Q.10 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

Expect sample output

3458 /index.html

300 /api/swagger-ui.html

100 /favi.ico

20 /robots.txt

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ10

$ nano Q10.sh

#! /bin/bash

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

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ10

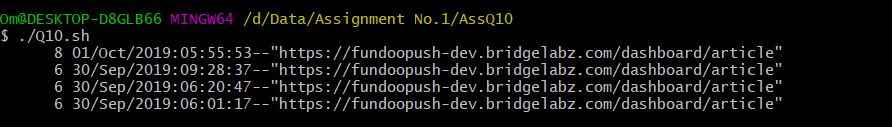
$ ./Q10.sh

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://fundoopushev.bridgelabz.com/dashboard/article"



Q.11 Print list of web response code count in the unique sorted order at specific hours

a) View access.log without opening it using editor.

b) Print web response code field which has given timestamp

c) Sort extracted response code and count it

d) Print 4 unique response code count

**Output**: -

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

$ cat access.log

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

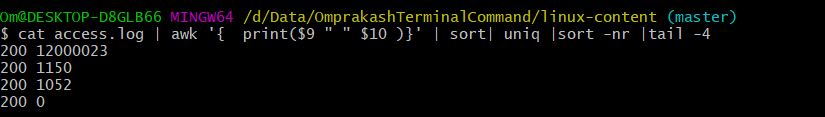
$ cat access.log | awk '{ print($9 " " $10 )}' | sort| uniq |sort -nr |tail -4

200 12000023

200 1150

200 1052

200 0



**Or**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ11

$ nano Q11.sh

# /bin/bash

echo -e |cat access.log | awk '{print $9}' | sort | uniq -c | sort -nr |head -4

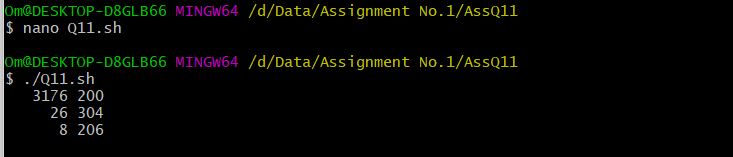
Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ11

$ ./Q11.sh

3176 200

26 304

8 206



Q.12 Print list of last 10 unique sorted client IP from /var/log/httpd/access.log

a) View 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

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

10.56.21.2

10.56.22.3

10.56.3.4

10.56.34.4

10.56.4.2

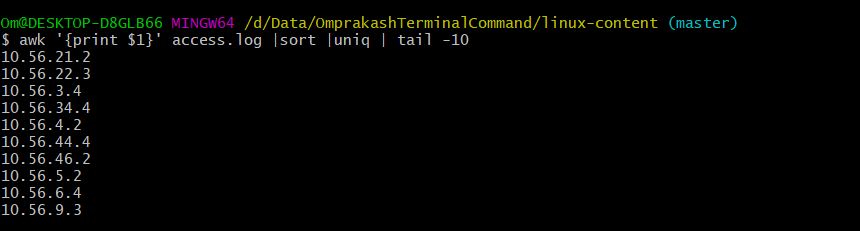
10.56.44.4

10.56.46.2

10.56.5.2

10.56.6.4

10.56.9.3



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

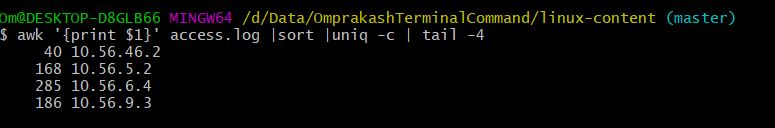
$ awk '{print $1}' access.log |sort |uniq -c | tail -4

40 10.56.46.2

168 10.56.5.2

285 10.56.6.4

186 10.56.9.3



**Or**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ12

$ nano Q12.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ12

$ ./Q12.sh

209.97.150.153

209.17.96.90

209.17.96.250

209.17.96.250

209.17.96.18

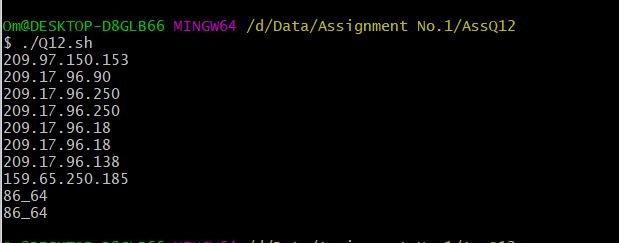
209.17.96.18

209.17.96.138

159.65.250.185

86\_64

86\_64



Q.13 Data analysis / manipulation (Awk)

i) Print Employee Name and TotalPay 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 Employee Name and TotalPay

**Output:** -

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

EmployeeName BasePay

NATHANIEL 167411

GARY 155966

ALBERT 212739

CHRISTOPHER 77916

PATRICK 134401

DAVID 118602

ALSON 92492

DAVID 256576

JOANNE 285262

PATRICIA 99722

EDWARD 294580



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

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

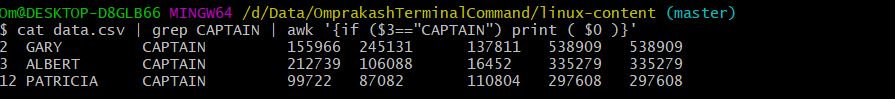
Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

2 GARY CAPTAIN 155966 245131 137811 538909 538909

3 ALBERT CAPTAIN 212739 106088 16452 335279 335279

12 PATRICIA CAPTAIN 99722 87082 110804 297608 297608



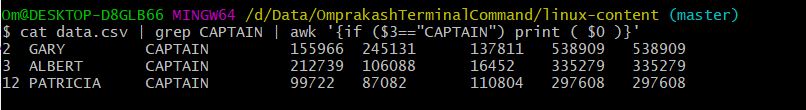
Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

2 GARY CAPTAIN 155966 245131 137811 538909 538909

3 ALBERT CAPTAIN 212739 106088 16452 335279 335279

12 PATRICIA CAPTAIN 99722 87082 110804 297608 297608

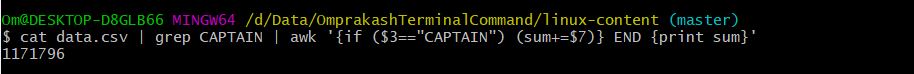


b) Extract TotalPay and calculate sum. Print the result on terminal

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

1171796



3) Print JobTitle and Overtimepay who has Overtimepay is between 7000 and 10000

a) Read data file data.csv from command line and extract jobtitle and overtime pay for column value range between 700010000

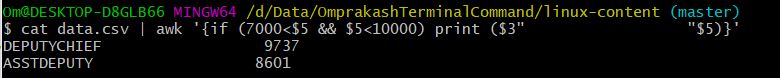
1. Print the result on terminal.

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

DEPUTYCHIEF 9737

ASSTDEPUTY 8601



4) Print average Base Pay

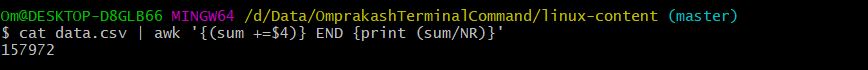
a) Read data file 'data.csv' from command line and extract BasePay values and calculate its average

b) Print the result on terminal.

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/OmprakashTerminalCommand/linux-content (master)

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

157972



**Or**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ13

$ nano Q13.sh

#! /bin/bash

# i) print EmployeeName and TotalPay who has BasePAy grater than 10000

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

echo "------------------------------"

# ii) What is the aggregate otalPay of employees whoose jobtitle is CAPTAIN

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

echo "------------------------------"

# iii) Print JobTitle and OverTimepay who has Ovevrtime is between 7000 and 10000

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

echo "------------------------------"

# iv) Print average BasePay

cat data.csv | awk '{sum += $4; cnt +=1}END {print sum/cnt}'

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ13

$ ./Q13.sh

EmployeeName : BasePay

NATHANIEL : 167411

GARY : 155966

ALBERT : 212739

CHRISTOPHER : 77916

PATRICK : 134401

DAVID : 118602

ALSON : 92492

DAVID : 256576

JOANNE : 285262

PATRICIA : 99722

EDWARD : 294580

------------------------------

Aggregat Pay : 1171796

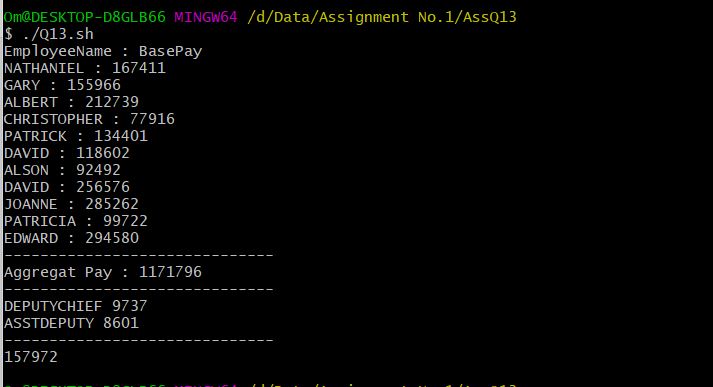
------------------------------

DEPUTYCHIEF 9737

ASSTDEPUTY 8601

------------------------------

157972



Q.14 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.

**Output: -**

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14

$ mkdir original updated

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14

$ cd original

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/original

$ nano original-file.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/original

$ cat original-file.sh

Omprakash

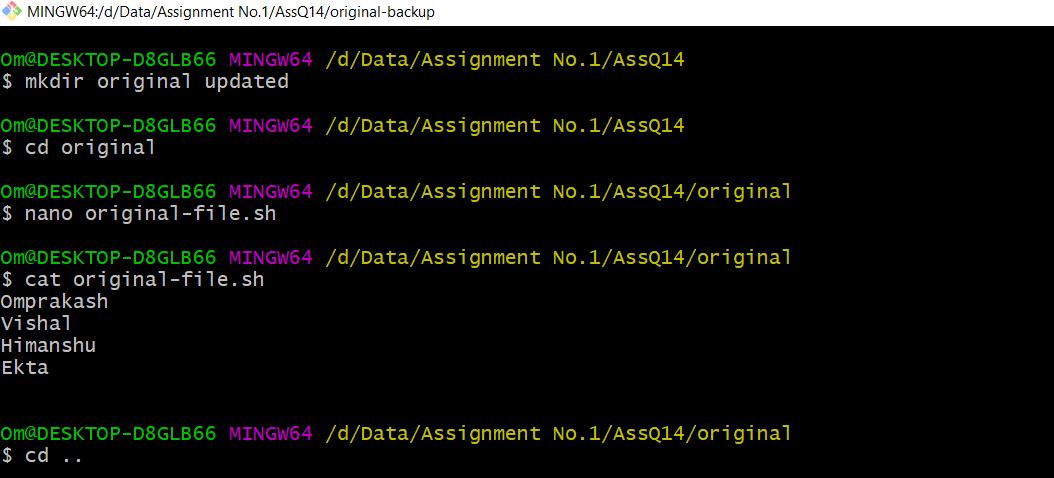
Vishal

Himanshu

Ekta

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/original

$ cd ..



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14

$ cd updated/

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/updated

$ nano updated-file.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/updated

$ cat updated-file.sh

Omprakash

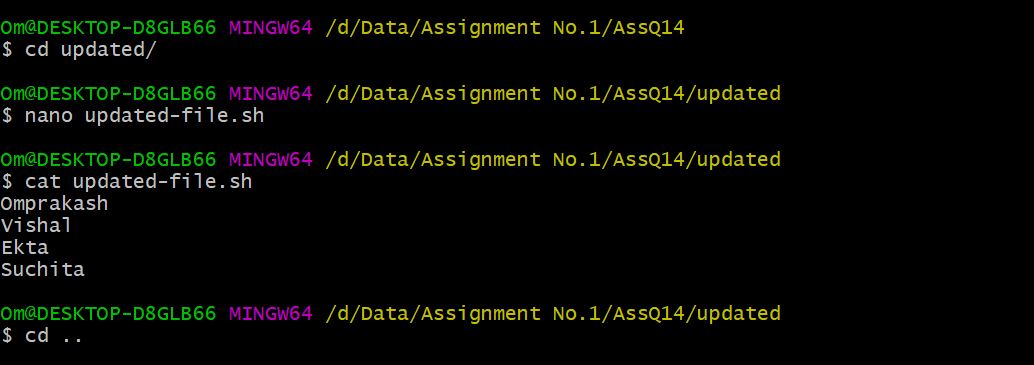
Vishal

Ekta

Suchita

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/updated

$ cd ..



#! /bin/bash

diff=`diff original/original-file.sh updated/updated-file.sh`

echo ${#diff}

if [ ${#diff} -gt 0 ]

then

echo "There is Difference in Files"

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

echo "applied changes to UpdatdeFolder"

else

echo "No difference found !! "

fi

if [ -d 'original-backup' ]

then

cp original/original-file.sh original-backup

else

mkdir original-backup

cp original/original-file.sh original-backup

fi

diff2=`diff original-backup/original-file.sh updated/updated-file.sh`

if [ ${#diff2} == 0 ]

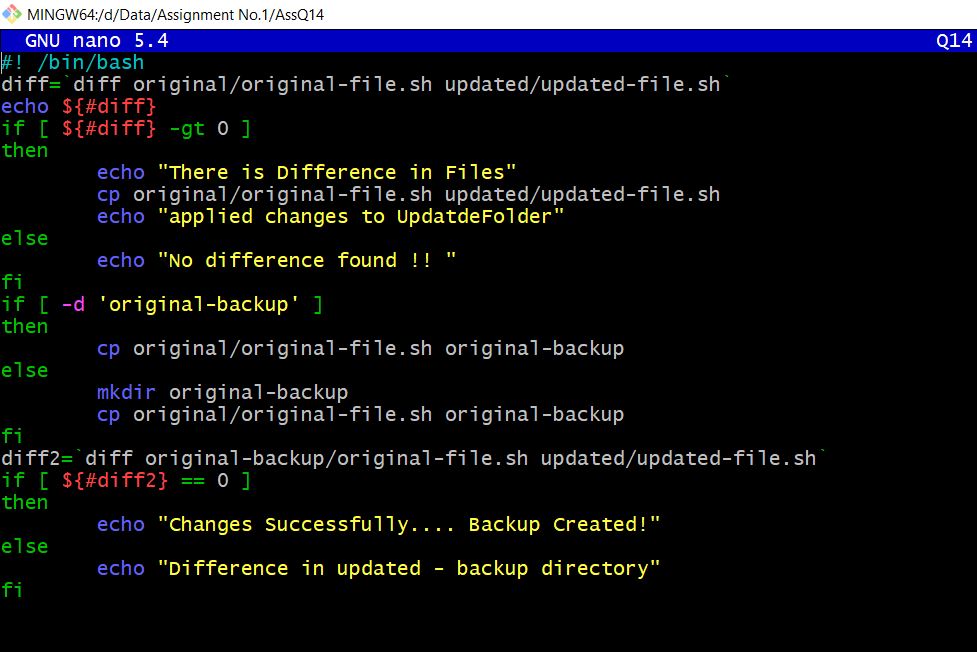
then

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

else

echo "Difference in updated - backup directory"

fi



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14

$ ./Q14.sh

35

There is Difference in Files

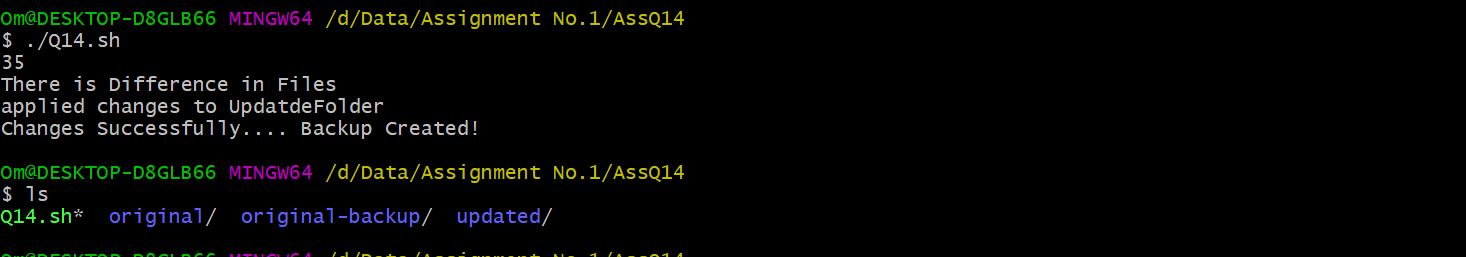
applied changes to UpdatdeFolder

Changes Successfully.... Backup Created!

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14

$ ls

Q14.sh\* original/ original-backup/ updated/



Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14

$ cd original-backup

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/original-backup

$ ls

original-file.sh

Om@DESKTOP-D8GLB66 MINGW64 /d/Data/Assignment No.1/AssQ14/original-backup

$ cat original-file.sh

Omprakash

Vishal

Himanshu

Ekta

