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.

Solution:

MINGW64:/c/Users/pc/Desktop/codinclub/day3/q1

```
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q1
$ for file in `ls *.txt`;
> do
> folderName=`echo $file | awk -F . '{ print $1 }'`;
> mkdir $folderName;
> cp $file $folderName/;
> echo "copied $file to $folderName/";
> done
copied abc.txt to abc/
copied def.txt to def/
copied ghi.txt to ghi/
copied jkl.txt to jkl/

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q1
$ ls
abc/  abc.txt  def/  def.txt  ghi/  ghi.txt  jkl/  jkl.txt
```

Q2

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.."

Solution:

```
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q2
$ ls
temp/

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q2
$ notepad foldertest.sh

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q2
$ bash foldertest.sh
Enter Folder Name
temp
Folder already exists

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q2
$ bash foldertest.sh
Enter Folder Name
test

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q2
$ ls
foldertest.sh*  temp/  test/
```

Q3

Execute command "hello" and "ls" 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 "ls" command at command prompt
d) Check execution status of "ls" command

Solution:

MINGW64:/c/Users/pc/Desktop/codinclub/day3/q3

```
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q3
$ echo "hello"
hello

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q3
$ echo $?
0

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q3
$ echo "ls"
ls

pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q3
$ echo $?
0
```

Q4

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	294580

i) Print EmployeeName 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 EmployeeName and TotalPay

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.

iii) 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 overtimepay for column value range between 7000-10000
b) Print the result on terminal.

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.

Solution:

i)

```
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub
$ cat data.csv | awk '{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)

```
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub
$ cat data.csv | grep -i CAPTAIN | awk '{sum+=$7} END {print sum}'
1171796
```

iii)

```
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub
$ cat data.csv | awk '{sum+=$4} END {print sum/NR}'
157972
```

iv)

```
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub
$ cat data.csv | awk '{if ($5>7000&&$5<10000) print $3,$5}'
DEPUTYCHIEF 9737
ASSTDEPUTY 8601
```

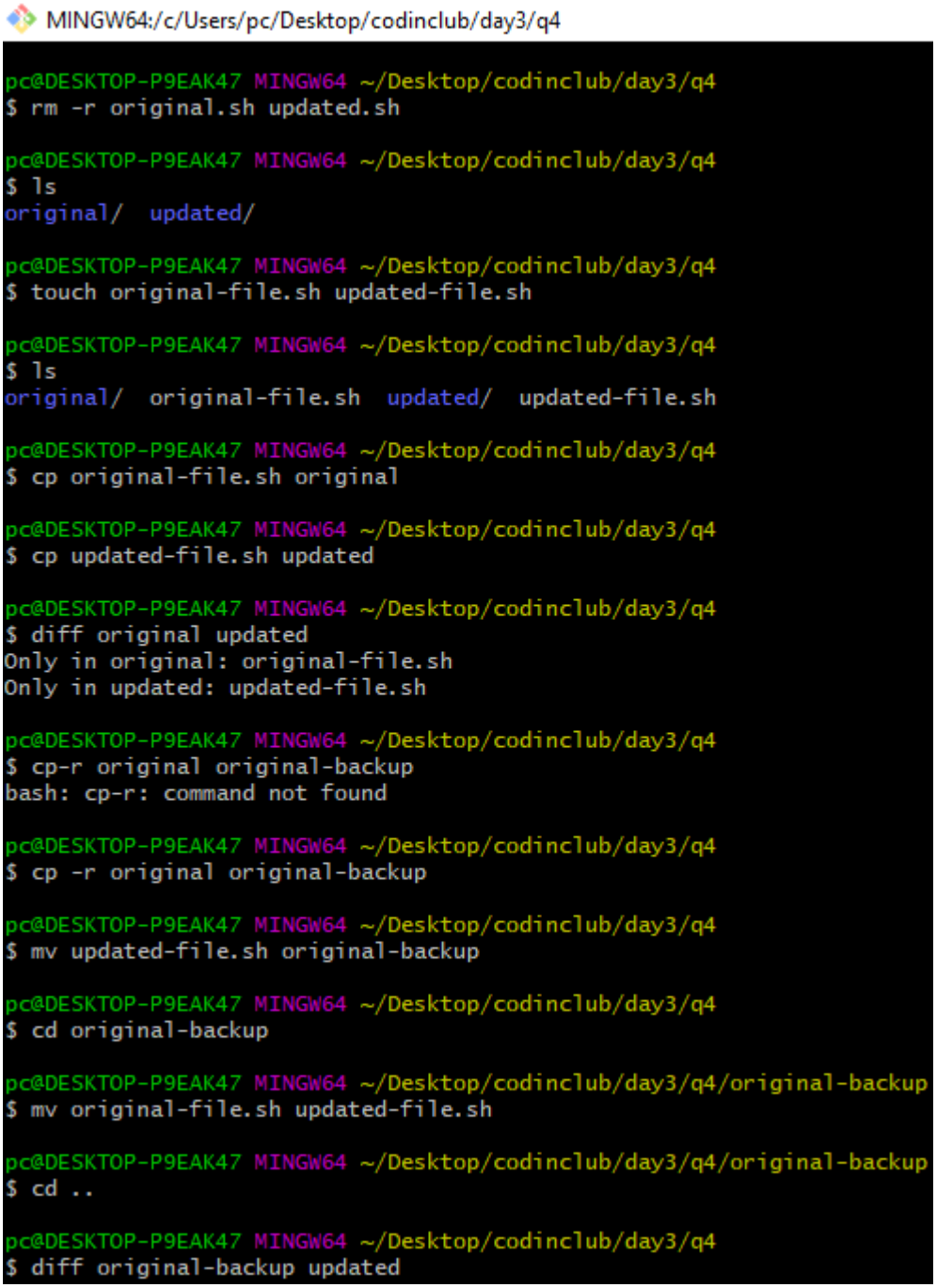
Q5

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.

Solution:


MINGW64:/c/Users/pc/Desktop/codinclub/day3/q4
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ touch original.sh updated.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ ls
original.sh updated.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ mkdir original updated
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ ls
original/ original.sh updated/ updated.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ rm -r original.sh updated.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ ls
original/ updated/
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ touch original-file.sh updated-file.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ ls
original/ original-file.sh updated/ updated-file.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp original-file.sh original
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp updated-file.sh updated
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ diff original updated
Only in original: original-file.sh
Only in updated: updated-file.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp-r original original-backup
bash: cp-r: command not found
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp -r original original-backup
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ mv updated-file.sh original-backup


MINGW64:/c/Users/pc/Desktop/codinclub/day3/q4
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ rm -r original.sh updated.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ ls
original/ updated/
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ touch original-file.sh updated-file.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ ls
original/ original-file.sh updated/ updated-file.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp original-file.sh original
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp updated-file.sh updated
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ diff original updated
Only in original: original-file.sh
Only in updated: updated-file.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp-r original original-backup
bash: cp-r: command not found
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cp -r original original-backup
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ mv updated-file.sh original-backup
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ cd original-backup
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4/original-backup
$ mv original-file.sh updated-file.sh
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4/original-backup
$ cd ..
pc@DESKTOP-P9EAK47 MINGW64 ~/Desktop/codinclub/day3/q4
$ diff original-backup updated