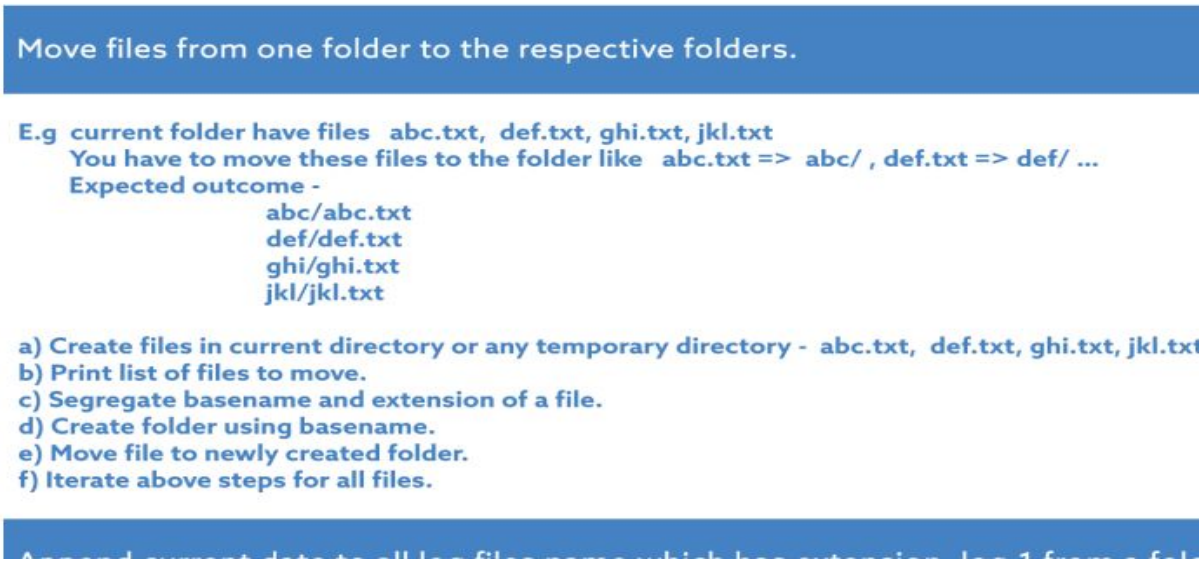
drwx------ 6 ashishv ashishv 4096 Aug 6 12:48 ashishv




```
#!/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;
        mv $file $folderName;
done
```

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.

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 ddmmyy
e) Append Date to the log file name
f) Iterate above steps for all files which has extension .log.1

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

```
#!/bin/bash

for file in `ls *.1`
do
        folderName=`echo $file | awk -F. '{print $1}'`
        currentDate=$(date +'%d%m%Y');
        MiddleName=`echo $file | awk -F. '{print $2}'`
        if [ -d $folderName ]
        then
                rm -r $folderName;
        fi
        mkdir $folderName;

        mv $file  $folderName-$currentDate.$MiddleName
        mv *.log $folderName

done
```

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

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

L I N U X C C

```
#!/bin/bash

for file in `ls *.log`
do
        folderName=`echo $file | awk -F. '{print $1}'`;
        currentdateAgo=$(date --date="7 days ago" +%d%m%y);
        creationDate=`echo $file | awk -F. '{print $2}'`;
        If [$creationDate <= $currentdateAgo ]
        then
                mv $file backupDir;
        fi

done
```

L I N U X

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.."

```
#!/bin/bash

print "Enter the Folder Name: ";
read folderName;
if [ -d $folderName ]
then
 echo "$folderName";
Else
```

```
mkdir $folderName
fi
```

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

```
#!/bin/bash

abc=`hello`
echo $?
def=`ls`
echo $?
```

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

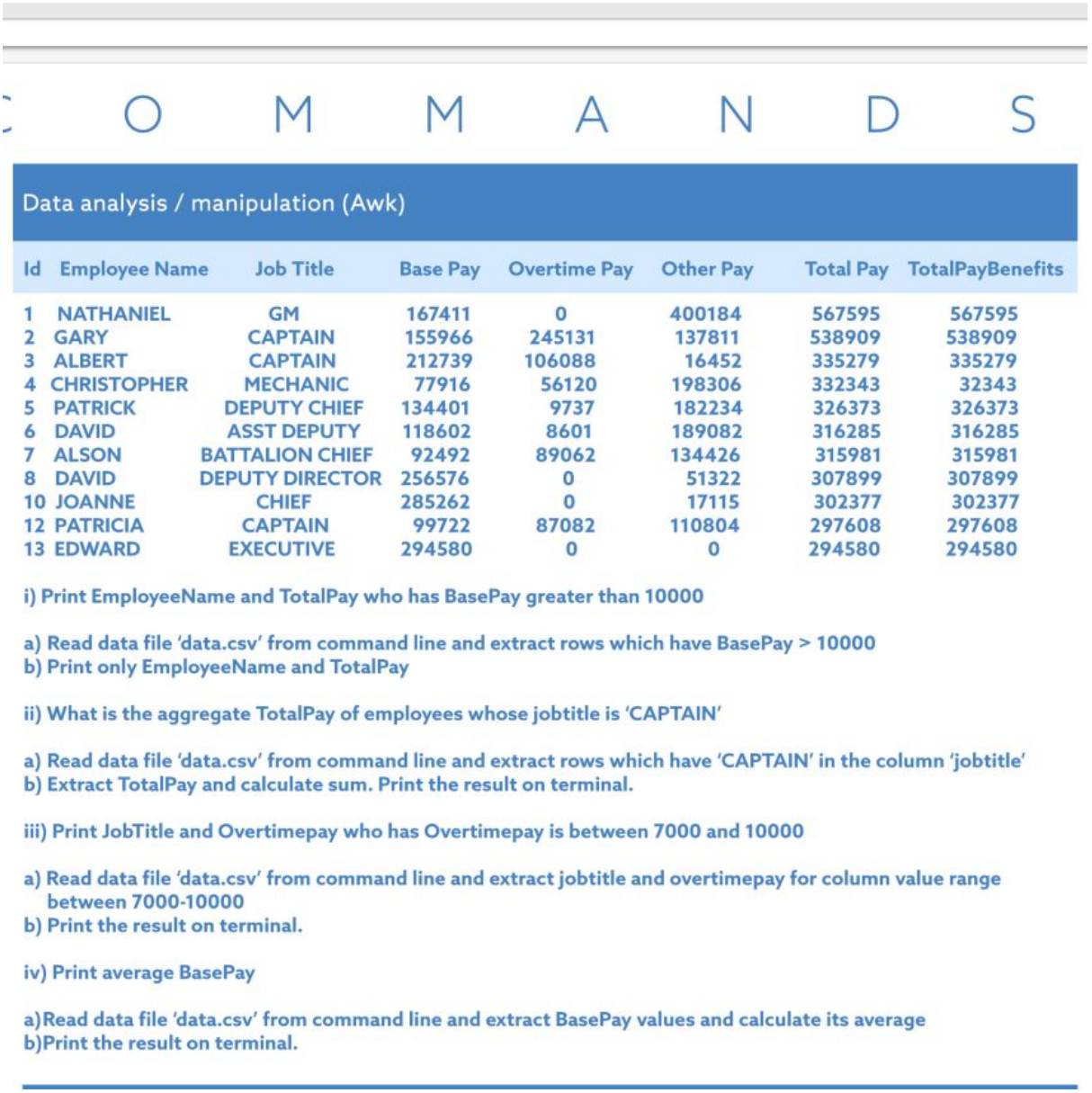
a) Use linux command to search word and print occurrence

```
#!/bin/bash -x

for file in `ls *.log`
do
count=`find . -name $file | xargs grep -i "systemd" | wc -l`

echo $count " " $file

done
```



1)

```
#!/bin/bash -x

data=`cat data.csv | awk '{if ($7 >="10000") print NR $2 " " $7}'`;
echo "$data "              ""
```

2)

```
#!/bin/bash -x

data=`cat data.csv |grep -i CAPT |  awk '{total= total+$7}END{print total}'`;
echo "$data "              ""
```

3)

```
!/bin/bash -x
```

```
data=`cat data.csv |  awk '{ if($5 >= "7000" && $5 <= "10000") print NR " " $2" "$5 }'`
echo "$data "  ""
```

4)

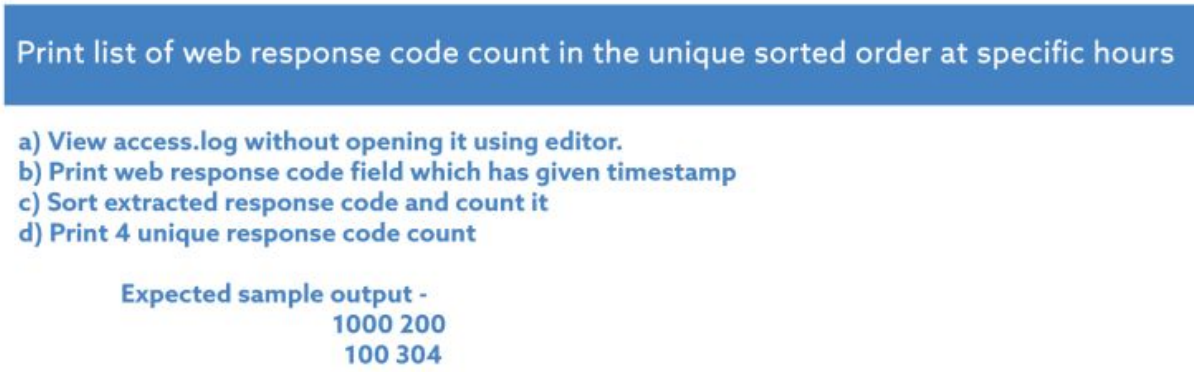
```
#!/bin/bash -x

data=`cat data.csv |grep -i CAPT |  awk '{total= total+$4}END{print total/NR}'`;
echo "$data"
```


```
#!/bin/bash -x

data=`grep -o "[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}" access.log | sort -n | uniq -c | sort -n | sort -rn | head -n 4`;

echo “$data “ “ “;
```




```
#!/bin/bash -x

data=`cat access.log | cut -d '"' -f3 | cut -d ' ' -f2 | sort | uniq -c | sort -rn | head -n 2`;

echo “$data “ “ “;
```


```
#!/bin/bash -x

data=`awk -vDate=`date -d'now-2 hours' +[%d/%b/%Y:%H:%M:%S` '$4 > Date {print $11}' access.log | grep -i “http” | sort -n | uniq -c | sort -n | sort -rn | head -n 4`;

echo “$data “ “ “;
```


```
#!/bin/bash -x

data=`cat access.log | grep -i "http" | awk '{print $11}' |grep -i "http" | sort | uniq -c |sort -nr | head -n 4`;

echo “$data “ “ “;
```

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

PID	PPID	CMD	%MEM	%CPU
760	1	/usr/bin/dockerd -H unix://	3.5	0.0
776	1	/usr/bin/containerd	0.7	0.1
7266	757	sshd: root@pts/0	0.6	0.0
759	1	/usr/sbin/rsyslogd -n	0.5	0.0
347	1	/usr/lib/systemd/systemd-jo	0.3	0.0
484	1	/usr/sbin/NetworkManager	0.3	0.0
1	0	/usr/lib/systemd/systemd	0.2	0.0
7268	7266	-bash	0.2	0.0
758	1	/usr/bin/python -Es /usr/sb	0.1	0.0

Command for creating:

$ top | awk '{print $2 " " $4 " " $13 " " $11 " " $10 }'

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

a) View /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

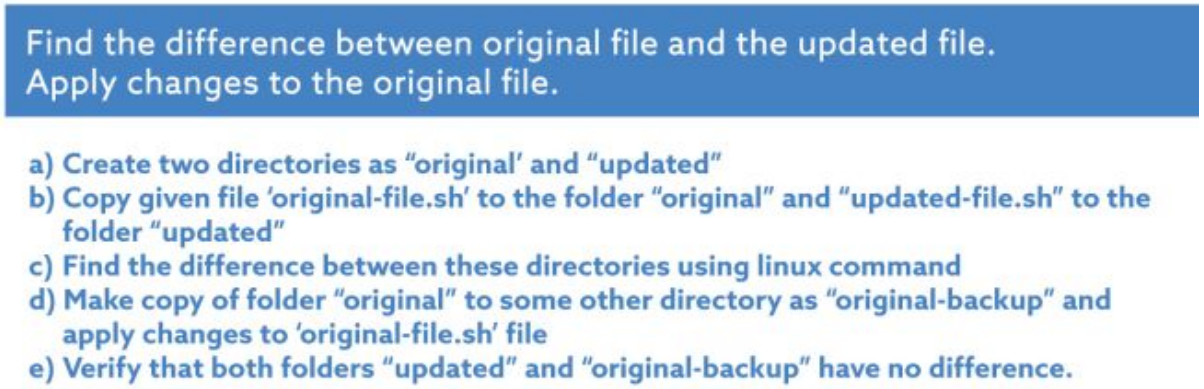
E.g /etc/passwd =>
ashishv:x:1002:1002::/home/ashishv:/bin/bash
extract values mark in bold - user and home directory
Expected output -
/home/ashishv:
drwx------ 6 ashishv ashishv 4096 Aug 6 12:48 ashishv

```
#!/bin/bash

for user in `cat /etc/passwd`
do
```

```
        UID= `cat /etc/passwd | awk -F: '{ print $3}'`
        if($UID -ge "1000")
        then
                homeDir= `cat /etc/passwd | awk -F: '{ if($3 >= 1000) print $7}'`

                Ownership= `ls -l /etc/passwd`
                chmod u+x /etc/passwd;
                changedOwInfo=`ls -l /etc/passwd | awk '{ print $6" " $7 " " $8 " "}'`
                echo " $homeDir :"
                echo  " $changedOwInfo ";
        fi
done
```



a)

```
        $ Mkdir original updated
```

b)

```
        $ cp original-file.sh original
        $ cp updated-files.sh updated
```

c)

```
        diff -q original/ updated/
Only in original/: original-file.sh
Only in updated/: updated-files.sh
```

d)

```
        $ cp original-file.sh original-backup
        $ diff -q original/ updated/
Only in original/: original-file.sh
Only in original/: original-backup
```

Only in updated/: updated-files.sh