Linux Commands

1) Move files from one folder to the respective folders.

```
#!/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
```

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

```
#!/bin/bash -x
for file in `ls *.log.1`;
do
  file1=`echo $file | awk -F. '{ print $1 }'`;
  date=`date +"%d%m%Y"`;
  cp $file $file1"-"$date".log";
done
```

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

```
#!/bin/bash -x
for file in `find . -name "*.txt" -mtime -7`;
do
  if [ -d $file ] ;
  then
    mv $file "backup";
  fi
done
```

4) Print last 4 frequently access URLs count in sorted order from /var/log/httpd/access.log.

```
#!/bin/bash -x

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

5) Print the last 4 frequently access unique URLs at particular hours from /var/log/httpd/access.log.

```
#!/bin/bash -x

cat access.log | awk '$4~/Oct/{print $11}' | sort | uniq -c | sort -nr | head -4
```

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

```
#!/bin/bash -x

cat access.log | awk '{print $10}' | sort | uniq -c | sort -nr | head -4
```

7) Print list of last 10 unique sorted clients IP from /var/log/httpd/access.log.

```
#!/bin/bash -x

cat access.log | awk '{print $17}' | sort | uniq -c | sort -nr | head -4
```

8) Check if a folder exists or not. If it's not present create it.

```
#!/bin/bash
read -p "Enter Your Folder Name:" name;
echo $name;
if [ -d $name ];
  then
    echo "Folder is already exist";
  else
    mkdir $name;
fi
```

9) Execute command “hello” and “ls” and check its execution status and print whether a command executed successfully or not.

```
#!/bin/bash -x
read -p "enter command:" command;
$command;
execute=`echo $?`;
if [ $execute -eq "0" ];
  then
    echo "Command Successfully Executed";
  else
    echo "Command Not Executed"
fi
```

10) Set environment usersecret=”dH34xlaa23” if it’s already not set.

```
#!/bin/bash -x
if [ -z "$usersecret" ]
then
      export usersecret='dH34xJaa23';
else
      echo "Error Occured: usersecret Already set..";
fi
echo "usersecret:" $usersecret;
```

11) Find the word “systemd” form all log files in the folder /var/log and print the number of occurrences more than 0 against each file.

```
#!/bin/bash -x
for file in `ls *.log`;
do
  occurrence=`grep -c "systemd" $file`;
  echo $occurrence;
done
```

12) Create a process list table displays process ID, parent process ID, command name, % of memory consumption, % of CPU utilization.

```
#!/bin/bash -x

ps -elf | awk '{ print $2 "    " $3 "    " $6 }';
```

13) Data analysis/manipulation (Awk)
    i) Print EmployeeName and TotalPay who has BasePay greater than10000.
    ii) What is the aggregate TtalPay of employee's job title is "CAPTAIN".
    iii) Print JobTitle and OvertimePay who has OvertimePay is between 7000 and 10000.
    iv) Print average BasePay.

```
#!/bin/bash -x
empDetail=`cat data.csv | awk '{ $4>10000 }END{ print $2 " " $7 }'`;
echo $empDetail;
totlePay=`cat data.csv | grep -i captain | awk '{ sum+=$7 }END{ print sum }'`;
echo $totlePay;
jobTitle=`cat data.csv | awk '{$6>7000&&$6<10000}END{print $3}'`;
echo $jobTitle;
avgBase=`cat data.csv | awk '{ sum+=$4 }END{ print sum/NR }'`;
echo $avgBase;
```

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

```
#!/bin/bash -x

for i in 'diff Original/Vishal.txt Updated/Vishal.txt'
do
  cp Original/Vishal.txt Updated/Vishal.txt;
done
```