



BUBT
Committed to Academic Excellence

**BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY**

Lab Assessment

Course Code: CSE 310

Course Title: Operating Systems

Submitted to:

Name: Suman Saha
Assistant Professor
Dept. of CSE
at Bangladesh University of Business
and Technology.

Submitted by:

Name: Syeda Nowshin Ibnat
ID: 17183103020
Intake: 39
Section: 01
Program: B.Sc. in CSE
Semester: Fall 20-21

Date of Submission: 23/03/2021

1. What commands can you use to determine who is logged in on a specific terminal?

```
nowshin@Lenovoip320:~$ finger user
Login: cups-pk-helper           Name: user for cups-pk-helper service
Directory: /home/cups-pk-helper Shell: /usr/sbin/nologin
Never logged in.
No mail.
No Plan.

Login: hplip                    Name: HPLIP system user
Directory: /run/hplip          Shell: /bin/false
Never logged in.
No mail.
No Plan.
nowshin@Lenovoip320:~$ who
nowshin :0      2021-03-23 14:59 (:0)
nowshin@Lenovoip320:~$
```

2. List all files whose names begin with the letter g, —have any three characters following the g, —and end with “tly.txt”.

```
nowshin@Lenovoip320:~$ touch grftly.txt
nowshin@Lenovoip320:~$ touch gdftly.txt
nowshin@Lenovoip320:~$ touch gkftly.txt
nowshin@Lenovoip320:~$ touch greatly.txt
nowshin@Lenovoip320:~$ ls g???tly.txt
greatly.txt
nowshin@Lenovoip320:~$
```

3. Which command can you use to look line#2 to line#6 of a file named status.report.

```
nowshin@Lenovoip320:~$ sed -n 2,6p status.report
nowshin@Lenovoip320:~$
```

4. Change the permission for the newly created file year2020 so that anyone can read and execute the file.

Ans:

To create new file touch year200.txt

```
nowshin@Lenovoip320:~$ touch year2020.txt
```

anyone to read and execute the file.

Before:

```
-rw-rw-r-- 1 nowshin nowshin 3248 सार्व 23 15:40 year2020.txt
```

After:

```
-rwxrwxr-x 1 nowshin nowshin 3248 মাঝে 23 15:40 year2020.txt
```

5. Give an example of a command that uses grep:

i. With both input and output redirected

```
nowshin@Lenovoip320:~$ grep best status.report  
nowshin@Lenovoip320:~$
```

ii. Within a pipe.

```
nowshin@Lenovoip320:~$ grep best status.report | cat > extracted.txt  
nowshin@Lenovoip320:~$
```

6.

a. List all files that begin with section.

Command: ls section*

```
nowshin@Lenovoip320:~$ ls section*  
section1 section2 section3 section4a section4b  
nowshin@Lenovoip320:~$
```

b. List the section1, section2, and section3 files only.

Command: ls section[1-3]

```
nowshin@Lenovoip320:~$ ls section[1-3]  
section1 section2 section3  
nowshin@Lenovoip320:~$
```

7. Create a file named book that contains the contents of two other files: part1 and part2.

```
nowshin@Lenovoip320:~$ cat part[12] > book
```

8. Create a shell script program that takes an input file and perform the following operations:

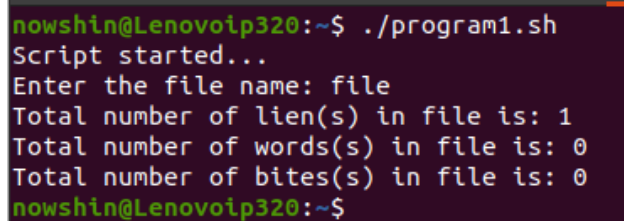
- i. Count the number of lines of the file.
- ii. Calculate the number of words and bytes of the file.

Ans:

Sample code:

```
#!/usr/bin/bash
echo "Script started..."
read -p "Enter the file name: " file
totalLine=$(wc -l < $file)
totalWord=$(wc -w < $file)
totalBite=$(wc -c < $file)
((totalLine+=1))
echo "Total number of lien(s) in $file is: $totalLine";
echo "Total number of words(s) in $file is: $totalWord";
echo "Total number of bites(s) in $file is: $totalBite";
```

Sample Output:



```
nowshin@Lenovoip320:~$ ./program1.sh
Script started...
Enter the file name: file
Total number of lien(s) in file is: 1
Total number of words(s) in file is: 0
Total number of bites(s) in file is: 0
nowshin@Lenovoip320:~$
```

9. Create a shell script program to implement the shortest job first scheduling algorithm.

Sample Code:

```
#!/bin/bash

clear ;

n=0;

read -p "Echo number of process: " n;

declare -a bt[n];

declare -a p[n];

declare -a wt[n];

declare -a tat[n];

bt[0]=0

# get burst time

echo "Enter Burst time: ";

for((i=0; i<n; i++)); do

((N=i+1))

read -p "process $N: " bt[i];

(( p[$i]=i+1 ))

done

btn=( $( printf "%s\n" "${bt[@]}" | sort -n ) )

(( wt[0]=0 ))

for (( i = 1; i < n; i++ )); do

(( wt[i]=0 ))

for(( j = 0; j < i; j++ )); do

(( wt[i]+=btn[j] ))

done

(( total+=wt[i] ))

done
```

```

(( avg_wt=total/n ))
(( avg_wt+=total%n ))
(( total = 0))
echo "Process   Burst Time   Waiting Time Turnaround Time";
for(( i = 0; i < n; i++ )) {
(( tat[i]=btn[i]+wt[i] ))
(( total+=tat[i] ))
echo "${p[i]}      ${btn[i]}      ${wt[i]}      ${tat[i]}" }
(( avg_tat=total/n ))
(( avg_tat+=total%n ))
echo "Average Waiting Time= " $avg_wt;
echo "Average Turnaround Time= " $avg_tat;
echo ;
echo ;
read -p "Enter Burst time: " burst;

```

Sample Output:

```

Echo number of process: 4
Enter Burst time:
process 1: 6
process 2: 8
process 3: 7
process 4: 3
Process   Burst Time   Waiting Time Turnaround Time
1         3           0           3
2         6           3           9
3         7           9          16
4         8          16          24
Average Waiting Time= 7
Average Turnaround Time= 13

```