



LAB MANUAL

Operating Systems



Submitted To:

SIR AMIR SABRI

Submitted By:

NAEEM UR RAHMAN SAJID

Registration No:

FA20-BCS-099

Comsats University, Islamabad (Sahiwal Campus)

Table of Content

1. UNIX commands.....	6
1.1. Date:.....	6
1.4. whoami:.....	6
1.5. Man command:.....	6
1.6. Calendar:.....	6
1.7. mkdir:.....	7
1.8. Print working directory:	7
1.9. Cat: (Create a file or add text in it):.....	7
1.10. Cat: (Print text of file in the bash):	7
1.11. Cat >> myfile (append the file):.....	7
1.12. Clear:.....	7
1.13. Head: (Print first 10 lines of file):	7
1.14. Tail (Print the last 10 lines):	7
1.15. wc (Word count):.....	7
1.16. Cp (copy file):	8
1.17. Rm (Remove file):.....	8
1.18. Vi (Open vi editor):	8
1.19. Nano (text editor):	8
1.20. Command -help:.....	8
2. SHELL VARIABLES:	9
2.1. Declaring Variables:	9
2.2. Displaying Values of Variables:	9
2.3. Making String using the variables:.....	9
2.4. Arithmetic Commands:	9
2.5. READ Statements:	9
2.6. READ more than one values at time:.....	10
2.7. ECHO Statements:	10
3. SHELL SCRIPTING:.....	11
3.1. CONCATENATION OF TWO STRINGS	11
3.2. COMPARISON OF TWO STRINGS.....	11

4. System Calls	13
4.1. To Create a Process in LINUX.....	13
4.2. Executing a command in LINUX.....	13
4.3. SLEEP COMMAND in LINUX	14
4.4. Sleep Command using getpid.....	14
4.5. Python in LINUX in multiprocessing.....	15
5. Lab Work Python Script	16
5.1. Script with Python Script in LINUX.....	16
6. Linux Commands.....	17
7. Making Executable script	20
7.1. Script 1.....	20
7.2. Script 2.....	21
7.3. Script 3.....	22
7.4. Script 4.....	23
7.5. Script 5.....	23
8. Creating the process in C++:.....	24
9. Creating the n child process in C++:	25
10. Creating the threads in C++:.....	27
11. To Create a Threads in LINUX and Assign the cores to threads:.....	28
12. Process Scheduling Algorithms in C++:	31
13. Sub-netting Calculator using Python.....	43
14. Bankers Algorithms in C++:	47
15. Creating the Charts in Python.....	60
16. Creating the Excel Workbook in Python.....	61

1. UNIX commands

1.1. Date:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ date
Tue Oct 4 09:10:05 AM PKT 2022
```

1.2. Echo for printing text in bash:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ echo "today is 'date'"
today is 'date'
```

1.3. Print date by back comma in echo:

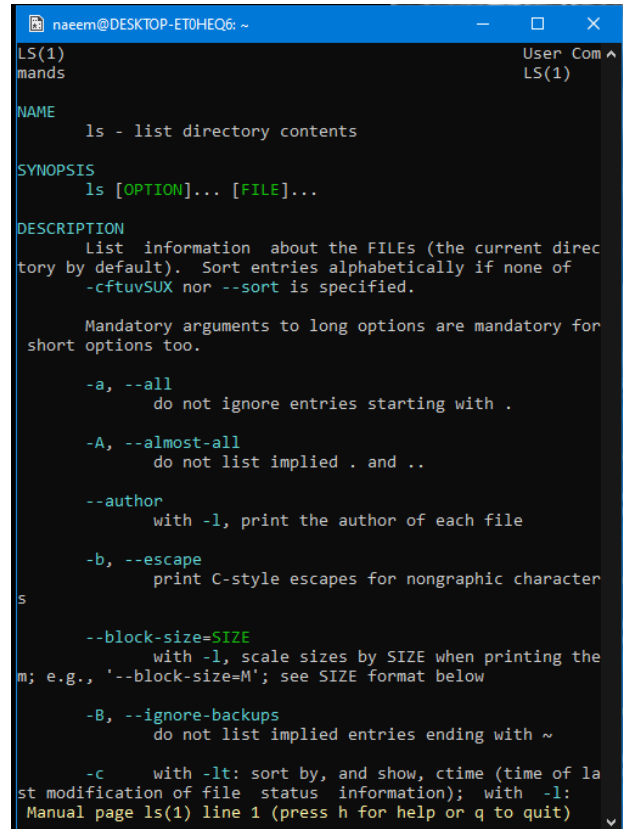
```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ echo "today is `date`"
today is Tue Oct 4 09:10:37 AM PKT 2022
```

1.4. whoami:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ whoami
naeem
```

1.5. Man command:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ man ls
```



```
naeem@DESKTOP-ET0HEQ6: ~
LS(1)
mands
User Com ^
LS(1)
NAME
ls - list directory contents
SYNOPSIS
ls [OPTION]... [FILE]...
DESCRIPTION
List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
Mandatory arguments to long options are mandatory for short options too.
-a, --all
do not ignore entries starting with .
-A, --almost-all
do not list implied . and ..
--author
with -l, print the author of each file
-b, --escape
print C-style escapes for nongraphic characters
--block-size=SIZE
with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
-B, --ignore-backups
do not list implied entries ending with ~
-c with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: Manual page ls(1) line 1 (press h for help or q to quit)
```

1.6. Calendar:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ cal
```

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ cal
October 2022
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

1.7. mkdir:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ mkdir Desktop

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ ls

Desktop
```

1.8. Print working directory:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ pwd

/home/naeem
```

1.9. Cat: (Create a file or add text in it):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ cat > myfile
```

1.10. Cat: (Print text of file in the bash):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ cat myfile
```

1.11. Cat >> myfile (append the file):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ cat >> myfile
```

1.12. Clear:

It clears the screen of the bash

1.13. Head: (Print first 10 lines of file):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ head myfile
```

1.14. Tail (Print the last 10 lines):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ tail myfile
```

1.15. wc (Word count):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ wc myfile

18 20 53 myfile
```

1.16. Cp (copy file):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
```

```
$ cp myfile newfile
```

1.17. Rm (Remove file):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
```

```
$ rm newfile
```

1.18. Vi (Open vi editor):

```
(naeem@DESKTOP-ET0HEQ6)-[~]
```

```
$ vi myfile
```

1.19. Nano (text editor):

```
(naeem@DESKTOP-ET0HEQ6)-[~/Desktop]
```

```
$ nano myfile
```

```
naeem@DESKTOP-ET0HEQ6: ~
GNU nano 6.4 myinfo.sh
#!/bin/sh

echo "My username is : `whoami`"
echo " My name is Muhammad Abdullah"
sleep 2
echo " My cgpa is 3.6"
sleep 1
echo "I am the bachelor Student in comsats of computer Science"
sleep 2
echo "This is my first Script"
echo "We are learning the Shell Scripting in Operating System >
```

1.20. Command -help:

```
(naeem@DESKTOP-ET0HEQ6)-[~/Desktop]
```

```
$ cat --help
```

```
naeem@DESKTOP-ET0HEQ6: ~
(naeem@DESKTOP-ET0HEQ6)-[~]
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                do not ignore entries starting with .
-A, --almost-all        do not list implied . and ..
--author                 with -l, print the author of each file
-b, --escape             print C-style escapes for nongraphic characters
--block-size=SIZE        with -l, scale sizes by SIZE when printing them;
                        e.g., '--block-size=M'; see SIZE format below

-B, --ignore-backups     do not list implied entries ending with ~
-c                        with -lt: sort by, and show, ctime (time of last
                        modification of file status information);
                        with -l: show ctime and sort by name;
                        otherwise: sort by ctime, newest first

-C                        list entries by columns
--color[=WHEN]           color the output WHEN; more info below
-d, --directory          list directories themselves, not their contents
-D, --dired              generate output designed for Emacs' dired mode
-f                        list all entries in directory order
-F, --classify[=WHEN]   append indicator (one of */->@|) to entries WHEN
                        likewise, except do not append '*'
--file-type              likewise, except do not append '*'
WORD                     across -x, commas -m, horizontal -x, long -l,
                        single-column -1, verbose -l, vertical -C

--full-time              like -l --time-style=full-iso
-g                        like -l, but do not list owner
--group-directories-first
```


2. SHELL VARIABLES:

2.1. Declaring Variables:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ myname=naeem

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ age=20
```

2.2. Displaying Values of Variables:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $myname

naeem

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $age

20
```

2.3. Making String using the variables:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo "My name is $myname and i
am $age years old"

My name is naeem and i am 20 years old
```

2.4. Arithmetic Commands:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $((12+8))

20

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $((12*4))

48

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ ans=$((12*4))

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $ans

48
```

2.5. READ Statements:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ read

My name is Naeem

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ read myname

Naeem ur rahman sajid

(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $myname

Naeem ur rahman sajid
```

2.6. READ more than one values at time:

```

└─(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ read x y
23 34

└─(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $x
23

└─(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo $y
34
    
```

2.7. ECHO Statements:

```

└─(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ echo "My CGPA is 3.54 and my
name is $myname"

My CGPA is 3.54 and my name is Naeem
ur rahman sajid
    
```

3. SHELL SCRIPTING:

3.1. CONCATENATION OF TWO STRINGS

Script:

```
concatstring.sh
#!/bin/sh

echo "Enter the String 1 "
read str1

echo "Enter the String 2 "
read str2

echo "Concatenated String is : $str1 $str2
"
```

Output:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ ./concatstring.sh
Enter the String 1
naeem ur
Enter the String 2
rahman
Concatenated String is : naeem ur
rahman
```

3.2. COMPARISON OF TWO STRINGS

Script:

```
GNU nano 6.4
comparestring.sh

echo "Enter the First String "
read str1

echo "Enter the Second String "
read str2

if [ $str1 = $str2 ]
then
echo "Strings are equal"
else
echo "Strings are not equal"
fi
```

Output:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ ./comparestring.sh
Enter the First String
name
Enter the Second String
name
Strings are equal
```

3.3. MAXIMUM OF THREE NUMBERS

Script:

```
echo "Enter X"
read x
sleep 2
echo "Enter Y"
read y
sleep 1
echo "Enter Z"
read z
sleep 2
if [ $x -gt $y -a $x -gt $z ]
then
    echo "X: $x is the greater number "
elif [ $y -gt $x -a $y -gt $z ]
then
    echo " Y: $y is the greater Number "
else
    echo " Z: $z is the greater Number "
fi
```

Output:

```
└─(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ ./maxnum.sh
```

Enter X

3

Enter Y

6

Enter Z

1

Y: 6 is the greater Number

4. System Calls

4.1. To Create a Process in LINUX

Script:

```
#include<iostream>

#include<unistd.h>

using namespace std;

int main(){

    int id;

    id= fork () ;

    if(id<0){

        cout<<"Cant create a file"<<endl;

        exit(-1);

    }

    if(id==0){

        cout<<"Child process"<<endl;

        exit(0);

    }

    else{

        cout<<"Parent process"<<endl;

    }

    return 0;

}
```

Output:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
```

```
$ ./process_creation
```

Parent process

Child process

4.2. Executing a command in LINUX

Script:

```
GNU nano 6.4
executing_command.sh      echo
"Program for executing UNIX command
using the shell programming"

echo "Most Welcome"

ps

exec wc executing_command.sh
```

Output:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
```

```
$ ./executing_command.sh
```

Program for executing UNIX command
using the shell programming

Most Welcome

PID	TTY	TIME	CMD
46	tty1	00:00:00	bash
78	tty1	00:00:00	bash
79	tty1	00:00:00	ps
4	17	122	executing_command.sh

4.3. SLEEP COMMAND in LINUX

Script:

```
GNU nano 6.4
sleep_command.c++
#include<iostream>

#include<unistd.h>

using namespace std;

int main(){

    int id=fork();

    if(id==-1){

        cout<<"Cannot create the
file"<<endl;

        exit(1);

    }else if(id==0){

        sleep(2);

        cout<<"This is the Child
process"<<endl;

    }else{

        cout<<"Parent process"<<endl;

        exit(1);

    }

    return 0;

}
```

Output:

```
└─(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ ./sleep_command
```

Parent process

```
└─(naeem@DESKTOP-ET0HEQ6)-[~]
└─$ This is the Child process
```

4.4. Sleep Command using getpid

Script:

```
#include<iostream>

#include<unistd.h>

#include<sys/wait.h>

using namespace std;

int main(){

    int pid;

    pid=fork();

    if(pid==0){

        cout<<"Child Process"<<endl;

        cout<<"Child Process id is :
"<<getpid()<<endl;

        cout<<"Its Parent process :
"<<getppid()<<endl;

        sleep(5);

        cout<<"Child Process after sleep =
5"<<endl;

        cout<<"Child Process id is :
"<<getpid()<<endl;
```

```

    cout<<"Its Parent process :
"<<getppid()<<endl;

    }else{

    cout<<"Parent Process"<<endl;

    sleep(10);

    cout<<"Child Process id is :
"<<getpid()<<endl;

    cout<<"Its Parent process :
"<<getppid()<<endl;

    cout<<"Parent terminates !"<<endl;

    sleep(3);

    }

    return 0;

}

```

```

Its Parent process : 10

Parent terminates !

```

4.5. [Python in LINUX in multiprocessing](#)

```

>>> import platform

>>> print(platform.system())

Linux

>>> from multiprocessing import
cpu_count

>>> print(cpu_count())

4

```

Output:

```

└─(naeem@DESKTOP-ET0HEQ6)-
[~/Exercise3]

└─$ ./sleep_getpid

Parent Process

Child Process

Child Process id is : 52

Its Parent process : 51

Child Process after sleep = 5

Child Process id is : 52

Its Parent process : 51

Child Process id is : 51

```

5. Lab Work Python Script

5.1. [Script with Python Script in LINUX](#)

Script file:

```
#!/bin/bash
#SBATCH --job-name=multiproc-test
#SBATCH --output=multiproc-output.txt
#SBATCH -c 4
#SBATCH --time=15:00
#SBATCH --mem=100M

srun ./multi_proc.py 40 1

PYTHON SCRIPT

#!/usr/bin/env python3
```

Python Script:

```
import multiprocessing
import os
import sys

def call_stress(mins):
    os.system(f'timeout {mins}m stress -c 1')

if __name__ == '__main__':
    sub_tasks = int(sys.argv[1])
    mins = int(sys.argv[2])
```

```
cores =
int(os.environ['SLURM_CPUS_PER_TASK'])

with multiprocessing.Pool(cores) as
pool:

    pool.map(call_stress,
              [mins for i in range(sub_tasks)])
```


6. Linux Commands

```
naeem@DESKTOP-ET0HEQ6: ~  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ ls  
Desktop  himom.sh  myfile  myinfo.sh  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ ls -a  
.          .bashrc      himom.sh  myfile      .zshrc  
..         .bashrc.original .java     myinfo.sh  
.bash_history .config     .lesshst  .profile  
.bash_logout Desktop     .local    .sudo_as_admin_successful  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ ls -l  
total 0  
drwxr-xr-x 1 naeem naeem 512 Oct  4 09:57 Desktop  
-rwxr-xr-x 1 naeem naeem  79 Oct  4 21:33 himom.sh  
-rw-r--r-- 1 naeem naeem 168 Oct  4 09:51 myfile  
-rwxr-xr-x 1 naeem naeem 298 Oct  4 22:50 myinfo.sh  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ cd Desktop  
  
(naeem@ DESKTOP-ET0HEQ6)-[~/Desktop]  
$ cd ..  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ cd /  
  
(naeem@ DESKTOP-ET0HEQ6)-[/]  
$ cd  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ pwd  
/home/naeem  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ man ls  
  
(naeem@ DESKTOP-ET0HEQ6)-[~]  
$ ls --help  
Usage: ls [OPTION]... [FILE]...  
List information about the FILES (the current directory by default).  
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.  
Mandatory arguments to long options are mandatory for short options too.
```

```
naeem@DESKTOP-ET0HEQ6: ~/Desktop
(naeem@DESKTOP-ET0HEQ6)-[~]
$ mkdir Pictures

(naeem@DESKTOP-ET0HEQ6)-[~]
$ cal
    October 2022
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31

(naeem@DESKTOP-ET0HEQ6)-[~]
$ touch new.txt

(naeem@DESKTOP-ET0HEQ6)-[~]
$ cat > new.txt
new file
^Z
[1]+  Stopped                  cat > new.txt

(naeem@DESKTOP-ET0HEQ6)-[~]
$ cat new.txt
new file

(naeem@DESKTOP-ET0HEQ6)-[~]
$ cat >> new.txt
appending file xyz^Z
[2]+  Stopped                  cat >> new.txt

(naeem@DESKTOP-ET0HEQ6)-[~]
$ cp new.txt /home/naeem/Desktop

(naeem@DESKTOP-ET0HEQ6)-[~]
$ mv new.txt /home/naeem/Pictures

(naeem@DESKTOP-ET0HEQ6)-[~]
$ cd Desktop/

(naeem@DESKTOP-ET0HEQ6)-[~/Desktop]
$ rm new.txt
```

```

naeem@DESKTOP-ET0HEQ6: ~
└─(naeem@ DESKTOP-ET0HEQ6) - [~]
$ head myfile
1
2
3
4
5
6
7
8
9
10

└─(naeem@ DESKTOP-ET0HEQ6) - [~]
$ tail myfile
C
C
C
C
C
C
C
C
C
C
d

└─(naeem@ DESKTOP-ET0HEQ6) - [~]
$ echo "My Name is Naeem ur Rahman"
My Name is Naeem ur Rahman

└─(naeem@ DESKTOP-ET0HEQ6) - [~]
$ whoami
naeem

└─(naeem@ DESKTOP-ET0HEQ6) - [~]
$ date
Tue Oct  4 10:58:25 PM PKT 2022

└─(naeem@ DESKTOP-ET0HEQ6) - [~]
$ read
23434

└─(naeem@ DESKTOP-ET0HEQ6) - [~]
$

```

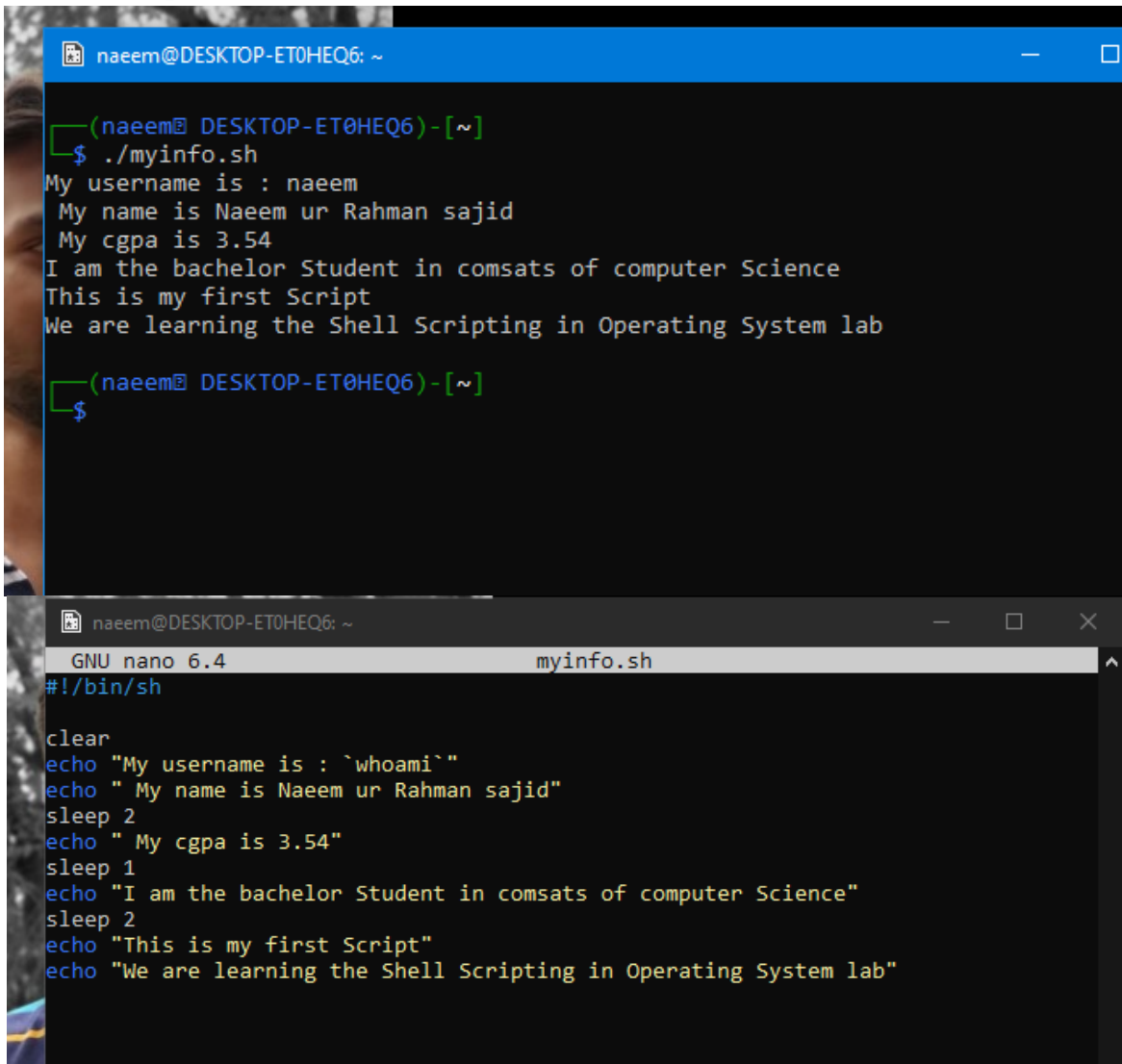
7. Making Executable script

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ nano variables.sh

(naeem@DESKTOP-ET0HEQ6)-[~]
$ chmod +x variables.sh

(naeem@DESKTOP-ET0HEQ6)-[~]
```

7.1. Script 1



```
naeem@DESKTOP-ET0HEQ6: ~
(naeem@DESKTOP-ET0HEQ6)-[~]
$ ./myinfo.sh
My username is : naeem
My name is Naeem ur Rahman sajid
My cgpa is 3.54
I am the bachelor Student in comsats of computer Science
This is my first Script
We are learning the Shell Scripting in Operating System lab

(naeem@DESKTOP-ET0HEQ6)-[~]
$
```

```
GNU nano 6.4 myinfo.sh
#!/bin/sh

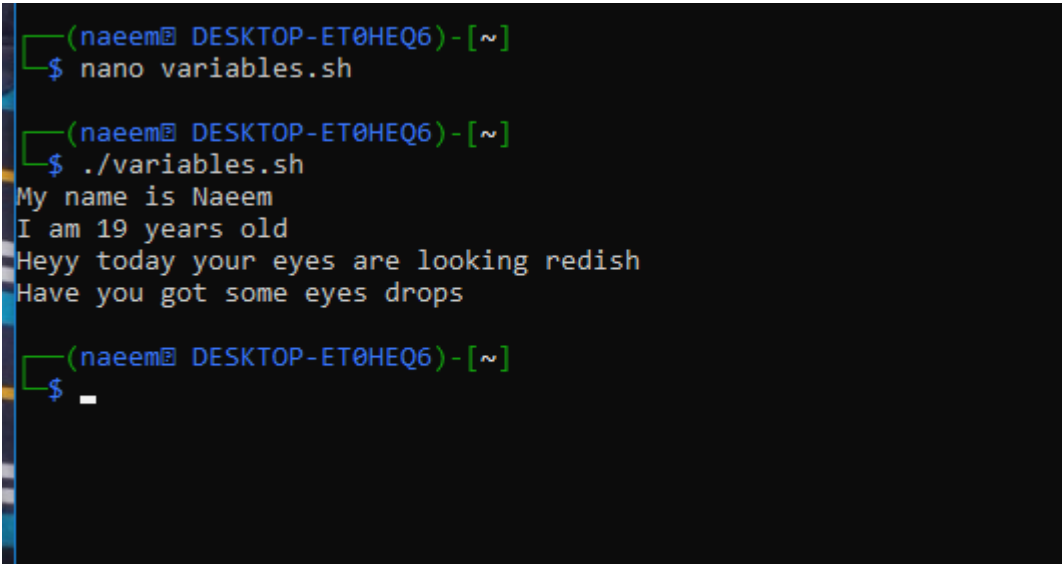
clear
echo "My username is : `whoami`"
echo " My name is Naeem ur Rahman sajid"
sleep 2
echo " My cgpa is 3.54"
sleep 1
echo "I am the bachelor Student in comsats of computer Science"
sleep 2
echo "This is my first Script"
echo "We are learning the Shell Scripting in Operating System lab"
```

7.2. Script 2

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ nano variables.sh

(naeem@DESKTOP-ET0HEQ6)-[~]
$ ./variables.sh
My name is Naeem
I am 19 years old
Heyy today your eyes are looking redish
Have you got some eyes drops

(naeem@DESKTOP-ET0HEQ6)-[~]
$
```



```
naeem@DESKTOP-ET0HEQ6: ~
GNU nano 6.4 variables.sh
#!/bin/sh

name="Naeem"
age="19"
complement="eyes"

echo "My name is $name"
sleep 2
echo "I am $age years old"
echo "Heyy today your $complement are looking redish"
sleep 3
echo "Have you got some $complement drops"
```

7.3. Script 3

```
naeem@DESKTOP-ET0HEQ6: ~  
GNU nano 6.4 table.sh  
#!/bin/sh  
  
echo "Enter the number"  
read val  
echo ""  
echo "Table of $val:"  
echo ""  
echo "$val * 01 = $((val*1))"  
echo "$val * 02 = $((val*2))"  
sleep 2  
echo "$val * 03 = $((val*3))"  
echo "$val * 04 = $((val*4))"  
echo "$val * 05 = $((val*5))"  
echo "$val * 06 = $((val*6))"  
echo "$val * 07 = $((val*7))"  
echo "$val * 08 = $((val*8))"  
echo "$val * 09 = $((val*9))"  
echo "$val * 10 = $((val*10))"
```

```
(naeem@DESKTOP-ET0HEQ6)-[~]  
$ ./table.sh  
Enter the number  
5  
  
Table of 5:  
  
5 * 01 = 5  
5 * 02 = 10  
5 * 03 = 15  
5 * 04 = 20  
5 * 05 = 25  
5 * 06 = 30  
5 * 07 = 35  
5 * 08 = 40  
5 * 09 = 45  
5 * 10 = 50
```

7.4. Script 4

```
(definitelykalilinux@ DESKTOP-MQD86LP)-[~]
$ ./hello20.sh
Enter first number : 10
Enter second number : 5
Addition : 15
Subtraction : 5
Multiplication : 50
Division : 2
```

7.5. Script 5

```
(definitelykalilinux@ DESKTOP-MQD86LP)-[~]
$ ./hello10.sh
Enter Number : 4
Table of 4
2 x 1 = 4
2 x 2 = 8
2 x 3 = 12
2 x 4 = 16
2 x 5 = 20
2 x 6 = 24
2 x 7 = 28
2 x 8 = 32
2 x 9 = 36
2 x 10 = 40
```

8. Creating the process in C++:

Code:

```
#include<iostream>

#include<unistd.h>

#include<sys/wait.h>

using namespace std;

int main(){

    int id;

    id= fork (); //create a process

    if(id<0){ // Error creating process

        cout<<"Can't create a file (Error)"<<endl;

        exit(-1);

    }

    if (id!=0){

        cout<<"Parent Process with id :"<<getpid()<<" and Parent id : "
        "<<getppid()<<endl;

    }else{

        cout<<"Child Process with id :"<<getpid()<<" and Parent id : "
        "<<getppid()<<endl;

    }

    return 0;

}
```


Output:

```
(naeem@DESKTOP-ET0HEQ6) - [~]
$ g++ process_creation.cpp -o process_creation

(naeem@DESKTOP-ET0HEQ6) - [~]
$ ./process_creation
Parent Process with id :198 and Parent id : 10
Child Process with id :199 and Parent id : 198

(naeem@DESKTOP-ET0HEQ6) - [~]
$
```

9. Creating the n child process in C++:

Code:

```
#include <iostream>
#include <unistd.h>
#include <sys/wait.h>
using namespace std;
int main(){
    int n;
    cout << "Enter the no Child process : "; cin >> n;
    int id;
    for(int j = 1; j <= n; j++){
        if(fork() == 0){
            cout << "Child process with id : " << getpid() << " with parent id : " << getppid() << endl;
            exit(0); } }
    wait(NULL);
    return 0;
}
```

Output:

```
(naeem@DESKTOP-ET0HEQ6)-[~]
$ nano n_process.c++

(naeem@DESKTOP-ET0HEQ6)-[~]
$ g++ n_process.c++ -o n_process

(naeem@DESKTOP-ET0HEQ6)-[~]
$ ./n_process
Enter the no Child process : 4
Child process with id :208 with parent id : 207
Child process with id :209 with parent id : 207
Child process with id :210 with parent id : 207
Child process with id :211 with parent id : 207

(naeem@DESKTOP-ET0HEQ6)-[~]
$
```

Show in Linux (Top):

```
naeem@DESKTOP-ET0HEQ6: ~
top - 22:32:54 up 2 min, 0 users, load average: 0.52, 0.58, 0.59
Tasks: 17 total, 1 running, 16 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6.1 us, 9.5 sy, 0.0 ni, 83.8 id, 0.0 wa, 0.6 hi, 0.0 si, 0.0 st
MiB Mem : 8067.1 total, 4079.2 free, 3763.9 used, 224.0 buff/cache
MiB Swap: 15193.2 total, 15026.5 free, 166.7 used. 4172.6 avail Mem

  PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
  38 naeem     20   0   18188   2136   1528 R   0.3   0.0   0:00.07 top
    1 root       20   0    8940    332    284 S   0.0   0.0   0:00.14 init
    9 root       20   0    8940    228    180 S   0.0   0.0   0:00.01 init
   10 naeem     20   0   15468   2596   2500 S   0.0   0.0   0:00.08 bash
   23 root       20   0    8940    228    180 S   0.0   0.0   0:00.01 init
   24 naeem     20   0   15468   2692   2600 S   0.0   0.0   0:00.05 bash
   40 naeem     20   0   13824   1616   1588 S   0.0   0.0   0:00.02 n_process
   41 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   42 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   43 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   44 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   45 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   46 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   47 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   48 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   49 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
   50 naeem     20   0   13824    180    100 S   0.0   0.0   0:00.00 n_process
```

10. Creating the threads in C++:

In this program thread is created and sum of first 100 integers are calculated in its function.

Code:

```
#include<iostream>

#include<unistd.h>

#include<pthread.h>

using namespace std;

void *thread_fun(void *arg);

int main(){

pthread_t mythread; //declare thread

pthread_create(&mythread,NULL,thread_fun,NULL); //create thread

pthread_join(mythread,NULL); //wait for completing thread

cout<<"Now in the main Program !"<<endl;

sleep(1);

cout<<"Ending main"<<endl;

sleep(1);

return 0;

}

void *thread_fun(void *arg){

cout<<"This is Thread"<<endl;

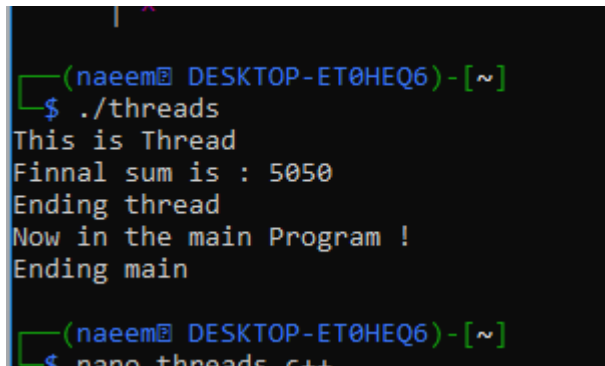
sleep(1);

int sum=0;

for (int i=1;i<=100;i++){
```

```
    sum+=i;
}
cout<<"Finnal sum is : "<<sum<<endl;
cout<<"Ending thread"<<endl;
}
```

Output:



```
(naeem@DESKTOP-ET0HEQ6)~$ ./threads
This is Thread
Finnal sum is : 5050
Ending thread
Now in the main Program !
Ending main
(naeem@DESKTOP-ET0HEQ6)~$ nano threads.cpp
```

11. To Create a Threads in LINUX and Assign the cores to threads:

In this program 2 threads are created and write the CPU intensive Function that is performed by the threads. Then we assign the different cores to the threads.

Thread 1 to logical processor 1

Thread 2 to logical Processor 2

Script:

```
// thread_core_assign.cpp *(Program file name)

#define _GNU_SOURCE

#include<iostream>

#include<pthread.h>

#include<math.h>
```

```
#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

using namespace std;

#define handle_error_en(en,msg)\
    do {errno = en; perror(msg);exit(EXIT_FAILURE);} while (0);

void *threadFunction(void *args){ // Declare the Time taking function for the Thread.

    float x = 1.5f;

    while(1){

        x *= sin(x)/atan(x)*tanh(x)*sqrt(x);

    }

}

int main(){

    cpu_set_t cpuset1;

    CPU_ZERO(&cpuset1);

    CPU_SET(1,&cpuset1);

    cpu_set_t cpuset2;

    CPU_ZERO(&cpuset2);

    CPU_SET(2,&cpuset2);

    pthread_t t1;

    pthread_t t2;

    int t1_op = pthread_create(&t1,NULL,threadFunction,NULL);

    int t2_op = pthread_create(&t2,NULL,threadFunction,NULL);
```

```

int s1 = pthread_setaffinity_np(t1,sizeof(cpu_set_t),&cpuset1);

if(s1 != 0){

    handle_error_en (s1,"pthread_setaffinity_np,s1");

}

int s2 = pthread_setaffinity_np(t1,sizeof(cpu_set_t),&cpuset2);

if(s2 != 0){

    handle_error_en (s2,"pthread_setaffinity_np,s2");

}

pthread_join(t1,NULL);

pthread_join(t2,NULL);

return 0;

}

```

Assign the Core to the Threads (Htop_preview):

naeem@DESKTOP-ET0HEQ6: ~

```

0[|||||] 72.3% Tasks: 9, 3 thr, 0 kthr; 3 running
1[|||||] 78.1% Load average: 0.52 0.58 0.59
2[|||||] 76.6% Uptime: 00:42:17
3[|||||] 94.9%
Mem[|||||] 4.58G/7.88G
Swp[|||||] 59.4M/14.8G

```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	8940	328	284	S	0.0	0.0	0:00.09	/init
8	root	20	0	8940	328	284	S	0.0	0.0	0:00.00	/init
23	naeem	20	0	13228	2384	1928	T	0.0	0.0	2:23.98	htop
42	naeem	20	0	13228	2404	1940	T	0.0	0.0	0:00.01	htop
46	root	20	0	8940	224	184	S	0.0	0.0	0:00.00	/init
47	naeem	20	0	15468	2704	2616	S	0.0	0.0	0:00.15	-bash
101	naeem	20	0	30216	1620	1376	S	177.3	0.0	1:05.26	./thread_core assign
102	naeem	20	0	30216	1620	1376	R	87.7	0.0	0:32.65	./thread_core assign
103	naeem	20	0	30216	1620	1376	R	89.0	0.0	0:32.60	./thread_core assign
62	root	20	0	8940	224	184	S	0.0	0.0	0:00.00	/init
63	naeem	20	0	15468	2584	2492	S	0.0	0.0	0:00.09	-bash
76	naeem	20	0	13228	2472	1704	R	0.0	0.0	0:06.77	htop

F1Help F2Setup F3Search F4Filter F5List F6SortBy F7Nice -F8Nice +F9Kill F10Quit

12. Process Scheduling Algorithms in C++:

(SJF, RR, Priority Scheduling)

Code:

```
#include<iostream>

#include<string.h>

#include <conio.h>

#include <iomanip>

using namespace std;

struct Node{

    string process_name;

    int burst_time;

    int priority;

    Node *next;

    Node(string p_n,int b,int p){

        process_name = p_n;

        burst_time = b;

        priority=p;

        next =NULL;

    }

};

class Queue{

    Node *rear;

    Node *front;

    int size;

    void line(){
```

```
        for(int a=1;a<=100;a++){
            cout <<"_";}cout<<endl;}

void line(int n){
    for(int a=1;a<=n;a++){
        cout <<"_";}cout<<endl;}

string process_name_value(){
    string name; cout<< "Enter the Process Name : ";getline(cin,name);
    return name; }

int brust_time_value(){
    int v; cout<< "Enter the brust time  : ";cin>>v;cin.ignore();
    return v; }

int priority_value(){
    int v; cout<< "Enter the priority    : ";cin>>v;cin.ignore();
    return v; }

Node* swap(Node* ptr1, Node* ptr2){
    struct Node* tmp = ptr2->next;ptr2->next = ptr1;ptr1->next = tmp;return ptr2;}

void bubbleSort_Brust_time(Node **head){
    Node** h;int i, j, swapped;

    for (i = 0; i <= size; i++){
        h = head;swapped = 0;
        for (j = 0; j < size - i - 1; j++){
            Node* p1 = *h;Node* p2 = p1->next;
            if (p1->brust_time > p2->brust_time){
                *h = swap(p1, p2);swapped = 1;}h = &(*h)->next;}

        if (swapped == 0) break;}
```



```
void bubbleSort_Priority(Node **head){

    Node** h; int i, j, swapped;

    for (i = 0; i <= size; i++){

        h = head;swapped = 0;

        for (j = 0; j < size - i - 1; j++){

            Node* p1 = *h; Node* p2 = p1->next;

            if (p1->priority > p2->priority){

                /* update the link after swapping */

                *h = swap(p1, p2); swapped = 1;} h = &(*h)->next;}

            if (swapped == 0)

                break;}

}

int original_brust_time(string name,Queue q){

    Node* temp = q.front;

    while(temp!=NULL){

        if(temp->process_name==name){

            return temp->brust_time;}

        temp = temp->next;}}

public :

    Queue(){

        front = NULL;

        rear = NULL;

        size = 0;}

    void enqueue(){

        if(front==NULL){

            front = new Node(process_name_value(),brust_time_value(),priority_value());
```

```
        size++;rear = front;

        }else{Node *temp = new
Node(process_name_value(),brust_time_value(),priority_value());

        size++;rear->next = temp;rear = temp;

    }}

void dequeue(){

    if(front==NULL){

        cout<< "Under Flow Queue"<<endl;

    }else{

        //cout << "Dequeue Value is : "<<front->process_name<<endl;

        front = front->next;

        size--;

        if(front == NULL){

            rear = NULL;}}

void peek(){

    if (front==NULL){

        cout << "Empty Queue "<<endl;

    }else{

        cout<< "Peek Value is : "<<front->process_name<<endl;}}

void display(){

    if(front==NULL){

        cout <<"Empty Queue"<<endl;return ;}

    cout << "Process Linked List Queue"<<endl;

    Node *temp = front;

    line();
```

```

cout<<setiosflags(ios::left)<<setw(20)<<"Process's
Name"<<setiosflags(ios::left)<<setw(15)<<"Brust
Time"<<setiosflags(ios::left)<<setw(15)<<"Priority"<<endl; line();

        while(temp!=NULL){

cout <<setiosflags(ios::left)<<setw(20)<<temp-
>process_name<<setiosflags(ios::left)<<setw(15)<<temp-
>brust_time<<setiosflags(ios::left)<<setw(15)<<temp->priority<<endl;

                temp = temp->next;}

        line();}

void copy_linkded_list(Queue q){

Node* tempf = q.front;

while(tempf!=NULL){

if(front==NULL){

front = new Node(tempf->process_name,tempf->brust_time,tempf->priority);

size++;rear = front;

}else{

Node *temp = new Node(tempf->process_name,tempf->brust_time,tempf->priority);

size++;rear->next = temp;rear = temp;}tempf= tempf->next;}}

float SJF(){

if(front==NULL){

cout <<"Empty Queue"<<endl;return 0 ;}

bubbleSort_Brust_time(&front);

int t = 0,newt=0;

int avg_wait_time=0,no_of_process=0;

Node *temp = front;cout<<endl;

cout<<"-----SJF Scheduling Chart-----"<<endl<<endl;line();

while(temp!=NULL){

```

```

        avg_wait_time+=t;

        newt += temp->brust_time;

cout <<"| "<<t<<" "<<setiosflags(ios::left)<<setw(temp-
>brust_time/2)<<" "<<setiosflags(ios::left)<<setw(temp->brust_time)<<temp-
>process_name <<" "<<newt<<"|";

        t=newt;temp = temp->next;no_of_process++; }

        cout<<endl;line(newt+70);cout<<endl;

        cout<<"Avarage Wait Time is :
"<<(float)avg_wait_time/no_of_process<<endl<<endl;

        return (float)avg_wait_time/no_of_process;}

float Priority(){

        if(front==NULL){

                cout <<"Empty Queue"<<endl;

                return 0 ;}

        bubbleSort_Priority(&front);

        int t = 0,newt=0;

        int avg_wait_time=0,no_of_process=0;

        Node *temp = front;

        cout<<endl;cout<<"-----Priority Scheduling Chart-----
"<<endl<<endl;line();

        while(temp!=NULL || size==0){

                avg_wait_time+=t;

                newt += temp->brust_time;

                cout <<"| "<<t<<" "<<setiosflags(ios::left)<<setw(temp-
>brust_time/2)<<" "<<
                                <<setiosflags(ios::left)<<setw(temp-
>brust_time)<<temp->process_name

```

```

        <<" "<<newt<<"|";

        t=newt;temp = temp->next;no_of_process++;  }

        cout<<endl;line(newt+70);cout<<endl;

        cout<<"Avarage Wait Time is :
"<<(float)avg_wait_time/no_of_process<<endl<<endl;

        return (float) avg_wait_time/no_of_process;}

float RR(int quantum,Queue q){

    if(front==NULL){

        cout <<"Empty Queue"<<endl;return 0;}

    int t = 0,newt=0,no_process=size;

    int avg_wait_time=0;cout<<endl;

    cout<<"-----Round Robin Scheduling Chart-----"<<endl<<endl;line();

    while(front!=NULL){

        if(front->brust_time<=quantum){

            newt += front->brust_time;

            cout <<"| "<<t<<" "<<setiosflags(ios::left)<<setw(quantum/2)<<" "

                <<setiosflags(ios::left)<<setw(quantum)<<front-
>process_name <<" "<<newt<<"|";

                avg_wait_time += newt - orignal_brust_time(front-
>process_name,q);

                t=newt; dequeue();

            }else{

                newt += quantum;

                cout <<"| "<<t<<" "<<setiosflags(ios::left)<<setw(quantum/2)<<" "

                    <<setiosflags(ios::left)<<setw(quantum)<<front-
>process_name <<" "<<newt<<"|";

                    t=newt;

```

```

        string name = front->process_name;

        int b_t = front->brust_time-quantum;

        int p = front->priority; dequeue();

        if(front==NULL){

            front = new Node(name,b_t,p);

            size++;

            rear = front;

        }else{

            Node *temp = new Node(name,b_t,p);

            size++;

            rear->next = temp;

            rear = temp;

        }}}

        cout<<endl;line(newt+70);cout<<endl;

        cout<<"Avarage Wait Time is : "<<(float)avg_wait_time/no_process<<endl<<endl;

        return (float) avg_wait_time/no_process;

    });

int main(){

    int n,quantum;

    cout<<"Enter the no of Programs : ";cin>>n;cin.ignore();

    cout<<"Enter the time Quantum   : ";cin>>quantum;cin.ignore();

    Queue q ;

    for (int i=1;i<=n;i++){

        cout<<"Enter the "<<i<<" Process Data ! "<<endl;

        q.enqueue();cout<<endl;}

```

```
cout<<endl;q.display();

Queue q1 , q2, q3;

q1.copy_linkded_list(q);q2.copy_linkded_list(q);q3.copy_linkded_list(q);

float SJF = q1.SJF();

float Pri = q2.Priority();

float RR = q3.RR(quantum,q);

cout<<endl;

if(SJF<Pri){

    if(SJF<RR){

cout<<"Shortest Job First (SJF) is the best algorithm for these Process Scheduling!
"<<endl;

        }else{

cout<<"Round Robin (RR) is the best algorithm for these Process Scheduling! "<<endl;}

        }else{

            if(Pri<RR){

cout<<"Priority Scheduling is the best algorithm for these Process Scheduling! "<<endl;

                }else{

cout<<"Round Robin (RR) is the best algorithm for these Process Scheduling! "<<endl;

                    }

                }

            return 0;    }
```

Output Example 1:

```

C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Scheduling Algorithms.exe

Process Linked List Queue

Process's Name    Brust Time    Priority
-----
P1                6             2
P2                8             1
P3                7             1
P4                3             3

-----SJF Scheduling Chart-----

|0, P4 ,3||3, P1 ,9||9, P3 ,16||16, P2 ,24|

Avarage Wait Time is : 7

-----Priority Scheduling Chart-----

|0, P2 ,8||8, P3 ,15||15, P1 ,21||21, P4 ,24|

Avarage Wait Time is : 11

-----Round Robin Scheduling Chart-----

|0, P1 ,3||3, P2 ,6||6, P3 ,9||9, P4 ,12||12, P1 ,15||15, P2 ,18||18, P3 ,21||21, P2 ,23||23, P3 ,24|

Avarage Wait Time is : 12.5

Shortest Job First (SJF) is the best algorithm for these Process Scheduling!

-----
Process exited after 64.52 seconds with return value 0
Press any key to continue . . .

```

Output Example 2:


```

C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Scheduling Algorithms.exe
Enter the no of Programs : 3
Enter the time Quantum : 4
Enter the 1 Process Data !
Enter the Process Name : P1
Enter the brust time : 24
Enter the priority : 1

Enter the 2 Process Data !
Enter the Process Name : P2
Enter the brust time : 3
Enter the priority : 3

Enter the 3 Process Data !
Enter the Process Name : P3
Enter the brust time : 3
Enter the priority : 2

Process Linked List Queue

Process's Name    Brust Time    Priority
-----
P1                24            1
P2                3            3
P3                3            2

-----SJF Scheduling Chart-----

|0, P2 ,3||3, P3 ,6||6, P1 ,30|

Avarage Wait Time is : 3

-----Priority Scheduling Chart-----

|0, P1 ,24||24, P3 ,27||27, P2 ,30|

Avarage Wait Time is : 17

-----Round Robin Scheduling Chart-----

|0, P1 ,4||4, P2 ,7||7, P3 ,10||10, P1 ,14||14, P1 ,18||18, P1 ,22||22, P1 ,26||26, P1 ,30|

Avarage Wait Time is : 5.66667

Shortest Job First (SJF) is the best algorithm for these Process Scheduling!

```

Output Example 3:

```
C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Scheduling Algorithms.exe

Process Linked List Queue

Process's Name    Brust Time    Priority
-----
P1                10           3
P2                1           1
P3                2           4
P4                1           5
P5                5           2

-----SJF Scheduling Chart-----

|0,P2,1||1,P4,2||2, P3,4||4, P5 ,9||9, P1 ,19|

Avarage Wait Time is : 3.2

-----Priority Scheduling Chart-----

|0,P2,1||1, P5 ,6||6, P1 ,16||16, P3,18||18,P4,19|

Avarage Wait Time is : 8.2

-----Round Robin Scheduling Chart-----

|0, P1 ,5||5, P2 ,6||6, P3 ,8||8, P4 ,9||9, P5 ,14||14, P1 ,19|

Avarage Wait Time is : 7.4

Shortest Job First (SJF) is the best algorithm for these Process Scheduling!

-----
Process exited after 70.34 seconds with return value 0
Press any key to continue . . .
```

13. Sub-netting Calculator using Python

Code:

```
# For colored output
CRED = '\033[91m'
CEND = '\033[0m'

# Function to convert Decimal to Binary
def dtb(n):
    return format(n, '08b')

# Function to convert from CIDR Notation to Subnet mask
def cidr_to_subnet(CIDR):
    counter = 0
    subnet=[]
    while (CIDR>8):
        CIDR = CIDR-8
        counter +=1
    for i in range(4):
        if i < counter:
            subnet.append(255)
        elif i == counter:
            subnet.append(256-pow(2, (8-CIDR)))
        else:
            subnet.append(0)
    return subnet

ip=[] # List to store ip address
check_ip=True # Variable to manage loop over ip address
while check_ip:
    try:
        # Ask user for ip address and then split it based on the '.' and put
        # it in a list
        ip = list(map(int, input("Please, Enter an IP Address: ").split('.')))
        check_ip=False
        # error handling for ip address
        if len(ip) != 4 :
            # Check for X.X.X.X Format
            print(CRED + "Error: IP Address supported format X.X.X.X" + CEND)
            check_ip=True
            continue
        for i in ip:
            # Check if the any digit exceeds 255
            if i > 255:
                print(CRED + "Error: IP Address Range: (0-255)" + CEND)
                check_ip=True
                break
            continue
    except ValueError:
        print(CRED + "Error: Only integers are allowed" + CEND)
        check_ip=True
        continue
    except KeyboardInterrupt:
        print()
        print(CRED + "IP Calculator has been terminated." + CEND)
```

```

        exit(0)
    except:
        print(CRED + "An error has occurred." + CEND)
        check_ip=True
        continue

subnet=[] # List to store subnet mask
check_subnet=True # Variable to manage loop over subnet mask
while(check_subnet):
    try:
        # Ask user for subnet mask with CIDR or full subnet mask
        CIDR=input("Please, Enter Subnet Mask: ")
        # CIDR Notation code
        if CIDR[0] == '/':
            CIDR=int(CIDR[1:])
            # if CIDR not in range (8-30) raise an error invalid subnet mask
            if not CIDR in range(8,31):
                print(CRED + "Invalid shorthand notation. Must be in the
range of 8-30." + CEND)
                check_subnet=True
                continue
            else:
                # else send cidr to the conversion fn to obtain full subnet
mask
                subnet=cidr_to_subnet(CIDR)
                # set check_subnet to false to exit while loop
                check_subnet=False
        else:
            # Ask user for subnet mask and then split it based on the '.' and
put it in a list
            subnet = list(map(int,CIDR.split('.')))
            # check if the first value not equal 255
            if subnet[0] != 255:
                print(CRED + "Invalid subnet mask. The lowest allowed mask
must be 255.0.0.0" + CEND)
                check_subnet=True
                continue
            # set check_subnet to false to exit while loop
            check_subnet=False
            # Check if user has inputted the full length of the subnet mask
            if len(subnet) != 4:
                print(CRED + "Error: Subnet Mask supported format X.X.X.X" +
CEND)
                check_subnet=True
                continue
            # check if the rest of the digits exceeds 255 skip the first digit we
have already checked above
            for j in subnet[1:] :
                # Check if the digits exceeds 255
                if j > 255:
                    print(CRED + "Error: Subnet Mask Range: (0-255)" + CEND)
                    check_subnet=True
                    break
                # Check for invalid subnetmask (continuous binary ones)
                if '01' in dtb(j):
                    print(CRED + "Error: Invalid subnet mask, it must contain
continuous ones." + CEND)

```

```
        check_subnet=True
        break
    continue
except ValueError:
    print(CRED + "Error: Only integers are allowed." + CEND)
    check_subnet=True
    continue
except KeyboardInterrupt:
    print()
    print(CRED + "IP Calculator has been terminated." + CEND)
    exit(0)
except:
    print(CRED + "An error has occurred." + CEND)
    check_subnet=True
    continue

subnet_ones=0 # count no of ones in subnet mask
t_no_hosts=0 # calculate total number of hosts
network_id=[] # list to store network id
broadcast_address=[0,0,0,0] # list to store broadcast address

for i in range(4):

    # count the number of ones in the subnet mask
    subnet_ones = subnet_ones + len(dtb(subnet[i]).strip('0'))

    # perform a logical and operation between ip and subnet mask to get the
    network id
    network_id.append(int(ip[i]) & int(subnet[i]))

    # Calculate Broadcast Address
    if subnet[i] == 255 :
        broadcast_address[i] = network_id[i]
    else:
        broadcast_address[i] = 255 - subnet[i] + network_id[i]

print("-----")

# Print the CIDR Notation for the given subnet mask
print("CIDR: /",subnet_ones)

# Print Network ID
print("Network ID:", '.'.join(map(str, network_id)))

# Calculate Total Number of Hosts in the subnet
t_no_hosts = pow(2,32-subnet_ones) - (2)

# Print First Host Address
first_id = network_id.copy()
first_id[3] += 1
print("First Host Address:", '.'.join(map(str, first_id)))

# Print Last Host Address
last_address = broadcast_address.copy()
last_address[3] -= 1
print("Last Host Address:", '.'.join(map(str, last_address)))
```

```
# Print Broadcast Address
print("Broadcast Address:", '.'.join(map(str, broadcast_address)))

# Print Total Number of Hosts
print("Total number of Hosts =", t_no_hosts)

# Exit successfully
exit(0)
```

Output:

20 bits subnet mask

```
>>> |===== RESTART: C:/Users/NAEEM UR RAHMAN/OneDrive/Desktop/Subnetting Code.py =====
Please, Enter an IP Address: 172.17.99.0
Please, Enter Subnet Mask: 255.255.240.0
-----
CIDR: / 20
Network ID: 172.17.96.0
First Host Address: 172.17.96.1
Last Host Address: 172.17.111.254
Broadcast Address: 172.17.111.255
Total number of Hosts = 4094
>>> |
```

24 bits subnet mask

```
>>> |===== RESTART: C:/Users/NAEEM UR RAHMAN/OneDrive/Desktop/Subnetting Code.py =====
Please, Enter an IP Address: 172.17.99.0
Please, Enter Subnet Mask: 255.255.255.0
-----
CIDR: / 24
Network ID: 172.17.99.0
First Host Address: 172.17.99.1
Last Host Address: 172.17.99.254
Broadcast Address: 172.17.99.255
Total number of Hosts = 254
>>> |
```

14. Bankers Algorithms in C++:

Code:

```
#include<iostream>
#include<iostream>
#include <iomanip>
using namespace std;
class Process{
    int  total_resources;
        int  total_processes;

        int  *max_resources;
        int  *allocated;
        int  *available;

        bool *running_processes;

        int  allocate_table[30][20];
        int  max_claim_process[30][20];

        void get_max_resoures(){
            cout<<"Enter the Total MAX resoures Instances"<<endl;
            for(int i=0;i<total_resources;i++){
                cout<<"Enter R"<<i+1<<" : ";cin>>max_resources[i];
            }
        }
    }
```

```
        cout<<endl;

    }

    void print_max_resources(){

        cout<<"Total MAX resoures Instances Are : "<<endl;

        for(int i=0;i<total_resources;i++){

            cout<<"R"<<i+1<<" :
"<<setiosflags(ios::left)<<setw(10)<<max_resources[i];

        }

        cout<<endl;

        line(70);

    }

    void line(int n){

        for(int a=1;a<=n;a++){

            cout <<" _ ";

        }

        cout<<endl;

    }

    void status_running_at_start(){

        for(int i=0 ;i<total_prcesses;i++){

            running_processes[i] = true;

        }

    }

    void get_allocate_table(){

        cout<<"Enter the allocation table of the Processes !"<<endl;

        for(int i=0;i<total_prcesses;i++){
```



```

        cout<<"Process "<<i+1<<endl;

        for(int j=0;j<total_resources;j++){

            cout<<"R"<<j+1<<" Allocated : ";cin>>allocate_table[i][j];

        }

        cout<<endl;

    }

}

void print_allocate_table(){

    cout<<"Allocated Table "<<endl;

    line(70);

    cout<<setiosflags(ios::left)<<setw(12)<<"Process";

    for(int p=1;p<=total_resources;p++){

        cout<<"R"<<setiosflags(ios::left)<<setw(12)<<p;

    }

    cout<<endl;

    line(70);

    for(int i=0;i<total_prcesses;i++){

        cout<<setiosflags(ios::left)<<setw(12)<<i+1;

        for(int j=0;j<total_resources;j++){

            cout<<setiosflags(ios::left)<<setw(12)<<allocate_table[i][j];

        }

        cout<<endl;

    }

    line(70);

```

```
    }

    void get_max_claim_table(){

        cout<<"Enter the MAX Claim table of the Processes !"<<endl;

        for(int i=0;i<total_prcesses;i++){

            cout<<"Process "<<i+1<<endl;

            for(int j=0;j<total_resourses;j++){

                cout<<"R"<<j+1<<" MAX Claim :
";cin>>max_claim_process[i][j];

            }

            cout<<endl;

        }

    }

    void print_max_claim_table(){

        cout<<"MAX Claim Table "<<endl;

        line(70);

        cout<<setiosflags(ios::left)<<setw(12)<<"Process";

        for(int p=1;p<=total_resourses;p++){

            cout<<"R"<<setiosflags(ios::left)<<setw(12)<<p;

        }

        cout<<endl;

        line(70);

        for(int i=0;i<total_prcesses;i++){

            cout<<setiosflags(ios::left)<<setw(12)<<i+1;

            for(int j=0;j<total_resourses;j++){
```

```

        cout<<setiosflags(ios::left)<<setw(12)<<max_claim_process[i][j];

        }

        cout<<endl;

    }

    line(70);

}

void get_allocated_resources(){

    for(int i=0;i<total_prcesses;i++){

        for(int j=0;j<total_resources;j++){

            allocated[j] += allocate_table[i][j];

        }

    }

}

void print_allocated_resources(){

    cout<<"Allocated Resource are : "<<endl;

    for(int i=0;i<total_resources;i++){

        cout<<"R"<<i+1<<" :

"<<setiosflags(ios::left)<<setw(10)<<allocated[i];

    }

    cout<<endl;

    line(70);

}

void get_available_resources(){

    for(int i=0;i<total_prcesses;i++){

```

```

        for(int j=0;j<total_resources;j++){
            available[j] = max_resources[j] - allocated[j];
        }
    }
}

void print_available_resoures(){
    cout<<"Available Resource are : "<<endl;
    for(int i=0;i<total_resources;i++){
        cout<<"R"<<i+1<<" :
"<<setiosflags(ios::left)<<setw(10)<<available[i];
    }
    cout<<endl;
    line(70);
}

void set_array(int *value,int size){
    for(int i=0;i<size;i++){
        value[i]=0;
    }
}

void check_save(){
    int safe,exec,count=total_prcesses;
    while (count != 0){
safe = 0;
    for (int i = 0; i < total_prcesses; i++){
        if (running_processes[i]){

```

```

        exec = 1;

        for (int j = 0; j < total_resources; j++){

            if (max_claim_process[i][j] - allocate_table[i][j] > available[j]){

                exec = 0;

                break;

            }

        }

        if (exec){

                                cout<<"Process "<< i + 1<<" is
executing"<<endl;

                running_processes[i] = false;

                count--;

                safe = 1;

                for (int j = 0; j < total_resources; j++){

                                available[j] += allocate_table[i][j];

                }

                break;

        }

    }

    if (!safe) {

                                cout<<"The process is in unsafe state."<<endl;

        break;
    }

```

```
    }  
    else {  
  
        cout<<"Your process is in safe state"<<endl;  
        cout<<"Available vector :";  
  
        for (int i = 0; i < total_resources; i++)  
        {  
            cout<< available[i] <<" ";  
        }  
        cout<<endl<<endl;  
    }  
}  
  
}  
  
public :  
  
Process(int resources,int processes){  
    total_prcesses = processes;  
    total_resources = resources;  
    max_resources = new int [total_resources];  
    allocated    = new int [total_resources];  
    available    = new int [total_resources];  
    set_array(allocated,total_resources);  
    set_array(available,total_resources);  
  
    get_max_resoures();  
}
```

```

        running_processes = new bool[total_processes];
        status_running_at_start();

        get_allocate_table();
        get_max_claim_table();
        get_allocated_resources();
        get_available_resources();

        print_allocate_table();
        print_max_claim_table();
        print_max_resources();
        print_allocated_resources();
        print_available_resources();
        check_save();
    }
};

int main(){
    int processes,resources;

    cout<<"Enter Number of Processes : ";cin>>processes;
    cout<<"Enter Number of Resources : ";cin>>resources;

    Process p = Process(resources,processes);

    return 0;
}

```

Output Example 1:

```

C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Bankers Algorithm.exe
Enter Number of Processes : 4
Enter Number of Resources : 2
Enter the Total MAX resoures Instances
Enter R1 : 12
Enter R2 : 12

Enter the allocation table of the Processes !
Process 1
R1 Allocated : 1
R2 Allocated : 1

Process 2
R1 Allocated : 3
R2 Allocated : 3

Process 3
R1 Allocated : 1
R2 Allocated : 2

Process 4
R1 Allocated : 1
R2 Allocated : 1

Enter the MAX Claim table of the Processes !
Process 1
R1 MAX Claim : 3
R2 MAX Claim : 3

Process 2
R1 MAX Claim : 4
R2 MAX Claim : 5

Process 3
R1 MAX Claim : 3
R2 MAX Claim : 3

Process 4
R1 MAX Claim : 3
R2 MAX Claim : 2
    
```



```

C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Bankers Algorithm.exe

Allocated Table
Process    R1    R2
1          1      1
2          3      3
3          1      2
4          1      1

MAX Claim Table
Process    R1    R2
1          3      3
2          4      5
3          3      3
4          3      2

Total MAX resources Instances Are :
R1 : 12    R2 : 12

Allocated Resource are :
R1 : 6     R2 : 7

Available Resource are :
R1 : 6     R2 : 5

Process 1 is executing
Your process is in safe state
Available vector :7 , 6 ,
Process 2 is executing
Your process is in safe state
Available vector :10 , 9 ,
Process 3 is executing
Your process is in safe state
Available vector :11 , 11 ,
Process 4 is executing
Your process is in safe state
Available vector :12 , 12 ,

```

Output Example 2:

C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Bankers Algorithm.exe

```
Enter Number of Processes : 5
Enter Number of Resources : 3
Enter the Total MAX resources Instances
Enter R1 : 5
Enter R2 : 5
Enter R3 : 5

Enter the allocation table of the Processes !
Process 1
R1 Allocated : 1
R2 Allocated : 0
R3 Allocated : 0

Process 2
R1 Allocated : 2
R2 Allocated : 2
R3 Allocated : 0

Process 3
R1 Allocated : 0
R2 Allocated : 0
R3 Allocated : 2

Process 4
R1 Allocated : 0
R2 Allocated : 0
R3 Allocated : 0

Process 5
R1 Allocated : 1
R2 Allocated : 1
R3 Allocated : 1

Enter the MAX Claim table of the Processes !
Process 1
R1 MAX Claim : 3
R2 MAX Claim : 3
R3 MAX Claim : 2

Process 2
R1 MAX Claim : 3
R2 MAX Claim : 3
R3 MAX Claim : 3

Process 3
R1 MAX Claim : 1
R2 MAX Claim : 1
R3 MAX Claim : 2

Process 4
R1 MAX Claim : 1
R2 MAX Claim : 1
R3 MAX Claim : 1

Process 5
R1 MAX Claim : 2
R2 MAX Claim : 2
R3 MAX Claim : 1
```

```

Allocated Table
Process      R1      R2      R3
1            1      0      0
2            2      2      0
3            0      0      2
4            0      0      0
5            1      1      1

MAX Claim Table
Process      R1      R2      R3
1            3      3      2
2            3      3      3
3            1      1      2
4            1      1      1
5            2      2      1

Total MAX resources Instances Are :
R1 : 5      R2 : 5      R3 : 5

Allocated Resource are :
R1 : 4      R2 : 3      R3 : 3

Available Resource are :
R1 : 1      R2 : 2      R3 : 2

Process 3 is executing
Your process is in safe state
Available vector :1 2 4

Process 2 is executing
Your process is in safe state
Available vector :3 4 4

Process 1 is executing
Your process is in safe state
Available vector :4 4 4

Process 4 is executing
Your process is in safe state
Available vector :4 4 4

Process 5 is executing
Your process is in safe state
Available vector :5 5 5

-----
Process exited after 226.7 seconds with return value 0
Press any key to continue . . .

```

15. Creating the Charts in Python

Script 1:

```
import numpy as np

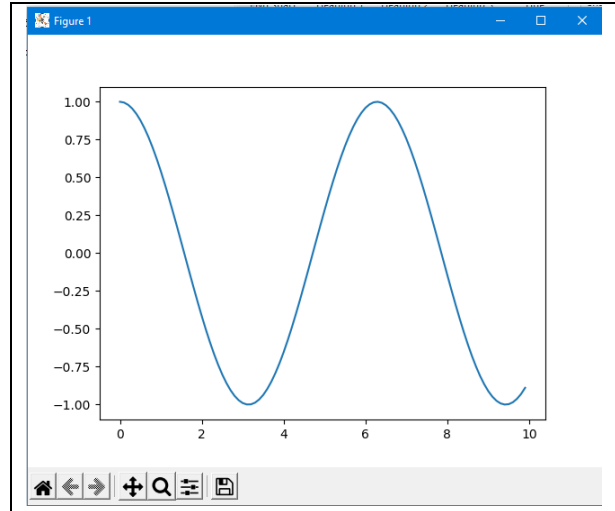
import matplotlib.pyplot as plot

time = np.arange(0, 10, 0.1);
amplitude = np.cos(time)

plot.plot(time, amplitude)

plot.show()
```

Output:



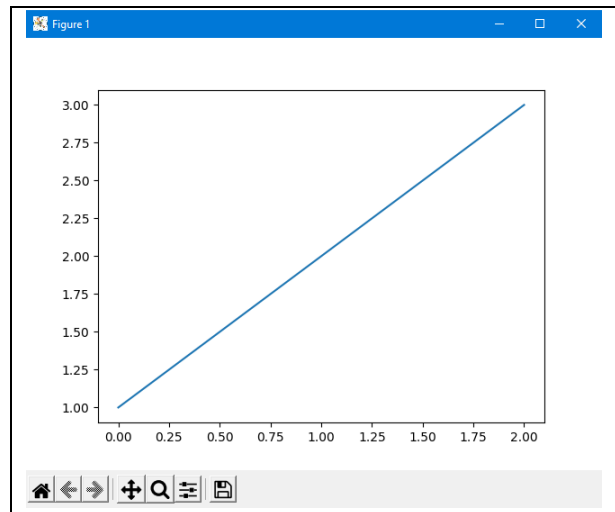
Script 2:

```
from pylab import *

plot([1,2,3])

show()
```

Output:



16. Creating the Excel Workbook in Python

Script:

```
# Writing to an excel
# sheet using Python

import xlwt

from xlwt import Workbook

# Workbook is created

wb = Workbook()

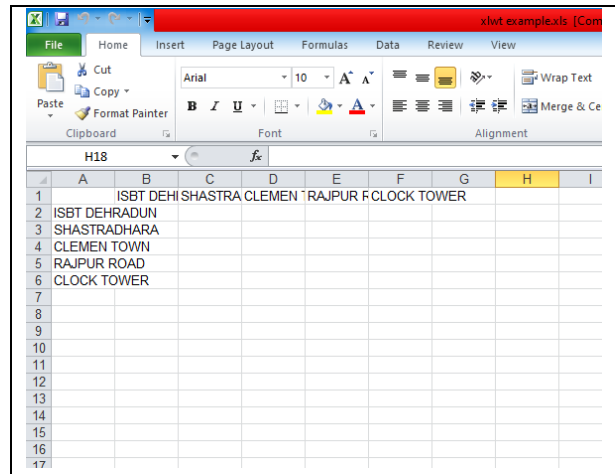
# add_sheet is used to create sheet.

sheet1 = wb.add_sheet('Sheet 1')

sheet1.write(1, 0, 'ISBT DEHRADUN')
sheet1.write(2, 0, 'SHASTRADHARA')
sheet1.write(3, 0, 'CLEMEN TOWN')
sheet1.write(4, 0, 'RAJPUR ROAD')
sheet1.write(5, 0, 'CLOCK TOWER')
sheet1.write(0, 1, 'ISBT DEHRADUN')
sheet1.write(0, 2, 'SHASTRADHARA')
sheet1.write(0, 3, 'CLEMEN TOWN')
sheet1.write(0, 4, 'RAJPUR ROAD')
sheet1.write(0, 5, 'CLOCK TOWER')
```

```
wb.save('xlwt example.xls')
```

Output:



THANK YOU!