

EXPERIMENT 10

NAME: Rahil Sharma

PRN: 18070123062

BATCH: EA-3

SUBJECT: ESRTOS

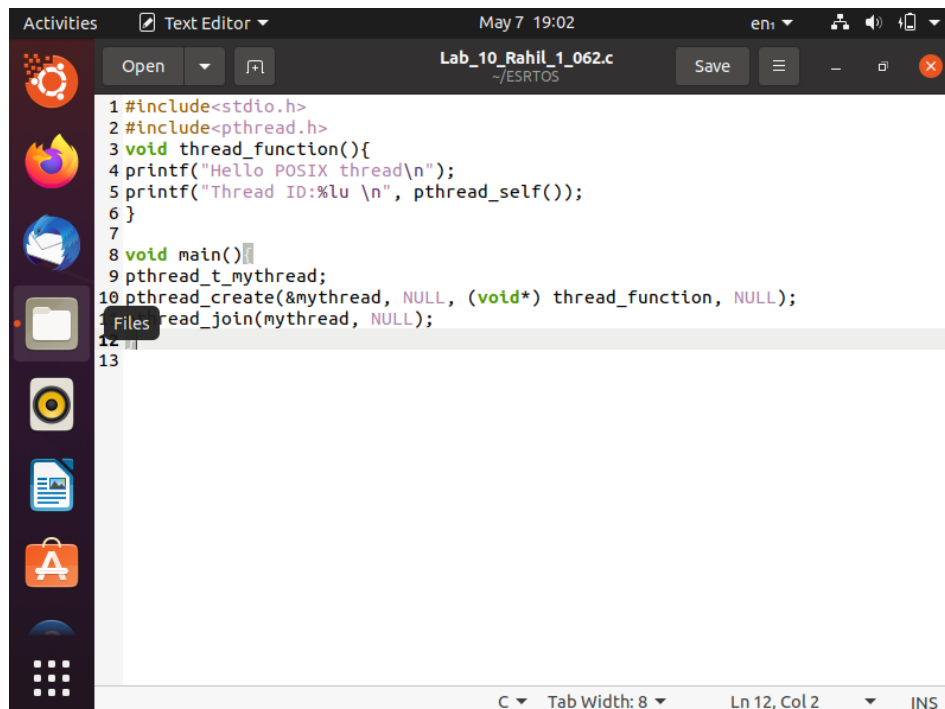
AIM: Write a C program to explain POSIX thread

i. Code 1:

Code of the Program:

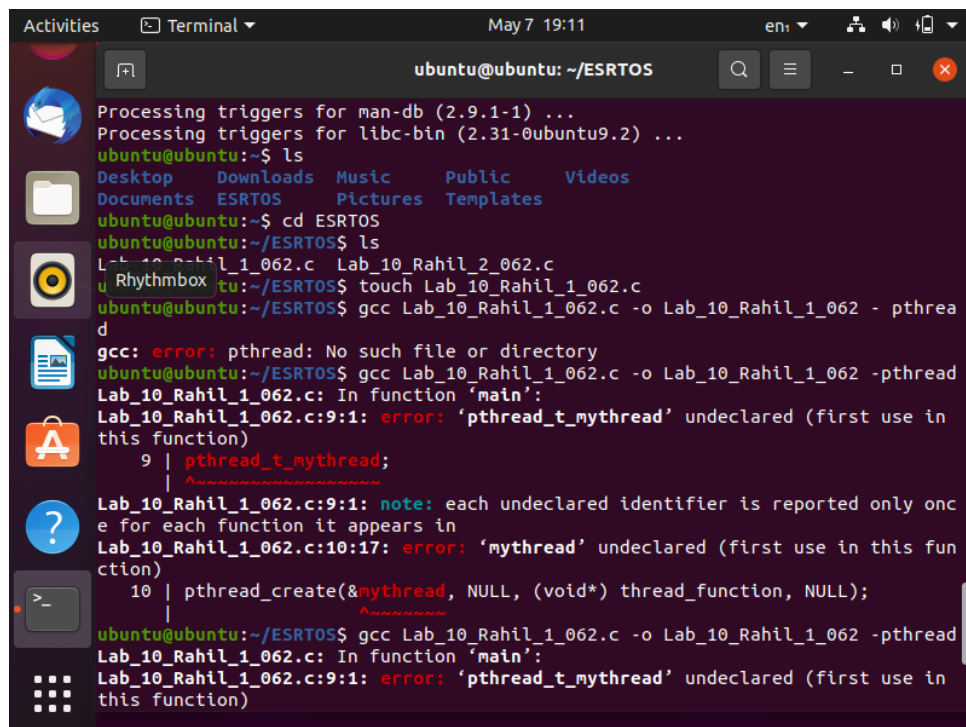
```
#include<stdio.h>
#include<pthread.h>
void thread_function(){
printf("Hello POSIX thread\n");
printf("Thread ID: %lu \n", pthread_self());
}
void main(){
pthread_t mythread;
pthread_create(&mythread, NULL, (void *) thread_function, NULL);
pthread_join(mythread, NULL);
}
```

Screenshot of the Program:



```
1 #include<stdio.h>
2 #include<pthread.h>
3 void thread_function(){
4 printf("Hello POSIX thread\n");
5 printf("Thread ID:%lu \n", pthread_self());
6 }
7
8 void main()
9 pthread_t_mythread;
10 pthread_create(&mythread, NULL, (void*) thread_function, NULL);
11 pthread_join(mythread, NULL);
12
13
```

Output of the Program:



```
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Music      Public     Videos
Documents ESRTOS    Pictures  Templates
ubuntu@ubuntu:~$ cd ESRTOS
ubuntu@ubuntu:~/ESRTOS$ ls
Lab_10_Rahil_1_062.c  Lab_10_Rahil_2_062.c
Rhythmbox
ubuntu@ubuntu:~/ESRTOS$ touch Lab_10_Rahil_1_062.c
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_1_062.c -o Lab_10_Rahil_1_062 -pthread
gcc: error: pthread: No such file or directory
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_1_062.c -o Lab_10_Rahil_1_062 -pthread
Lab_10_Rahil_1_062.c: In function 'main':
Lab_10_Rahil_1_062.c:9:1: error: 'pthread_t_mythread' undeclared (first use in this function)
   9 | pthread_t_mythread;
     | ~~~~~
Lab_10_Rahil_1_062.c:9:1: note: each undeclared identifier is reported only once for each function it appears in
Lab_10_Rahil_1_062.c:10:17: error: 'mythread' undeclared (first use in this function)
   10 | pthread_create(&mythread, NULL, (void*) thread_function, NULL);
      | ~~~~~
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_1_062.c -o Lab_10_Rahil_1_062 -pthread
Lab_10_Rahil_1_062.c: In function 'main':
Lab_10_Rahil_1_062.c:9:1: error: 'pthread_t_mythread' undeclared (first use in this function)
```

The screenshot shows a terminal window with the following content:

```
Activities Terminal May 7 19:12 en1
ubuntu@ubuntu: ~/ESRTOS

Lab_10_Rahil_1_062.c: In function 'main':
Lab_10_Rahil_1_062.c:9:1: error: 'pthread_t_mythread' undeclared (first use in this function)
  9 | pthread_t_mythread;
    | ^
Lab_10_Rahil_1_062.c:9:1: note: each undeclared identifier is reported only once for each function it appears in
Lab_10_Rahil_1_062.c:10:17: error: 'mythread' undeclared (first use in this function)
 10 | pthread_create(&mythread, NULL, (void*) thread_function, NULL);
    | ^
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_1_062.c -o Lab_10_Rahil_1_062 -pthread
Lab_10_Rahil_1_062.c: In function 'main':
Lab_10_Rahil_1_062.c:9:1: error: 'pthread_t_mythread' undeclared (first use in this function)
  9 | pthread_t_mythread;
    | ^
Lab_10_Rahil_1_062.c:9:1: note: each undeclared identifier is reported only once for each function it appears in
Lab_10_Rahil_1_062.c:10:17: error: 'mythread' undeclared (first use in this function)
 10 | pthread_create(&mythread, NULL, (void *) thread_function, NULL);
    | ^
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_1_062.c -o Lab_10_Rahil_1_062 -pthread
ubuntu@ubuntu:~/ESRTOS$ ./Lab_10_Rahil_1_062
Hello POSIX thread
Thread ID:140023587858176
ubuntu@ubuntu:~/ESRTOS$
```

ii. Code 2:

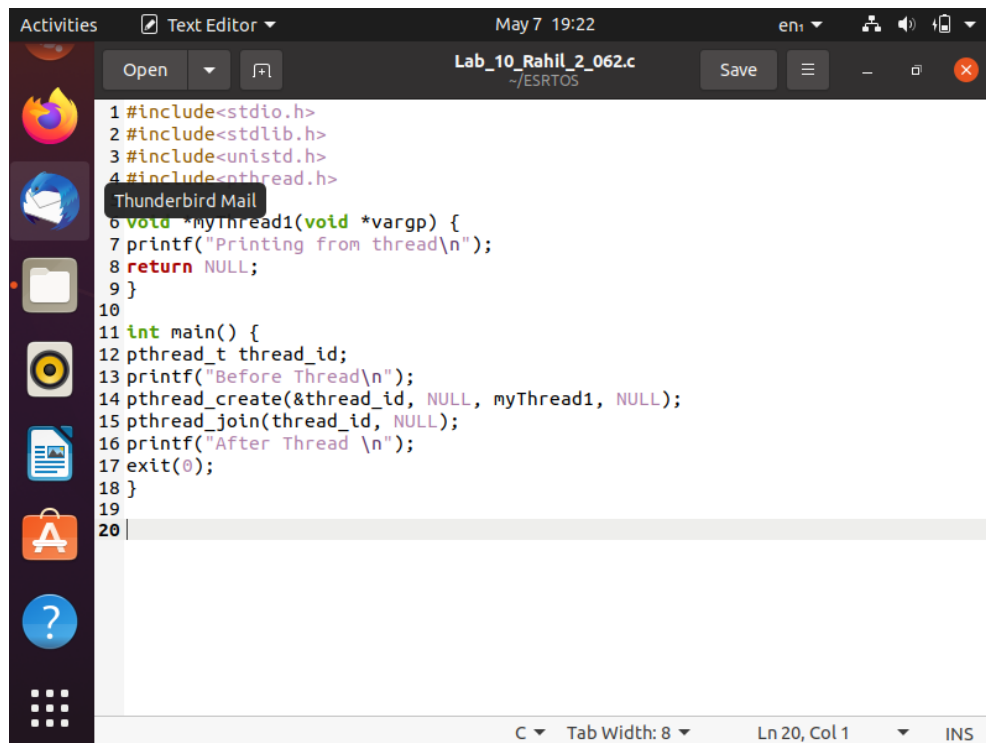
Code of the Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>

void *myThread1(void *vargp){

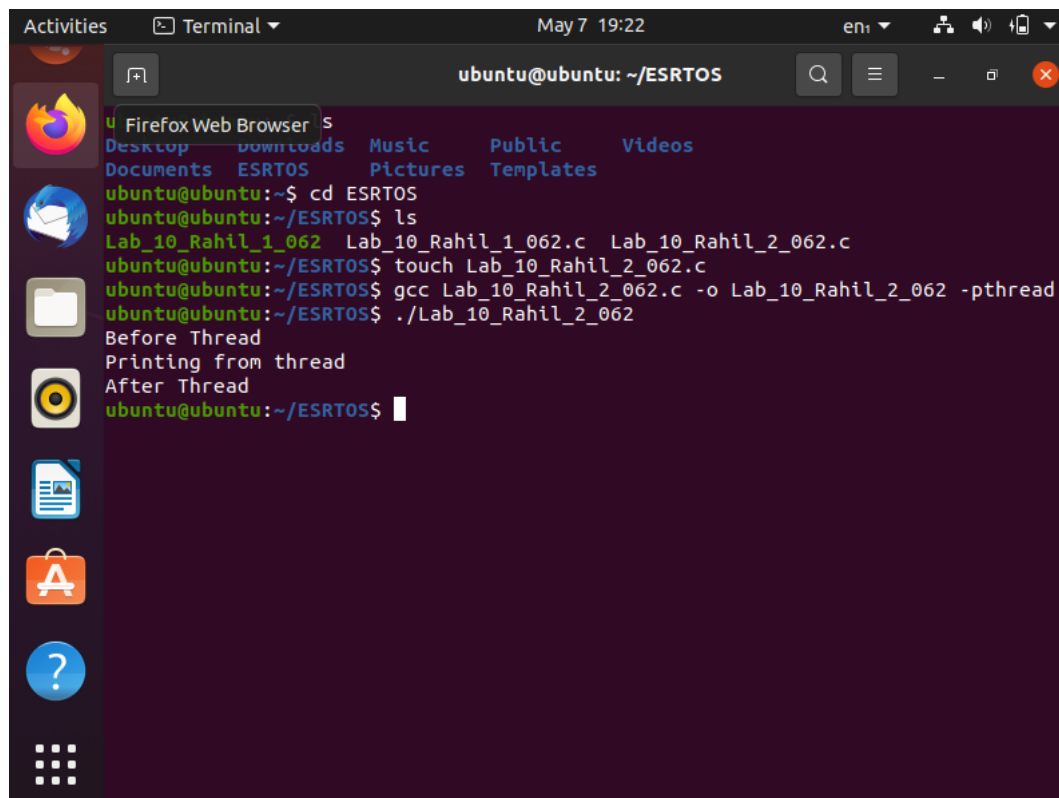
printf("Printing from thread\n");
return NULL;
}
int main(){
pthread_t thread_id;
printf("Before Thread\n");
pthread_create(&thread_id, NULL, myThread1,
NULL); pthread_join(thread_id, NULL);
printf("After Thread \n");
exit(0);
}
```

Screenshot of the Program:



```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<pthread.h>
5
6 void *myThread1(void *vargp) {
7     printf("Printing from thread\n");
8     return NULL;
9 }
10
11 int main() {
12     pthread_t thread_id;
13     printf("Before Thread\n");
14     pthread_create(&thread_id, NULL, myThread1, NULL);
15     pthread_join(thread_id, NULL);
16     printf("After Thread \n");
17     exit(0);
18 }
19
20
```

Output of the Program:



```
ubuntu@ubuntu: ~/ESRTOS
$ cd ESRTOS
$ ls
Lab_10_Rahil_1_062  Lab_10_Rahil_1_062.c  Lab_10_Rahil_2_062.c
$ touch Lab_10_Rahil_2_062.c
$ gcc Lab_10_Rahil_2_062.c -o Lab_10_Rahil_2_062 -pthread
$ ./Lab_10_Rahil_2_062
Before Thread
Printing from thread
After Thread
ubuntu@ubuntu:~/ESRTOS$
```

iii. Code 3

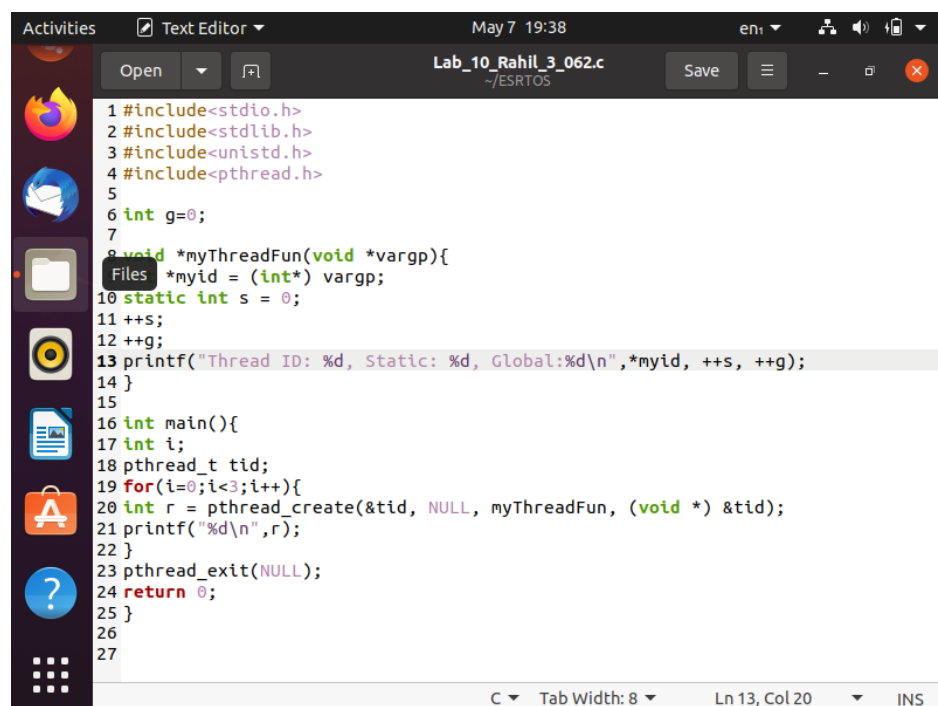
Code of the Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
int g=0;
void *myThreadFun(void *vargp){
int *myid = (int*) vargp;
static int s = 0;
++s;
++g;
printf("Thread ID: %d, Static: %d, Global: %d\n", *myid, ++s, ++g); }

int main(){

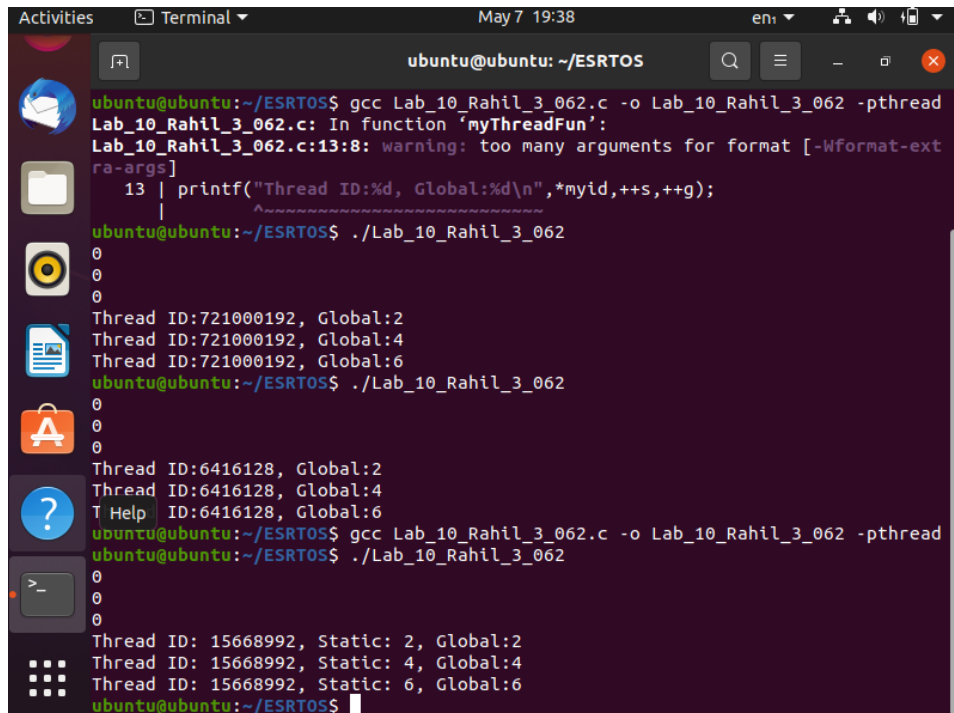
int i;
pthread_t tid;
for (i=0;i<3;i++){
int r = pthread_create(&tid, NULL, myThreadFun, (void *) &tid);
printf("%d\n",r);
}
pthread_exit(NULL);
return 0;
}
```

Screenshot of the Program:



```
Activities Text Editor May 7 19:38 en1
Lab_10_Rahil_3_062.c
~/ESRTOS Save
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<pthread.h>
5
6 int g=0;
7
8 void *myThreadFun(void *vargp){
9     *myid = (int*) vargp;
10    static int s = 0;
11    ++s;
12    ++g;
13    printf("Thread ID: %d, Static: %d, Global:%d\n",*myid, ++s, ++g);
14 }
15
16 int main(){
17     int i;
18     pthread_t tid;
19     for(i=0;i<3;i++){
20         int r = pthread_create(&tid, NULL, myThreadFun, (void *) &tid);
21         printf("%d\n",r);
22     }
23     pthread_exit(NULL);
24     return 0;
25 }
26
27
C Tab Width: 8 Ln 13, Col 20 INS
```

Output of the Program:



```
ubuntu@ubuntu: ~/ESRTOS
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_3_062.c -o Lab_10_Rahil_3_062 -pthread
Lab_10_Rahil_3_062.c: In function 'myThreadFun':
Lab_10_Rahil_3_062.c:13:8: warning: too many arguments for format [-Wformat-extra-args]
   13 | printf("Thread ID:%d, Global:%d\n",*myid,++s,++g);
      |
ubuntu@ubuntu:~/ESRTOS$ ./Lab_10_Rahil_3_062
0
0
0
Thread ID:721000192, Global:2
Thread ID:721000192, Global:4
Thread ID:721000192, Global:6
ubuntu@ubuntu:~/ESRTOS$ ./Lab_10_Rahil_3_062
0
0
0
Thread ID:6416128, Global:2
Thread ID:6416128, Global:4
T Help ID:6416128, Global:6
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_3_062.c -o Lab_10_Rahil_3_062 -pthread
ubuntu@ubuntu:~/ESRTOS$ ./Lab_10_Rahil_3_062
0
0
0
Thread ID: 15668992, Static: 2, Global:2
Thread ID: 15668992, Static: 4, Global:4
Thread ID: 15668992, Static: 6, Global:6
ubuntu@ubuntu:~/ESRTOS$
```

iv. CODE 4:

Code of the Program:

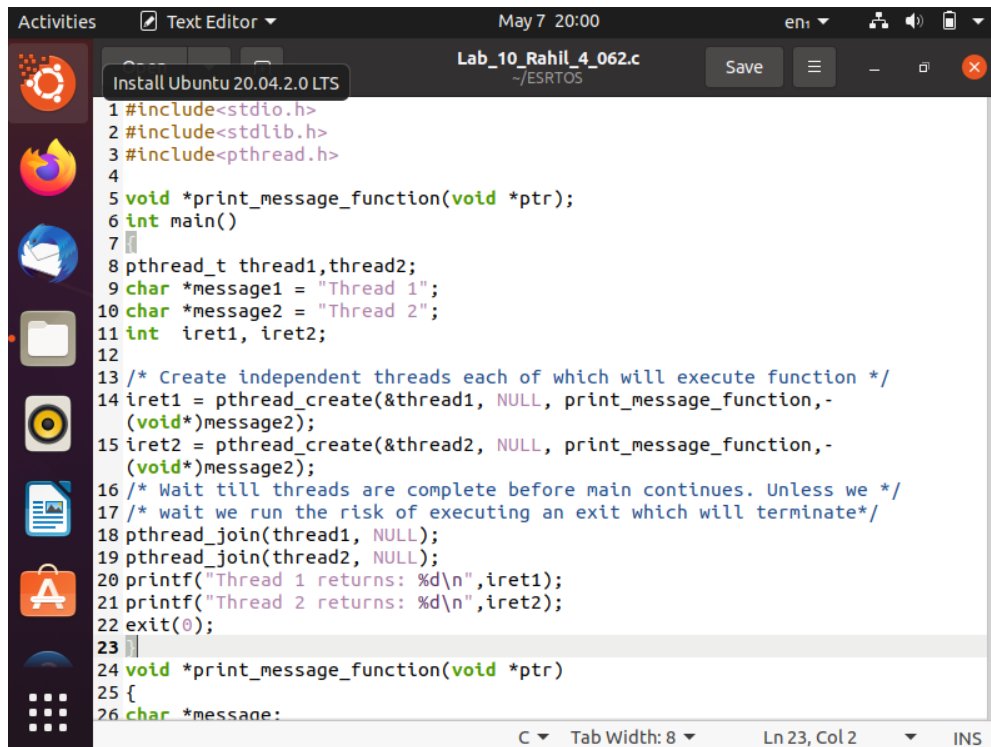
```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
void *print_message_function( void *ptr );
int main()
{
    pthread_t thread1, thread2;
    char *message1 = "Thread 1";
    char *message2 = "Thread 2";
    int iret1, iret2;
    /* Create independent threads each of which will execute function */
    pthread_create( &thread1, NULL, print_message_function, (void*) message1);
    iret2 = pthread_create( &thread2, NULL, print_message_function,
    (void*) message2);
    /* Wait till threads are complete before main continues. Unless we wait we run the
    risk of executing an exit which will terminate the process and all threads before the
    threads have completed. The pthread_join() function provides a simple mechanism
    allowing an application to wait for a thread to terminate. After the thread
    terminates, the application may then choose to clean up resources that were used
    by the thread.
    pthread_join( thread1, NULL);
    pthread_join( thread2, NULL);
    printf("Thread 1 returns: %d\n",iret1);
    printf("Thread 2 returns: %d\n",iret2);
    exit(0);
}
```

```

void *print_message_function( void *ptr )
{
    char *message;
    message = (char *) ptr;
    printf("%s \n", message);
}

```

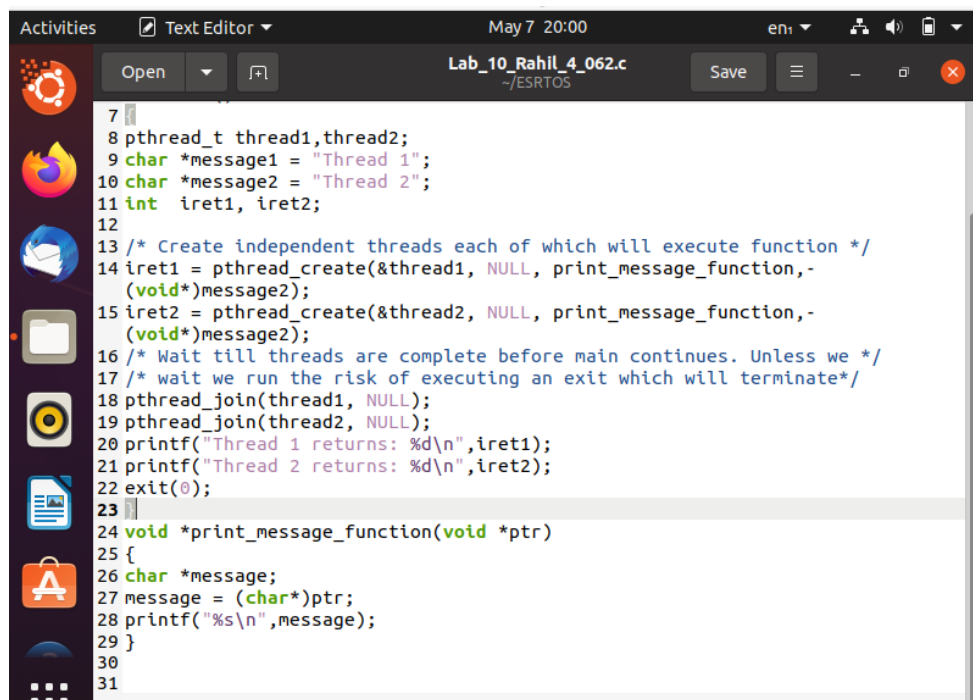
Screenshot of the Program:



```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<pthread.h>
4
5 void *print_message_function(void *ptr);
6 int main()
7 {
8     pthread_t thread1,thread2;
9     char *message1 = "Thread 1";
10    char *message2 = "Thread 2";
11    int iret1, iret2;
12
13    /* Create independent threads each of which will execute function */
14    iret1 = pthread_create(&thread1, NULL, print_message_function,-
15        (void*)message2);
16    iret2 = pthread_create(&thread2, NULL, print_message_function,-
17        (void*)message2);
18    /* Wait till threads are complete before main continues. Unless we */
19    /* wait we run the risk of executing an exit which will terminate*/
20    pthread_join(thread1, NULL);
21    pthread_join(thread2, NULL);
22    printf("Thread 1 returns: %d\n",iret1);
23    printf("Thread 2 returns: %d\n",iret2);
24    exit(0);
25 }
26 void *print_message_function(void *ptr)
27 {
28     char *message;

```

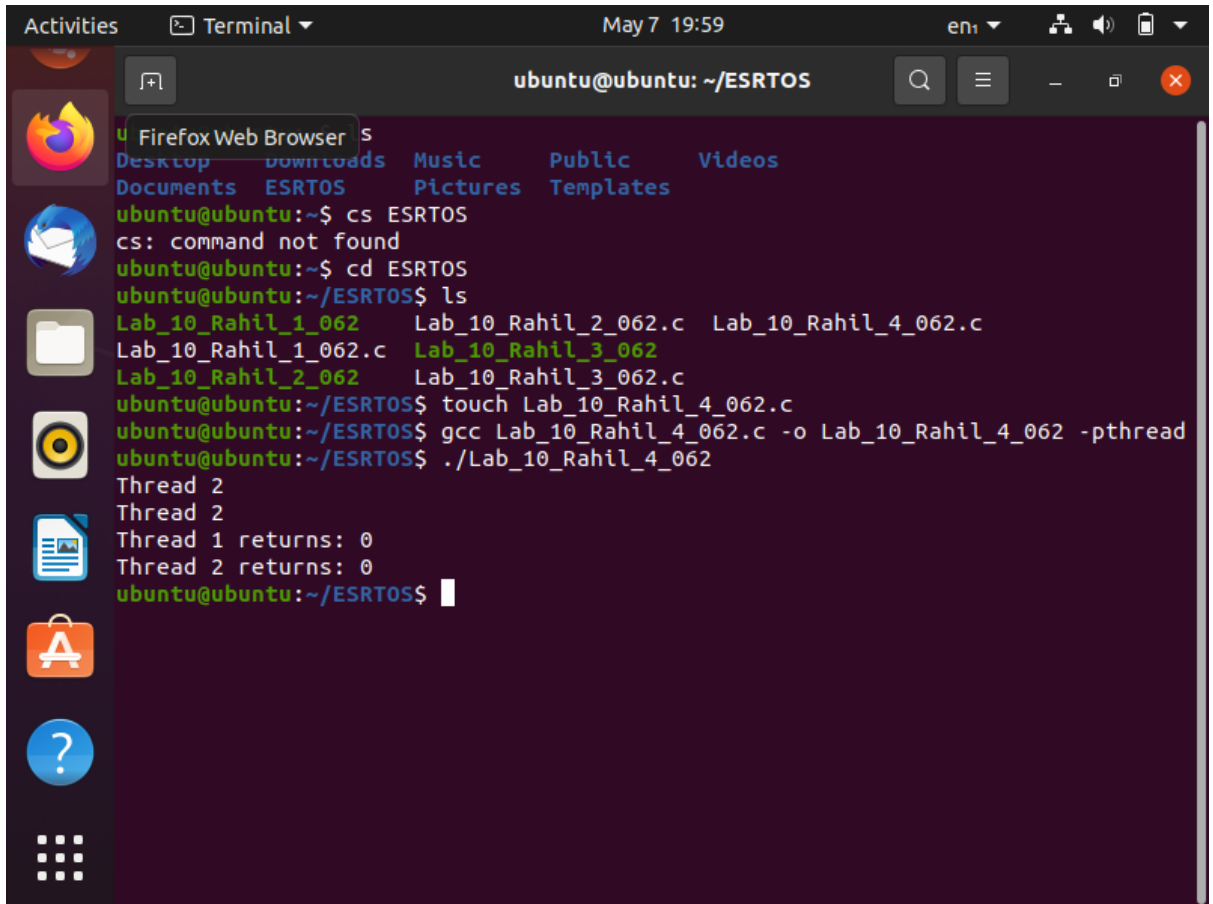


```

7
8 pthread_t thread1,thread2;
9 char *message1 = "Thread 1";
10 char *message2 = "Thread 2";
11 int iret1, iret2;
12
13 /* Create independent threads each of which will execute function */
14 iret1 = pthread_create(&thread1, NULL, print_message_function,-
15     (void*)message2);
16 iret2 = pthread_create(&thread2, NULL, print_message_function,-
17     (void*)message2);
18 /* Wait till threads are complete before main continues. Unless we */
19 /* wait we run the risk of executing an exit which will terminate*/
20 pthread_join(thread1, NULL);
21 pthread_join(thread2, NULL);
22 printf("Thread 1 returns: %d\n",iret1);
23 printf("Thread 2 returns: %d\n",iret2);
24 exit(0);
25 }
26 void *print_message_function(void *ptr)
27 {
28     char *message;
29     message = (char*)ptr;
30     printf("%s\n",message);
31 }

```

Output of the Program:



The screenshot shows a terminal window titled 'Terminal' with the date 'May 7 19:59' and the user 'ubuntu@ubuntu: ~/ESRTOS'. The terminal displays the following commands and output:

```
ubuntu@ubuntu:~$ cs ESRTOS
cs: command not found
ubuntu@ubuntu:~$ cd ESRTOS
ubuntu@ubuntu:~/ESRTOS$ ls
Lab_10_Rahil_1_062.c  Lab_10_Rahil_2_062.c  Lab_10_Rahil_4_062.c
Lab_10_Rahil_1_062.c  Lab_10_Rahil_3_062.c
Lab_10_Rahil_2_062.c  Lab_10_Rahil_3_062.c
ubuntu@ubuntu:~/ESRTOS$ touch Lab_10_Rahil_4_062.c
ubuntu@ubuntu:~/ESRTOS$ gcc Lab_10_Rahil_4_062.c -o Lab_10_Rahil_4_062 -pthread
ubuntu@ubuntu:~/ESRTOS$ ./Lab_10_Rahil_4_062
Thread 2
Thread 2
Thread 1 returns: 0
Thread 2 returns: 0
ubuntu@ubuntu:~/ESRTOS$
```

v. CODE 5:

Code of the Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
#define NUM_THREADS 5
void *PrintHello(int threadid)
{
    long int tid;
    tid= (int)threadid;
    printf("Hello World! It's me, thread #%ld!\n", tid);
    pthread_exit(NULL);
}
int main (int argc, char *argv[])
{
    pthread_t threads[NUM_THREADS];
    long int rc, t;
    for(t=0; t<NUM_THREADS; t++)
    {
        printf("In main: creating thread %ld\n", t);
```

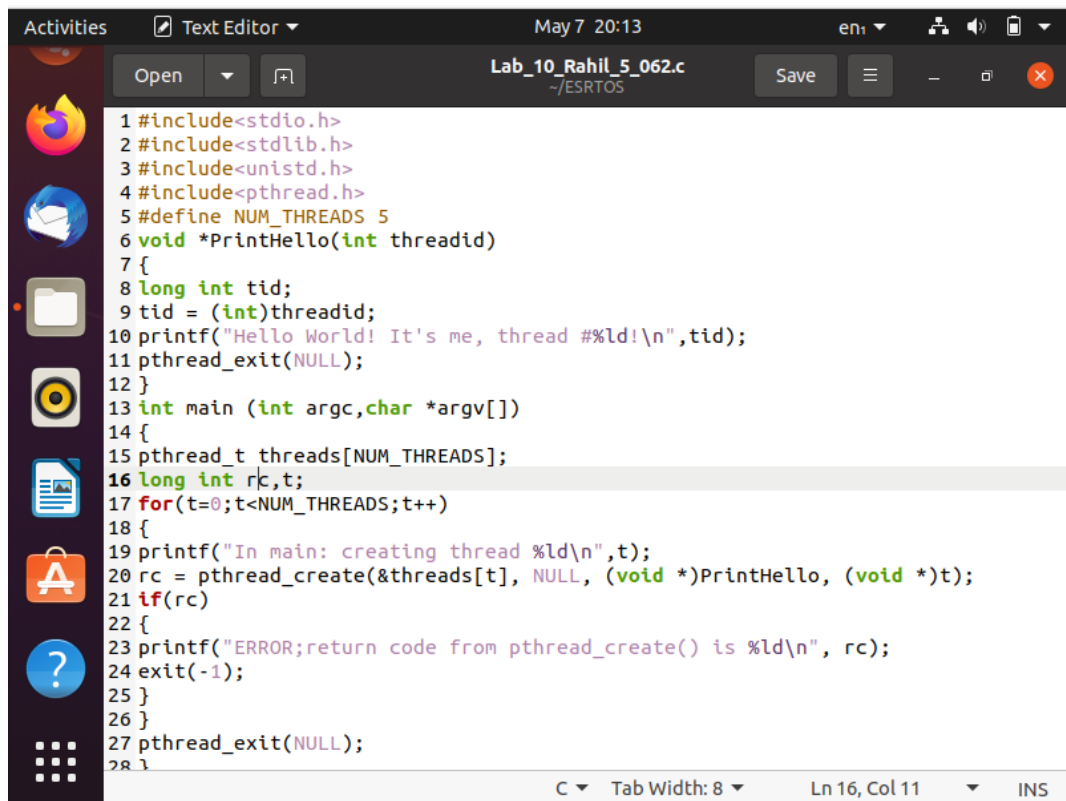


```

rc= pthread_create(&threads[t],NULL, (void *)PrintHello, (void *)t);
if (rc)
{
printf("ERROR; return code from pthread_create() is %ld\n", rc);
exit(-1);
}
}
pthread_exit(NULL);
}

```

Screenshot of the Program:



The screenshot shows a Linux desktop environment with a text editor window titled "Lab_10_Rahil_5_062.c". The editor displays a C program that creates 5 threads. The code is as follows:

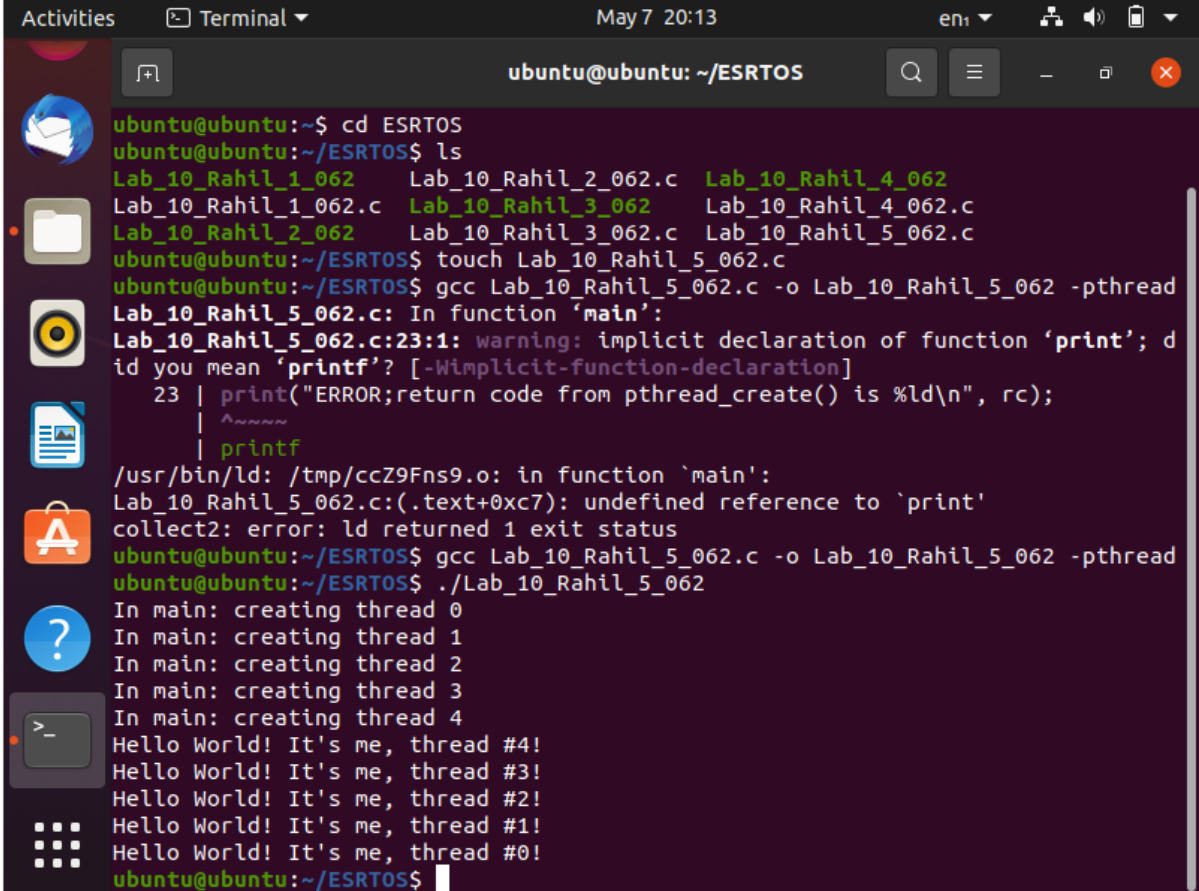
```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<pthread.h>
5 #define NUM_THREADS 5
6 void *PrintHello(int threadid)
7 {
8     long int tid;
9     tid = (int)threadid;
10    printf("Hello World! It's me, thread #%ld!\n",tid);
11    pthread_exit(NULL);
12 }
13 int main (int argc,char *argv[])
14 {
15    pthread_t threads[NUM_THREADS];
16    long int rc,t;
17    for(t=0;t<NUM_THREADS;t++)
18    {
19        printf("In main: creating thread %ld\n",t);
20        rc = pthread_create(&threads[t], NULL, (void *)PrintHello, (void *)t);
21        if(rc)
22        {
23            printf("ERROR;return code from pthread_create() is %ld\n", rc);
24            exit(-1);
25        }
26    }
27    pthread_exit(NULL);
28 }

```

The status bar at the bottom indicates the current cursor position is at Line 16, Column 11, and the text is in Insert (INS) mode.

Output of the Program:



```
ubuntu@ubuntu: ~$ cd ESRTOS
ubuntu@ubuntu: ~/ESRTOS$ ls
Lab_10_Rahil_1_062      Lab_10_Rahil_2_062.c  Lab_10_Rahil_4_062
Lab_10_Rahil_1_062.c    Lab_10_Rahil_3_062    Lab_10_Rahil_4_062.c
Lab_10_Rahil_2_062      Lab_10_Rahil_3_062.c  Lab_10_Rahil_5_062.c
ubuntu@ubuntu: ~/ESRTOS$ touch Lab_10_Rahil_5_062.c
ubuntu@ubuntu: ~/ESRTOS$ gcc Lab_10_Rahil_5_062.c -o Lab_10_Rahil_5_062 -pthread
Lab_10_Rahil_5_062.c: In function 'main':
Lab_10_Rahil_5_062.c:23:1: warning: implicit declaration of function 'print'; did you mean 'printf'? [-Wimplicit-function-declaration]
   23 | print("ERROR;return code from pthread_create() is %ld\n", rc);
      | ~~~~~
      | printf
/usr/bin/ld: /tmp/ccZ9Fns9.o: in function 'main':
Lab_10_Rahil_5_062.c:(.text+0xc7): undefined reference to 'print'
collect2: error: ld returned 1 exit status
ubuntu@ubuntu: ~/ESRTOS$ gcc Lab_10_Rahil_5_062.c -o Lab_10_Rahil_5_062 -pthread
ubuntu@ubuntu: ~/ESRTOS$ ./Lab_10_Rahil_5_062
In main: creating thread 0
In main: creating thread 1
In main: creating thread 2
In main: creating thread 3
In main: creating thread 4
Hello World! It's me, thread #4!
Hello World! It's me, thread #3!
Hello World! It's me, thread #2!
Hello World! It's me, thread #1!
Hello World! It's me, thread #0!
ubuntu@ubuntu: ~/ESRTOS$
```

CONCLUSION: In this experiment we have learnt about POSIX thread in C and Linux.

