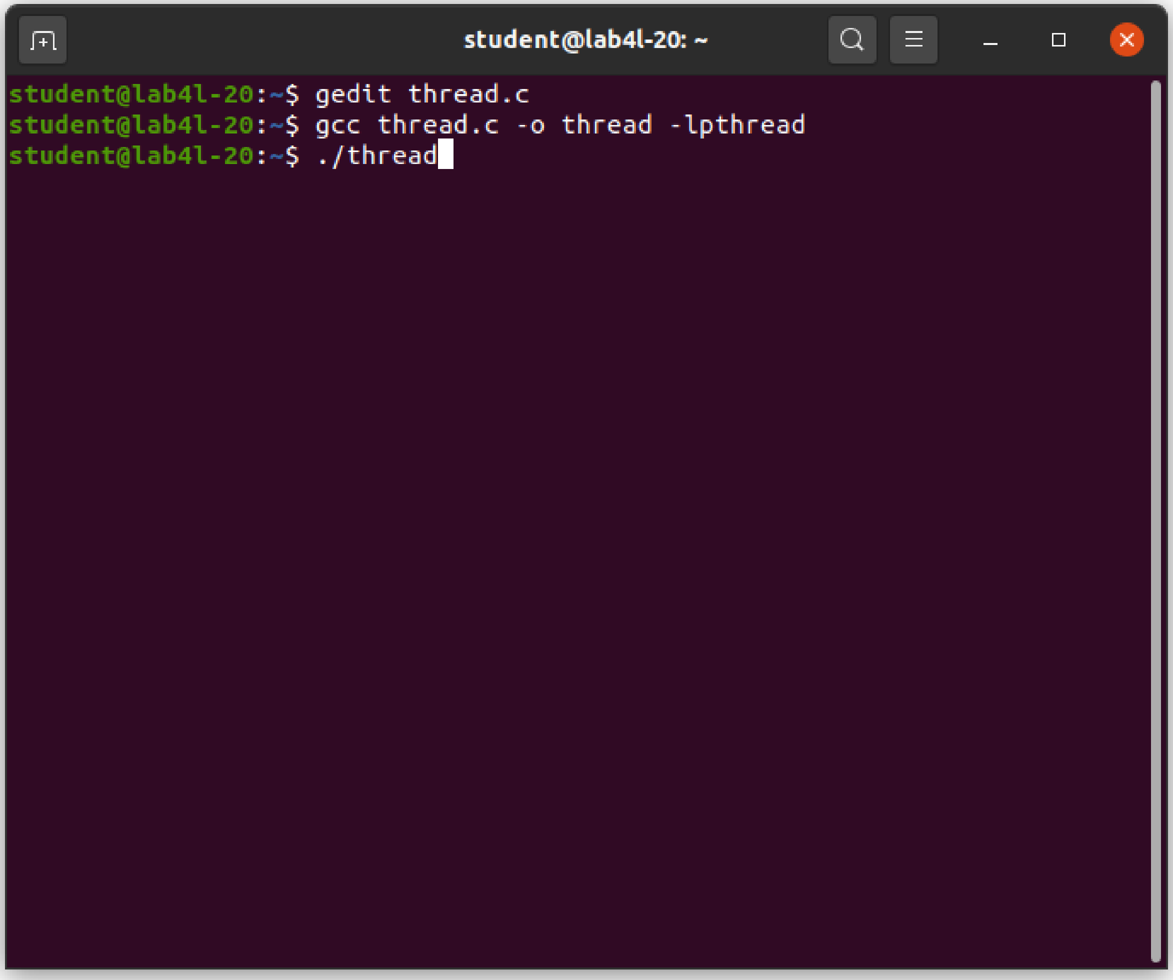
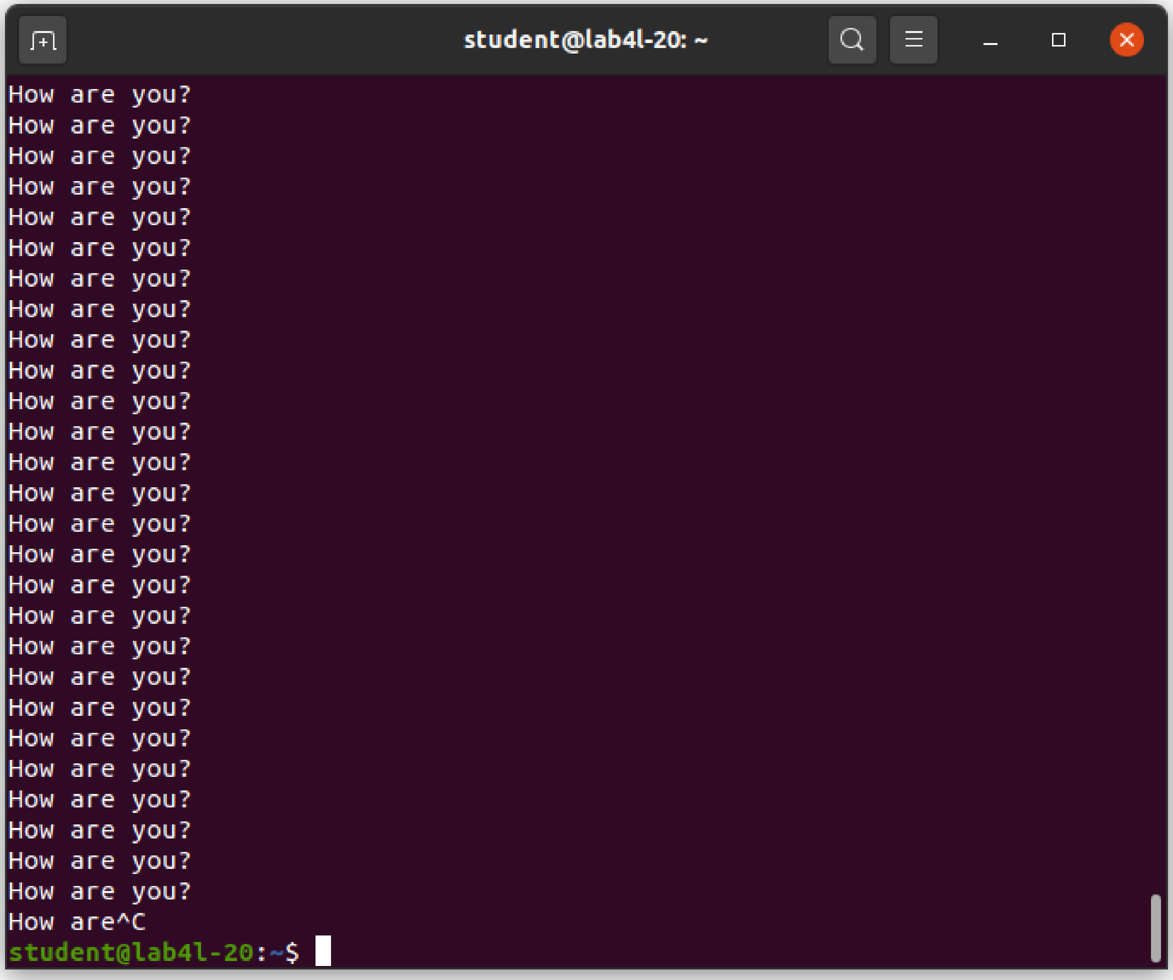
**Ayaz Hasan**
**20k-1044**
**SE-5A**

# LAB 8

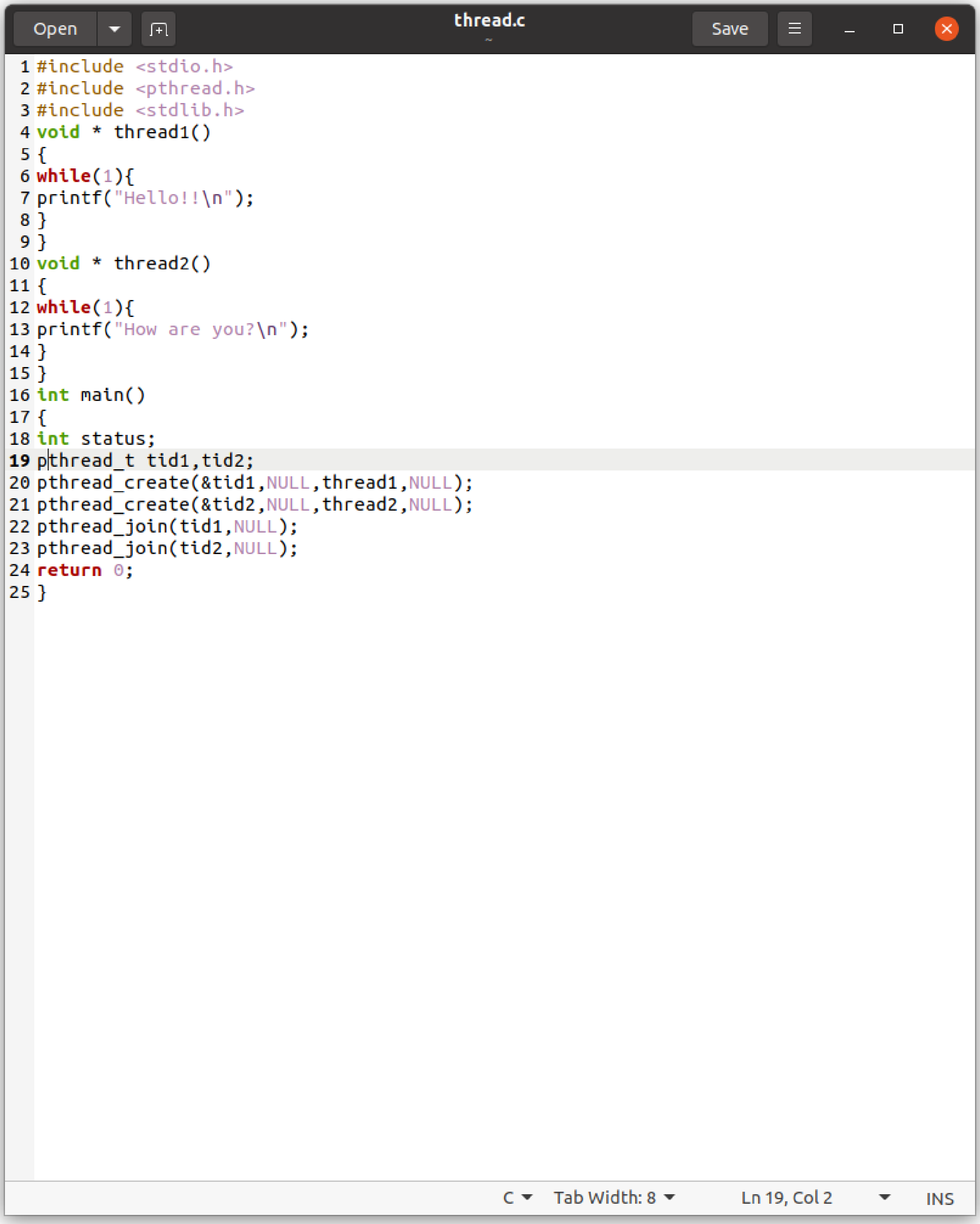
### Example 1



```
student@lab4l-20:~$ gedit thread.c
student@lab4l-20:~$ gcc thread.c -o thread -lpthread
student@lab4l-20:~$ ./thread
```

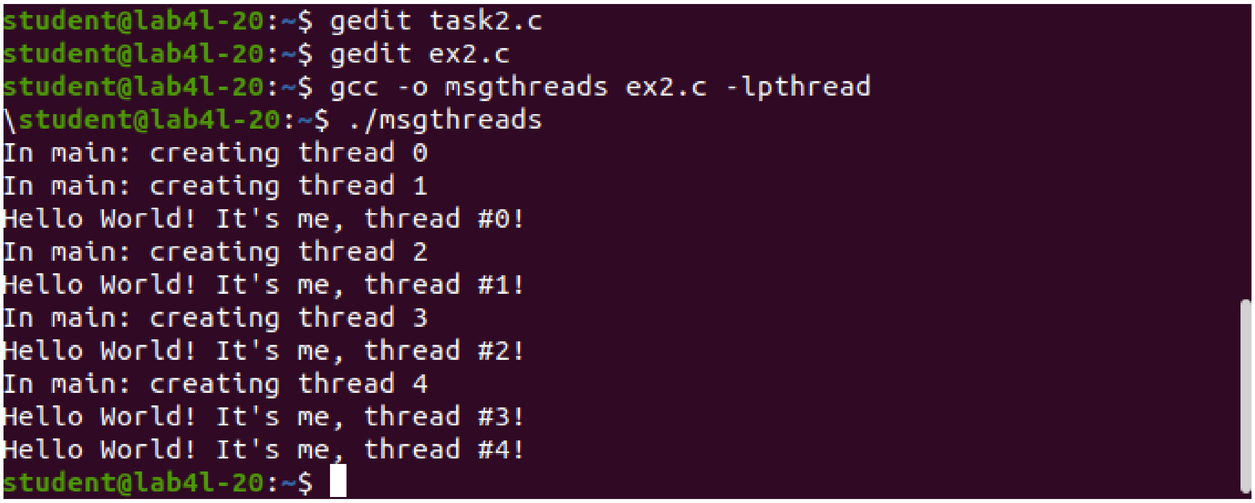


```
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are you?
How are^C
student@lab4l-20:~$ 
```

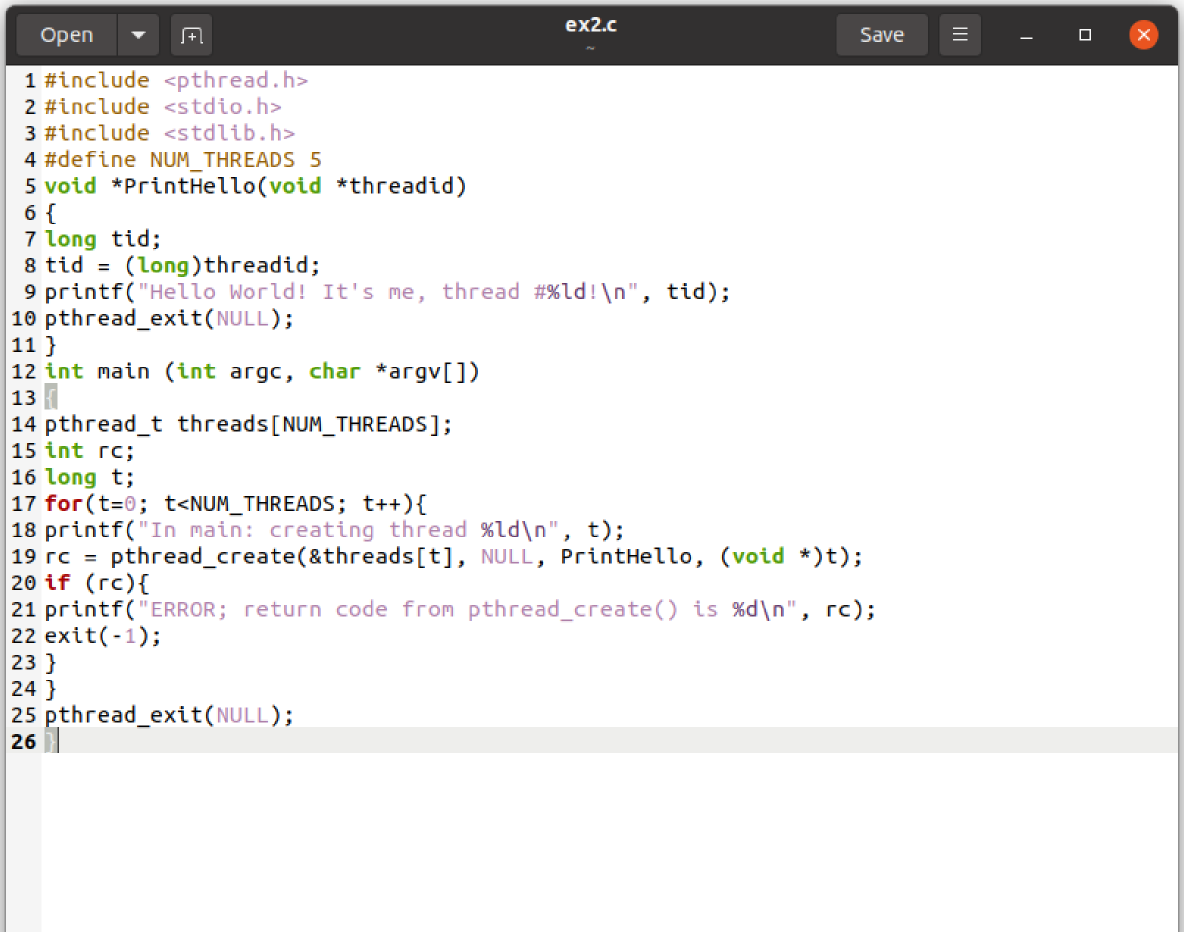


```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void * thread1()
{
while(1){
printf("Hello!!\n");
}
}
void * thread2()
{
while(1){
printf("How are you?\n");
}
}
int main()
{
int status;
pthread_t tid1,tid2;
pthread_create(&tid1,NULL,thread1,NULL);
pthread_create(&tid2,NULL,thread2,NULL);
pthread_join(tid1,NULL);
pthread_join(tid2,NULL);
return 0;
}
```

# Example 2

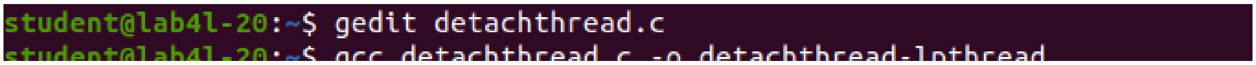


```
student@lab4l-20:~$ gedit task2.c
student@lab4l-20:~$ gedit ex2.c
student@lab4l-20:~$ gcc -o msgthreads ex2.c -lpthread
\student@lab4l-20:~$ ./msgthreads
In main: creating thread 0
In main: creating thread 1
Hello World! It's me, thread #0!
In main: creating thread 2
Hello World! It's me, thread #1!
In main: creating thread 3
Hello World! It's me, thread #2!
In main: creating thread 4
Hello World! It's me, thread #3!
Hello World! It's me, thread #4!
student@lab4l-20:~$
```

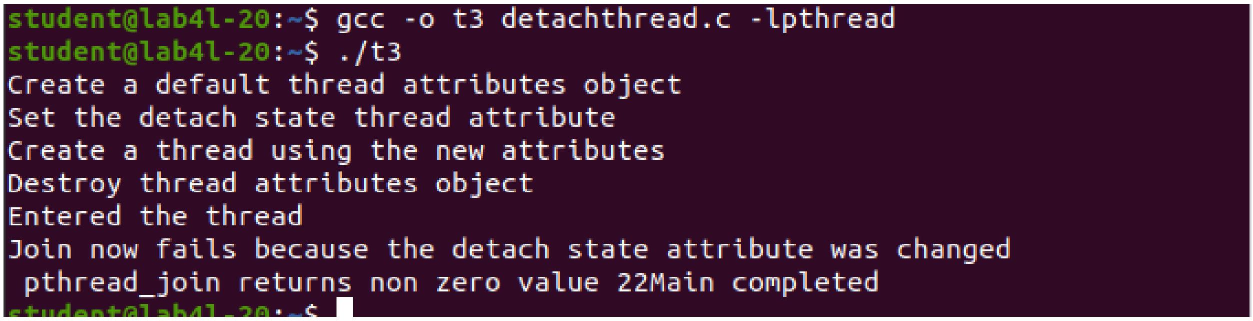


```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUM_THREADS 5
void *PrintHello(void *threadid)
{
long tid;
tid = (long)threadid;
printf("Hello World! It's me, thread #%ld!\n", tid);
pthread_exit(NULL);
}
int main (int argc, char *argv[])
{
pthread_t threads[NUM_THREADS];
int rc;
long t;
for(t=0; t<NUM_THREADS; t++){
printf("In main: creating thread %ld\n", t);
rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
if (rc){
printf("ERROR; return code from pthread_create() is %d\n", rc);
exit(-1);
}
}
pthread_exit(NULL);
}
```

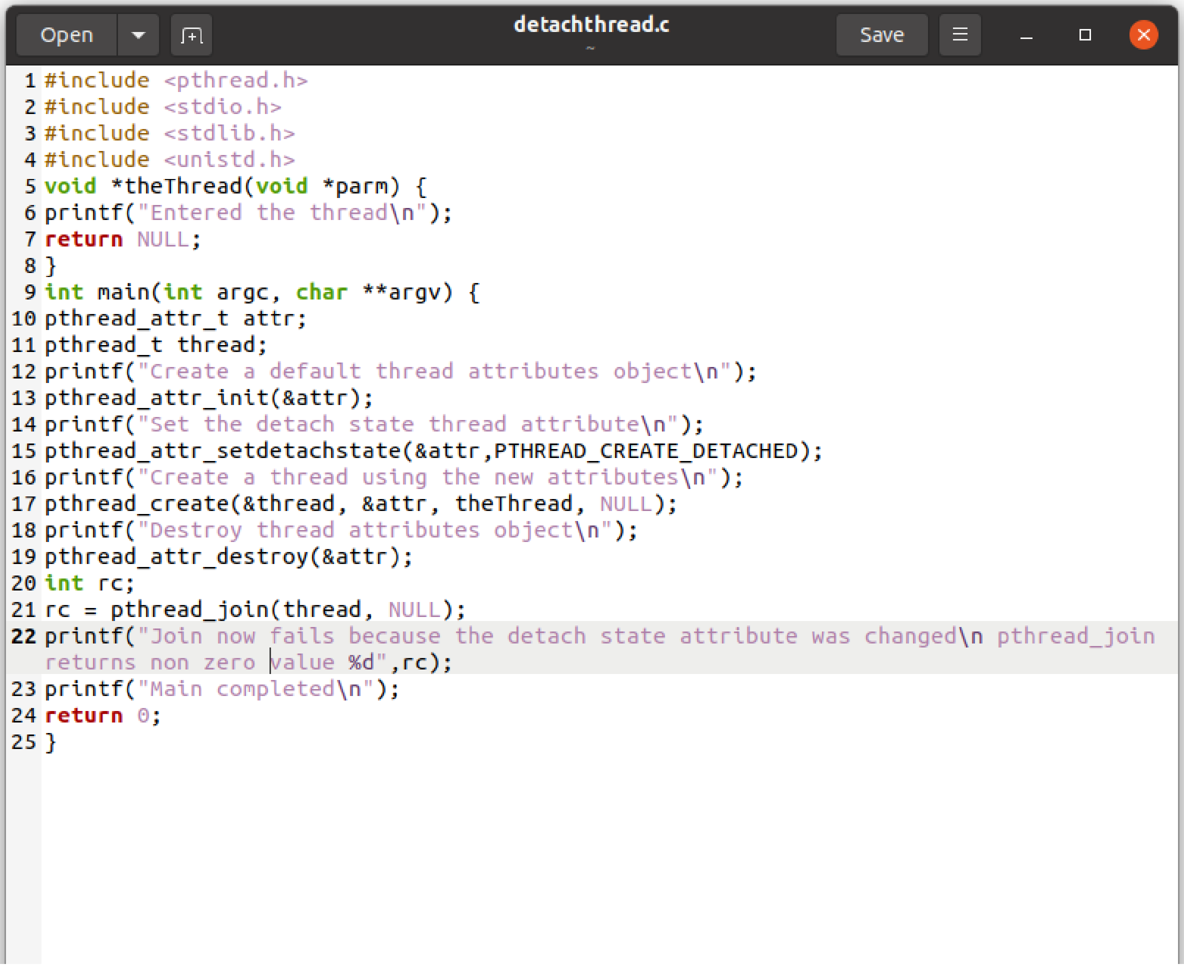
## Example 3



```
student@lab4l-20:~$ gedit detachthread.c
student@lab4l-20:~$ gcc detachthread.c -o detachthread-lpthread
```



```
student@lab4l-20:~$ gcc -o t3 detachthread.c -lpthread
student@lab4l-20:~$ ./t3
Create a default thread attributes object
Set the detach state thread attribute
Create a thread using the new attributes
Destroy thread attributes object
Entered the thread
Join now fails because the detach state attribute was changed
 pthread_join returns non zero value 22Main completed
student@lab4l-20:~$
```



```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
void *theThread(void *parm) {
printf("Entered the thread\n");
return NULL;
}
int main(int argc, char **argv) {
pthread_attr_t attr;
pthread_t thread;
printf("Create a default thread attributes object\n");
pthread_attr_init(&attr);
printf("Set the detach state thread attribute\n");
pthread_attr_setdetachstate(&attr,PTHREAD_CREATE_DETACHED);
printf("Create a thread using the new attributes\n");
pthread_create(&thread, &attr, theThread, NULL);
printf("Destroy thread attributes object\n");
pthread_attr_destroy(&attr);
int rc;
rc = pthread_join(thread, NULL);
printf("Join now fails because the detach state attribute was changed\n pthread_join returns non zero value %d",rc);
printf("Main completed\n");
return 0;
}
```