

Pthreads

Operating System



September 18, 2018

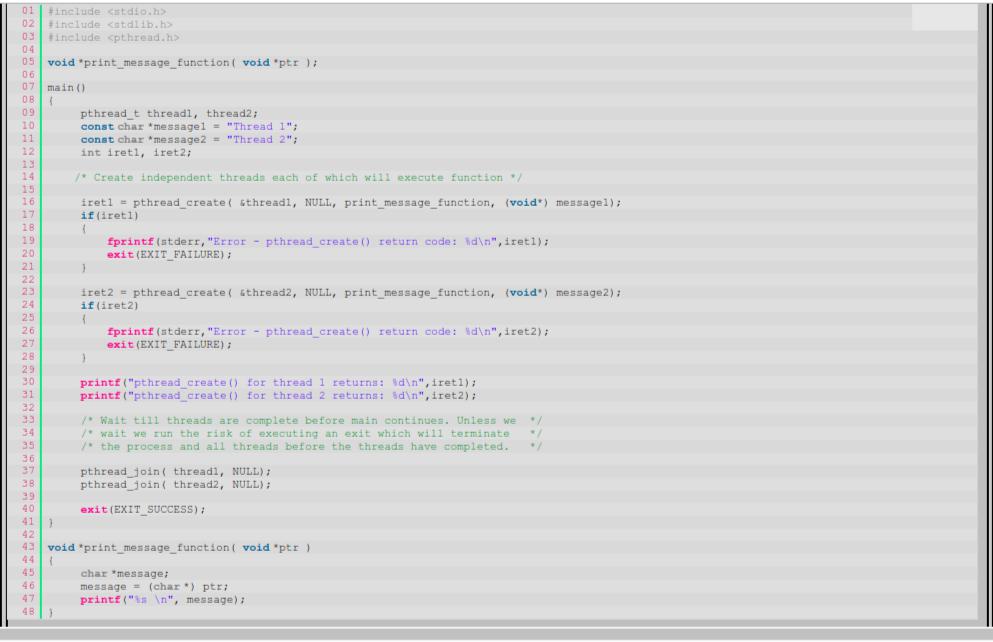
syed abbas hussain naqvi

177410

## **Thread Creation:**

1. For the compilation of test.c file the command which I used is “cc -pthread test.c”. Then to execute the file I used this command “/a.out”.

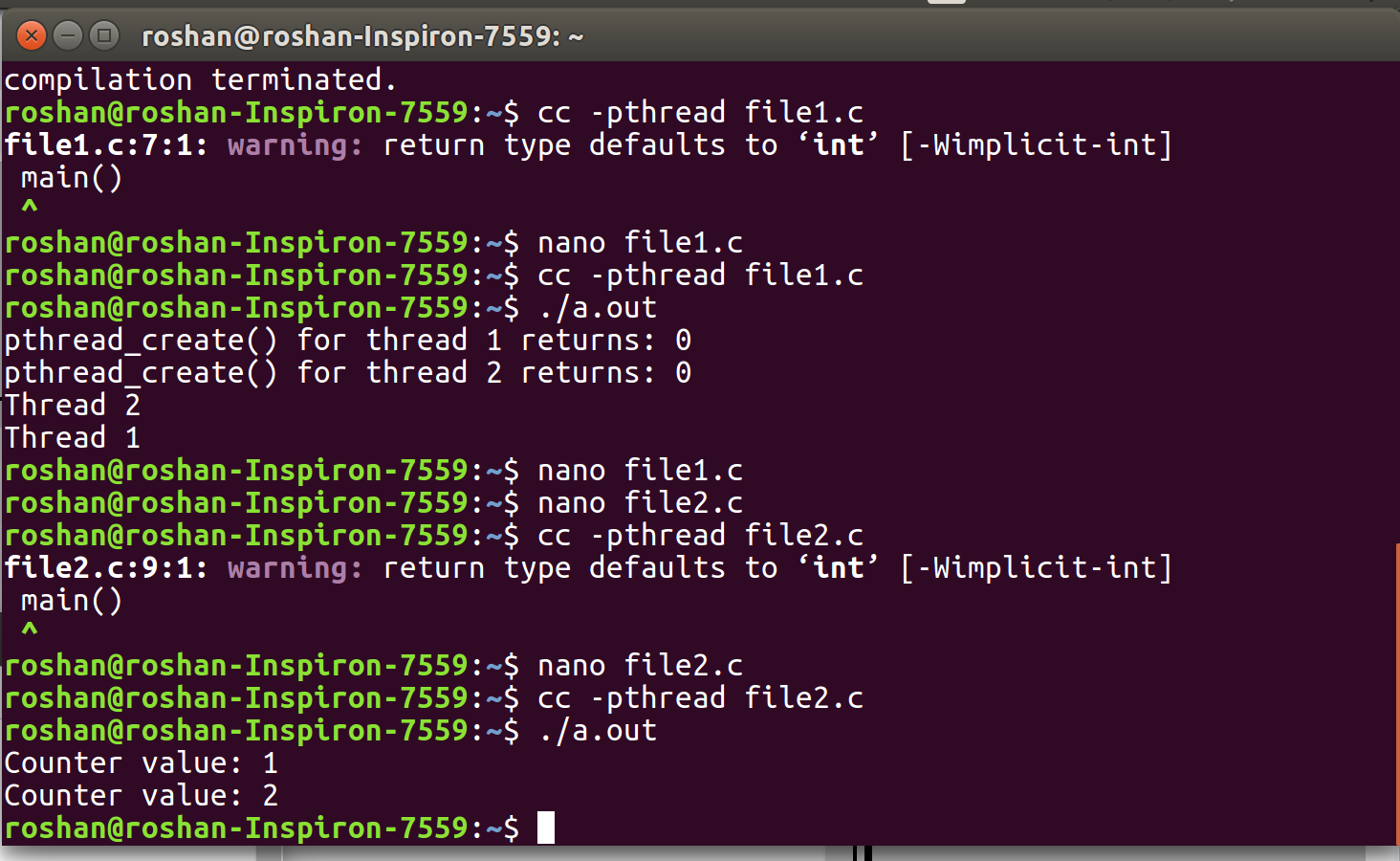
Figure 1 : Thread Creation

1. Basically in this code we are creating two threads by using pthread API for c and c++. Pthread\_create() function creates an independent thread which will print the message function on the terminal. The return type of this function is integer and it will return the thread id.
2. Here we are reusing the function with different arguments. But the function of two thread can be different.
3. The below code will terminated when we call pthread.exit().

## **Mutually Exclusive Locks (mutexes):**

1. For the compilation of test1.c file the command which I used is “cc -pthread test1.c”. Then to execute the file I used this command “/a.out”.
2. Basically in this code we are creating two threads by using pthread\_create() function.
3. As many threads are performing operations on same memory at the same time so to prevent data inconsistency we use mutexes.
4. By using mutex if it try to access a mutex which is used by another thread then the thread is blocked until the mutex is not unlocked.
5. As in this code we are creating two threads so 1st thread will access the counter variable and increment it, whereas second thread will remain block because mutex is locked. But when the mutex is unlocked 2nd thread will then access the value of variable.

Figure : Mutex



**Thread Joins:**

1. For the compilation of test2.c file the command which I used is “cc -pthread test2.c”. Then to execute the file I used this command “/a.out”.
2. Inside the main function we are creating a new thread by calling pthread\_create() function and assigning them the task which they have to perform like inside thread\_function() we are printing the message which is “Thread number something” according to thread id. Number of threads are equal to number of id’s we used to initialize the array of thread id.
3. After thread creation we are calling pthread\_join() function to make sure that program does not start executing next line before completing all the threads.
4. Once all the threads are executed counter value will be printed on the display.

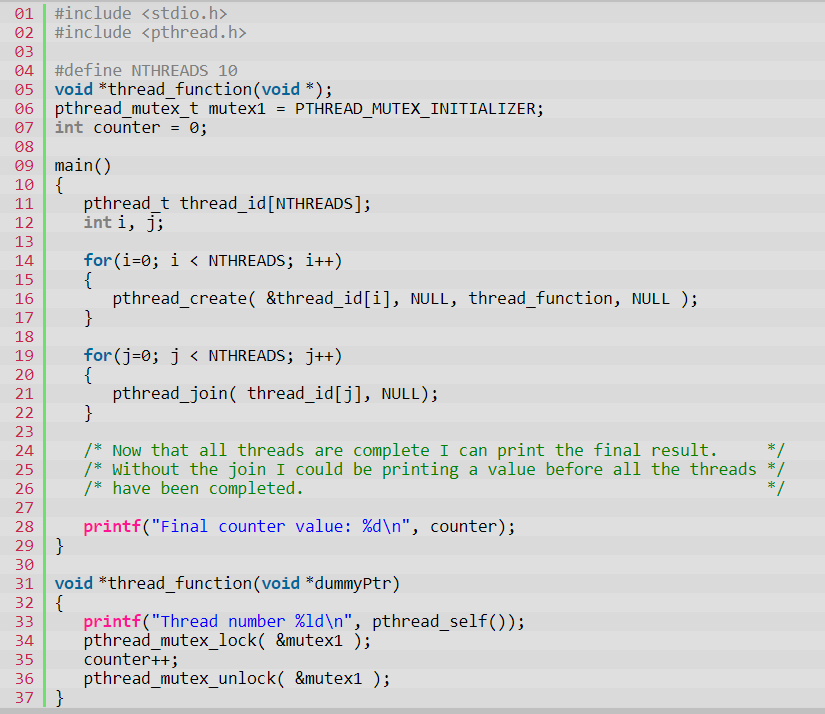


Figure : Join

