

Question :

Multithreading in C++

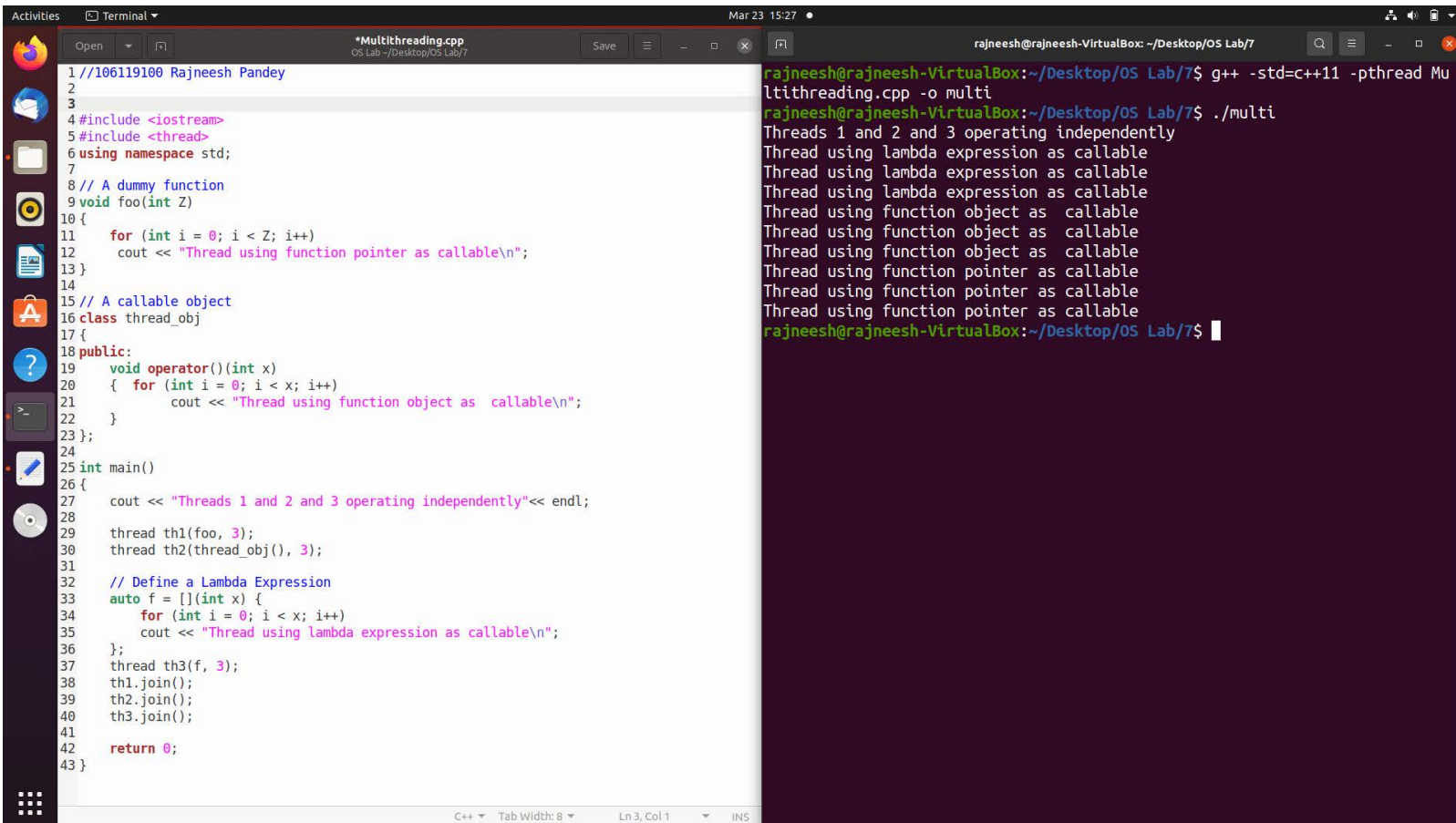
Multithreading support was introduced in C++11. Prior to C++11, we had to use [POSIX threads or pthreads library in C](#). While this library did the job the lack of any standard language provided feature-set caused serious portability issues. C++ 11 did away with all that and gave us **std::thread**. The thread classes and related functions are defined in the **thread** header file.

std::thread is the thread class that represents a single thread in C++. To start a thread we simply need to create a new thread object and pass the executing code to be called (i.e, a callable object) into the constructor of the object. Once the object is created a new thread is launched which will execute the code specified in callable.

A callable can be either of the three

- A function pointer
- A function object
- A lambda expression

Program and Input/Output



```
1 //106119100 Rajneesh Pandey
2
3
4 #include <iostream>
5 #include <thread>
6 using namespace std;
7
8 // A dummy function
9 void foo(int Z)
10 {
11     for (int i = 0; i < Z; i++)
12         cout << "Thread using function pointer as callable\n";
13 }
14
15 // A callable object
16 class thread_obj
17 {
18 public:
19     void operator()(int x)
20     { for (int i = 0; i < x; i++)
21         cout << "Thread using function object as callable\n";
22     }
23 };
24
25 int main()
26 {
27     cout << "Threads 1 and 2 and 3 operating independently" << endl;
28     thread th1(foo, 3);
29     thread th2(thread_obj(), 3);
30
31     // Define a Lambda Expression
32     auto f = [](int x) {
33         for (int i = 0; i < x; i++)
34             cout << "Thread using lambda expression as callable\n";
35     };
36     thread th3(f, 3);
37     th1.join();
38     th2.join();
39     th3.join();
40
41     return 0;
42 }
```

```
rajneesh@rajneesh-VirtualBox: ~/Desktop/OS Lab/7$ g++ -std=c++11 -pthread Multithreading.cpp -o multi
rajneesh@rajneesh-VirtualBox: ~/Desktop/OS Lab/7$ ./multi
Threads 1 and 2 and 3 operating independently
Thread using lambda expression as callable
Thread using lambda expression as callable
Thread using lambda expression as callable
Thread using function object as callable
Thread using function object as callable
Thread using function object as callable
Thread using function pointer as callable
Thread using function pointer as callable
Thread using function pointer as callable
rajneesh@rajneesh-VirtualBox: ~/Desktop/OS Lab/7$
```