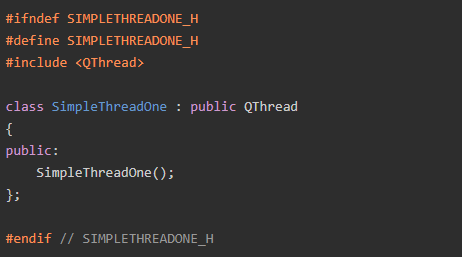
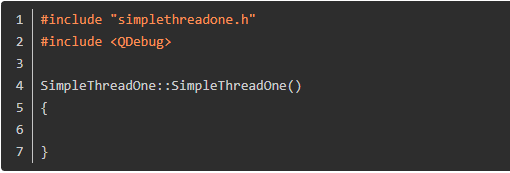
Qt子线程

QT多线程的实现有两种方法，一种是继承QThread的多线程使用方法，另外一种是使用QObject实现多线的方法。传统的方式是继承QTread，但是这种方式比较的容易出错，QT官方推荐使用的是第二种方式。这里介绍这两种方式的最简单的一种创建方式。

1. 方法一: 继承QThread
2. 创建一个QT应用
3. 创建线程类，继承QThread

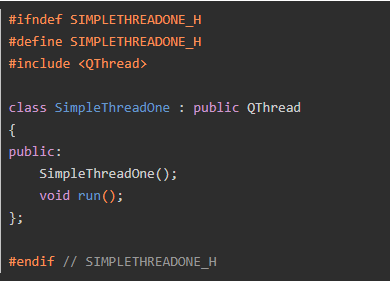


图一：simplethreadone.h

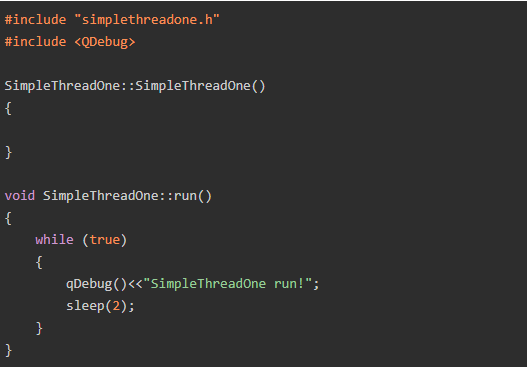


图二：simplethreadone.cpp

1. 重写QThread的run方法

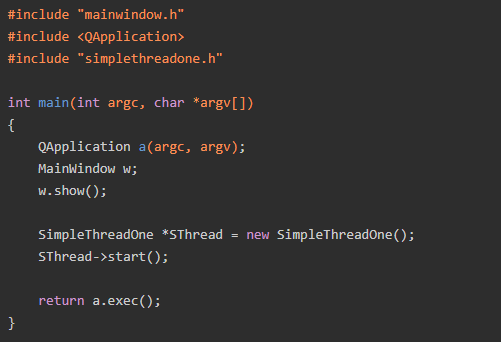


图三：simplethreadone.h

. 

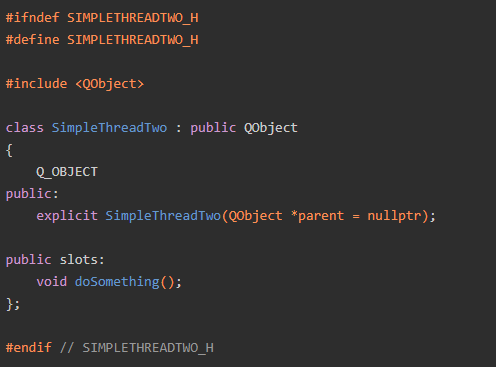
图四：simplethreadone.cpp

1. 主线程中创建线程对象，使用start()方法启动线程。

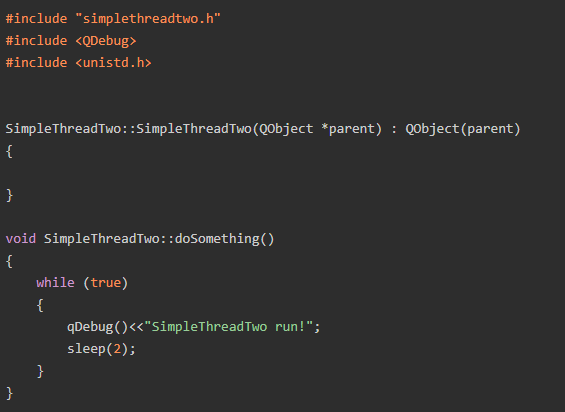


图五：main.cpp

1. 方法二: 使用QObject
2. 创建一个QT应用
3. 创建线程类，继承QObject



图六：simplethreadtwo.h

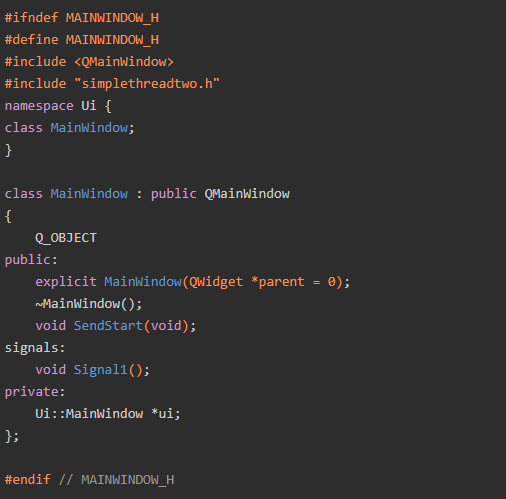


图七：simplethreadtwo.cpp

其中：doSomething 是一个槽函数（功能就是QThread中的run），通过信号调用该槽函数启动循环函数。该信号由主线程发送。

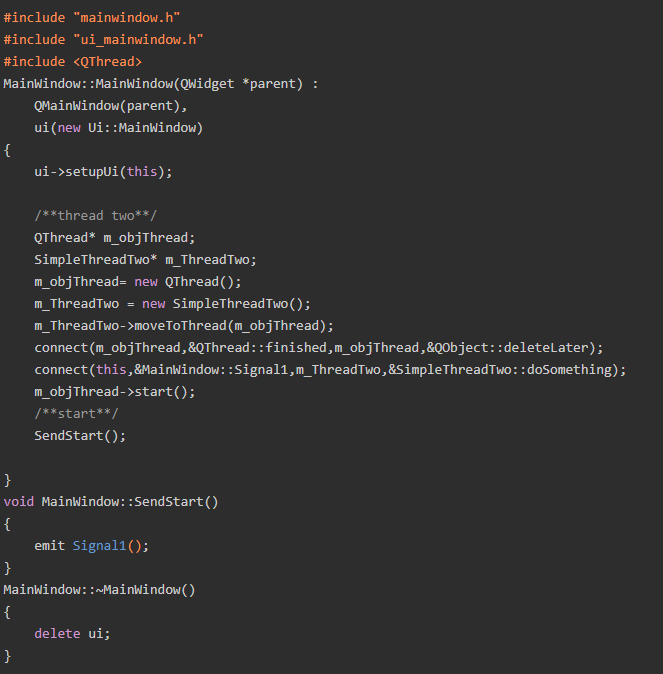
1. 主线程创建线程，拉起新建的线程

一般默认的主线程是mainwindow



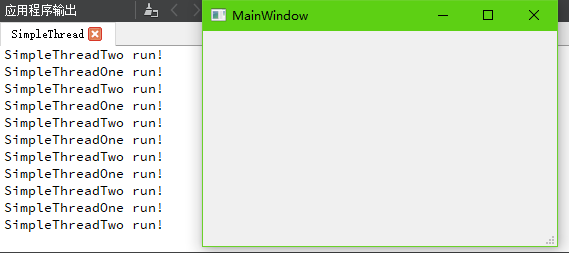
图八：MainWindow.h

主线程通过SendStart函数发送自定义的信号Signal1，从而触发 SimpleThreadTwo 线程的循环函数



图九：MainWindow.cpp

运行结果展示：



图十：运行结果