<https://blog.csdn.net/octdream/article/details/77587530>

在最近的一个项目中，遇到了一个问题，就是主界面调用一个线程，然后再线程中开启一个线程池进行数据生成，线程池调用的线程对象必须继承自QRunable类，这个类有个缺点，就是因为它无法继承QObject，所以不能向外面发送信号，但是我们需要在主界面显示它输出的信息。怎么办呢？

* **编写一个QRunable子类**
* **编写一个QThread子类**
* **调用QThread子类**

**编写一个QRunable子类**

**编写一个QRunable子类MyRunable**

MyRunable.h代码，如下：

#ifndef MYRUNNABLE\_H

#define MYRUNNABLE\_H

#include <QRunnable>

#include <QMetaObject>

class MyRunnable : public QRunnable

{

public:

MyRunnable(QObject \*parent = 0);

~MyRunnable();

void run();

void setID(const int &id);

//向外传送消息

void requestMsg(const QString &msg);

private:

//父对象

QObject \*mParent;

int runnableID;

};

#endif // MYRUNNABLE\_H

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26

MyRunable.cpp代码，如下：

#include "myrunnable.h"

#include <QDebug>

#include <QThread>

MyRunnable::MyRunnable(QObject \*parent)

: QRunnable()

{

mParent = parent;

}

MyRunnable::~MyRunnable()

{

runnableID = 0;

}

void MyRunnable::setID(const int &id)

{

runnableID = id;

}

void MyRunnable::requestMsg(const QString &msg)

{

QMetaObject::invokeMethod(mParent, "requestMsg", Qt::QueuedConnection, Q\_ARG(QString, msg));

}

void MyRunnable::run()

{

for(int i = 0;i < 10;i++)

{

requestMsg(QString("this is a MyRunnable %1").arg(runnableID));

QThread::sleep(1);

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33

**编写一个QThread子类**

**编写一个QThread子类MyThread**

MyThread.h代码，如下：

#ifndef MYTHREAD\_H

#define MYTHREAD\_H

#include <QThread>

class MyThread : public QThread

{

Q\_OBJECT

public:

explicit MyThread(QObject \*parent = 0);

protected:

void run();

signals:

void requestMsg(const QString &msg);

public slots:

};

#endif // MYTHREAD\_H

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23

MyThread.cpp代码，如下：

#include "mythread.h"

#include <QThreadPool>

#include "myrunnable.h"

#include <QDebug>

MyThread::MyThread(QObject \*parent) :

QThread(parent)

{

}

void MyThread::run()

{

qDebug() << "MyThread";

QThreadPool myPool;

myPool.setMaxThreadCount(4);

for(int i = 0;i < 4;i++)

{

MyRunnable \*subThread = new MyRunnable(this);

subThread->setID(i);

myPool.start(subThread);

}

myPool.waitForDone();

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24

**调用QThread子类**

**在MainWindow中调用MyThread并关联信号槽**

MainWindow .h代码，如下：

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include "mythread.h"

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~MainWindow();

private slots:

void showMsg(const QString &msg);

private:

Ui::MainWindow \*ui;

MyThread myThread;

};

#endif // MAINWINDOW\_H

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26

MainWindow .cpp代码，如下：

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QDebug>

#include <QDateTime>

#include <QThreadPool>

#include <myrunnable.h>

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

connect(&myThread,SIGNAL(requestMsg(const QString&)),this,SLOT(showMsg(const QString&)));

myThread.start();

}

void MainWindow::showMsg(const QString &msg)

{

qDebug()<< msg;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21

**运行结果**

MyThread

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"

"this is a MyRunnable 0"

"this is a MyRunnable 1"

"this is a MyRunnable 2"

"this is a MyRunnable 3"