在Qt（C++）中使用QThread实现多线程

1. 引言

多线程对于需要处理耗时任务的应用很有用，一方面响应用户操作、更新界面显示，另一方面在"后台"进行耗时操作，比如大量运算、复制大文件、网络传输等。
使用Qt框架开发应用程序时，使用QThread类可以方便快捷地创建管理多线程。而多线程之间的通信也可使用Qt特有的"信号-槽"机制实现。
下面的说明以文件复制为例。主线程负责提供交互界面，显示复制进度等；子线程负责复制文件。最后附有可以执行的代码。

2. QThread使用方法1——重写run()函数

第一种使用方法是自己写一个类继承QThread，并重写其 `run()` 函数。
大家知道，C/C++程序都是从 `main()` 函数开始执行的。 `main()` 函数其实就是主进程的入口， `main()` 函数退出了，则主进程退出，整个进程也就结束了。
而对于使用Qthread创建的进程而言， `run()` 函数则是新线程的入口， `run()` 函数退出，意味着线程的终止。复制文件的功能，就是在run()函数中执行的。
下面举个文件复制的例子。自定义一个类，继承自Qthread

```
CopyFileThread: public QThread
{
    Q_OBJECT
public:
    CopyFileThread(QObject * parent = 0);

protected:
    void run(); // 新线程入口
// 省略掉一些内容
}
```

在对应的cpp文件中，定义run()

```
void CopyFileThread::run()
{
    // 新线程入口
    // 初始化和操作放在这里
}
```

将这个类写好之后，在主线程的代码中生成一个CopyFileThread的实例，例如在mainwindow.cpp中写：

```
// mainwindow.h中
CopyFileThread * m_cpyThread;

// mainwindow.cpp中
m_cpyThread = new CopyFileThread;
```

在要开始复制的时候，比如按下"复制"按钮后，让这个线程开始执行：

```
m_cpyThread->start();
```

注意，使用start()函数来启动子线程，而不是run()。start()会自动调用run()。
线程开始执行后，就进入run()函数，执行复制文件的操作。而此时，主线程的显示和操作都不受影响。
如果需要进行对复制过程中可能发生的事件进行处理，例如界面显示复制进度、出错返回等等，应该从CopyFileThread中发出信号(signal），并事先连接到mainwindow的槽，由这些槽函数来处理事件。

3. QThread使用方法2——moveToThread()

如果不想每执行一种任务就自定义一个新线程，那么可以自定义用于完成任务的类，并让它们继承自QObject。例如，自定义一个FileCopier类，用于复制文件。

```
class FileCopier : public QObject
{
    Q_OBJECT
```

公告

昵称： 星夜之夏
园龄： 3年8个月
粉丝： 10
关注： 1
+加关注

< 2019年12月 >

日	一	二	三	四	五	六
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	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

C++(9)
Qt(9)
Linux(8)
Windows(6)
ubuntu(5)
QtCreator(5)
LeetCode(5)
共享文件夹(2)
Python(2)
QIF(1)
更多

随笔分类

LeetCode(5)
Linux系统(7)
Qt(8)
网络技术(1)

随笔档案

2019年8月(1)

```
public:
    explicit FileCopier(QObject *parent = 0);

public slots:
    void startCopying();
    void cancelCopying();
}
```

注意这里我们定义了两个槽函数，分别用于复制的开始和取消。
这个类本身的实例化是在主线程中进行的，例如：

```
// mainwindow.h中
private:
    FileCopier* m_copier;

// mainwindow.cpp中，初始化时
    m_copier = new FileCopier;
```

此时m_copier还是属于主线程的。要将其移动到子线程处理，需要首先声明并实例化一个QThread：

```
// mainwindow.h中
signals:
    void startCopyRsquested();
private:
    QThread * m_childThread; // m_copier将被移动到此线程执行

// mainwindow.cpp中，初始化时
    m_childThread = new QThread; // 子线程，本身不负责复制
```

然后使用moveToThread()将m_copier移动到新线程。注意moveToThread()是QObject的公有函数，因此用于复制文件的类FileCopier必须继承自QObject。移动之后启动子线程。此时复制还没有开始。

```
    m_copier->moveToThread(m_childThread); // 将实例移动到新的线程，实现多线程运行
    m_childThread->start(); // 启动子线程
```

注意一定要记得启动子线程，否则线程没有运行，m_copier的功能也无法执行。
要开始复制，需要使用信号-槽机制，触发FileCopier的槽函数实现。因此要事先定义信号并连接：

```
// mainwindow.h中
signals:
    void startCopyRsquested();
// mainwindow.cpp中，初始化时
// 使用信号-槽机制，发出开始指令
    connect(this, SIGNAL(startCopyRsquested()), m_copier, SLOT(startCopying()));
```

当按下"复制"按钮后，发出信号。

```
    emit startCopyRsquested(); // 发送信号
```

m_copier在另一个线程接收到信号后，触发槽函数，开始复制文件。

4.常见问题

4.1. 子线程中能不能进行UI操作?

Qt中的UI操作，比如QMainWindow、QWidget之类的创建、操作，只能位于主线程!
这个限制意味着你不能在新的线程中使用QDialog、QMessageBox等。比如在新线程中复制文件出错，想弹出对话框警告?
可以，但是必须将错误信息传到主线程，由主线程实现对话框警告。
因此一般思路是，主线程负责提供界面，子线程负责无UI的单一任务，通过"信号-槽"与主线程交互。

4.2. QThread中的哪些代码属于子线程?

QThread，以及继承QThread的类（以下统称QThread），他们的实例都属于新线程吗？答案是：不。
需要注意的是，QThread本身的实例是属于创建该实例的线程的。比如在主线程中创建一个QThread，那么这个QThread实例本身属于主线程。当然，QThread会开辟一个新线程（入口是 `run()` ），但是QThread本身并不属于这个新线程。也就是说，QThread本身的成员都**不属于**新线程，而且在QThread构造函数里通过 `new` 得到的实例，也**不属于**新线程。这一特性意味着，如果要实现多线程操作，那么你希望属于新线程的实例、变量等，应该在 `run()` 中进行初始化、实例化等操作。本文给出的例子就是这样操作的。
如果你的多线程程序运行起来，会出现关于thread的报警，思考一下，各种变量、实例是不是放对了位置，是不是真的位于新的线程里。

4.3. 怎么查看是不是真的实现了多线程?

可以打印出当前线程。对于所有继承自QObject的类，例如QMainwindow、QThread，以及自定义的各种类，可以调用 `QObject::thread()` 查看当前线程，这个函数返回的是一个QThread的指针。例如用qDebug()打印：
在mainwindow.cpp的某个函数里、QThread的run()函数里、自定义类的某个函数里，写上：

2019年7月(3)
2019年6月(2)
2019年3月(6)
2019年1月(2)
2018年12月(2)
2018年1月(1)
2017年11月(2)
2017年8月(1)
2017年6月(1)
2017年5月(5)
2017年2月(2)
2016年4月(1)

最新评论

1. Re:解决virtualbox共享文件夹没有访问权限的问题
good
--SoulsFly
2. Re:在Qt中使用SQLite数据库
给你点个赞
--JackeyLea
3. Re:在Qt(C++)中与Python混合编程
你好，为什么我包装的的C++类不会进释放函数呢？还有编译出来的DLL只看到实例化没看到释放。
--漂泊的浮萍
4. Re:在Qt中使用SQLite数据库
写的超级好
--榛子狸
5. Re:Qt程序无法输入中文的问题
复制进去了还是没有效果
--liouvilles

阅读排行榜

1. 在Qt中使用SQLite数据库(54769)
2. 在Linux下访问Windows共享文件夹(23344)
3. 在Qt(C++)中与Python混合编程(17472)
4. 在Notepad++中快捷选中多行(17019)
5. Ubuntu下搜狗拼音输入法打不出汉字的解决方法(9355)

评论排行榜

1. Ubuntu下搜狗拼音输入法打不出汉字的解决方法(4)
2. 在Qt中使用SQLite数据库(3)
3. Qt程序无法输入中文的问题(1)
4. 在Qt(C++)中与Python混合编程(1)
5. 解决virtualbox共享文件夹没有访问权限的问题(1)

推荐排行榜

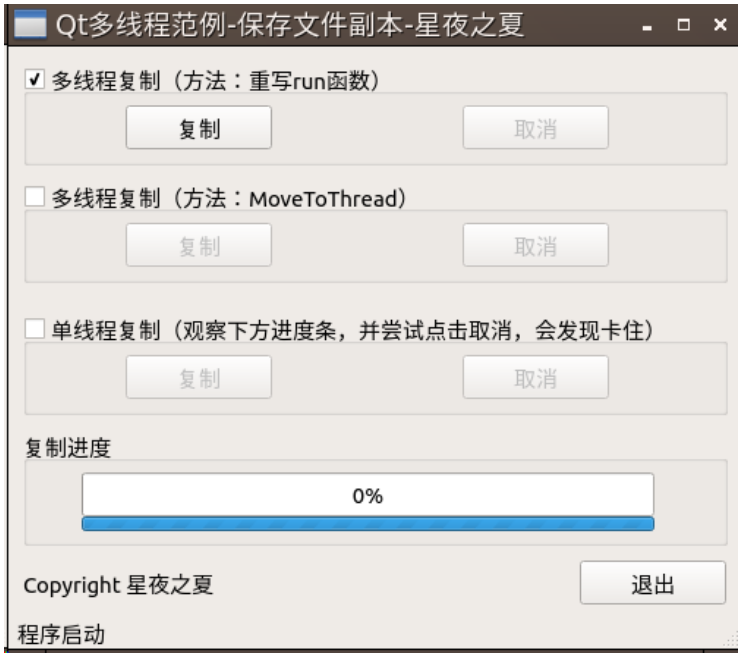
1. 在Qt中使用SQLite数据库(9)
2. Ubuntu下搜狗拼音输入法打不出汉字的解决方法(4)
3. 在Notepad++中快捷选中多行(2)
4. 解决virtualbox共享文件夹没有访问权限的问题(1)
5. 机械+固态双硬盘时机械硬盘卡顿问题解决(1)

```
qDebug() << "Current thread:" << thread();
```

对比不同位置打印的指针，就可以知道它们是不是位于同一个线程了。

5.范例

范例实现了多线程复制文本文件。
提供的范例文件可用QtCreator编译运行。界面如下（不同的操作系统略有不同）：



范例中实现了本文介绍的两种方法，同时也给出了单线程复制对比。打钩选择不同的复制方法。可以发现，在使用多线程的时候，界面不会假死，第二根进度条的动画是持续的；而使用单线程复制的时候，“取消”按钮按不动，界面假死，而且第二根进度条的动画也停止了。
由于范例处理的文件很小，为了让复制过程持续较长时间以便使得现象明显，复制文件的时候，每复制一行加入了等待。

范例代码：
https://github.com/Xia-Weiwen/CopyFile

分类: Linux系统, Qt

标签: Qt, 多线程, C++

好文要顶 关注我 收藏该文

星夜之夏
关注 - 1
粉丝 - 10
+加关注

1 0

« 上一篇： 在Linux（Ubuntu）下安装Arial、Times New Roman等字体
» 下一篇： LeetCode做题笔记 - 66 - 加一（简单）

posted @ 2019-03-02 22:18 星夜之夏 阅读(4286) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册， 访问 网站首页。

【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
【推荐】腾讯云热门云产品限时秒杀，爆款1核2G云服务器99元/年!
【推荐】阿里云双11返场来袭，热门产品低至一折等你来抢!
【推荐】天翼云双十一翼降到底，云主机11.11元起，抽奖送大礼
【推荐】流程自动化专家UiBot，体系化教程成就高薪RPA工程师
【活动】京东云服务器_云主机低于1折，低价高性能产品备战双11
【优惠】七牛云采购嘉年华，云存储、CDN等云产品低至1折

相关博文：

· QT的多线程使用
· Qt 线程基础(QThread、Qtconcurrent等)
· QT之深入理解QThread
· Qt多线程学习：创建多线程
· Qt在多线程中使用信号槽的示例
» 更多推荐...

《Flutter in action》开放下载！闲鱼Flutter企业级实践精选

最新 IT 新闻:

· 11 月全球 Web 服务器调查报告：nginx 表现最佳
· DeepMind联合创始人加入谷歌 负责AI政策方面工作
· 长鑫存储：收获大量原奇梦达内存专利
· 高通做笔电芯片：主打长续航和低价格 挑战英特尔
· 高通谷歌联手：安卓手机最快明年可取代身份证和驾照
» 更多新闻...

Copyright © 2019 星夜之夏
Powered by .NET Core 3.1.0 on Linux