Linux线程并发

使用Qt编写程序，同时画圆与画方

华中科技大学软件学院2003班

2020/3/31

侯皓斐

U202010851

01

实验环境

01

01

明德　　厚学　　求是　　创新

实验环境

- 使用虚拟机环境：VMWare Pro 16
- Ubuntu 20.04.4 LTS
- Qt 5.9.9版本

02

实验目的

02

02

明德　　厚学　　求是　　创新

实验目的

- 使用Qt，在Linux环境下用线程实现同时画圆和画方形。

03

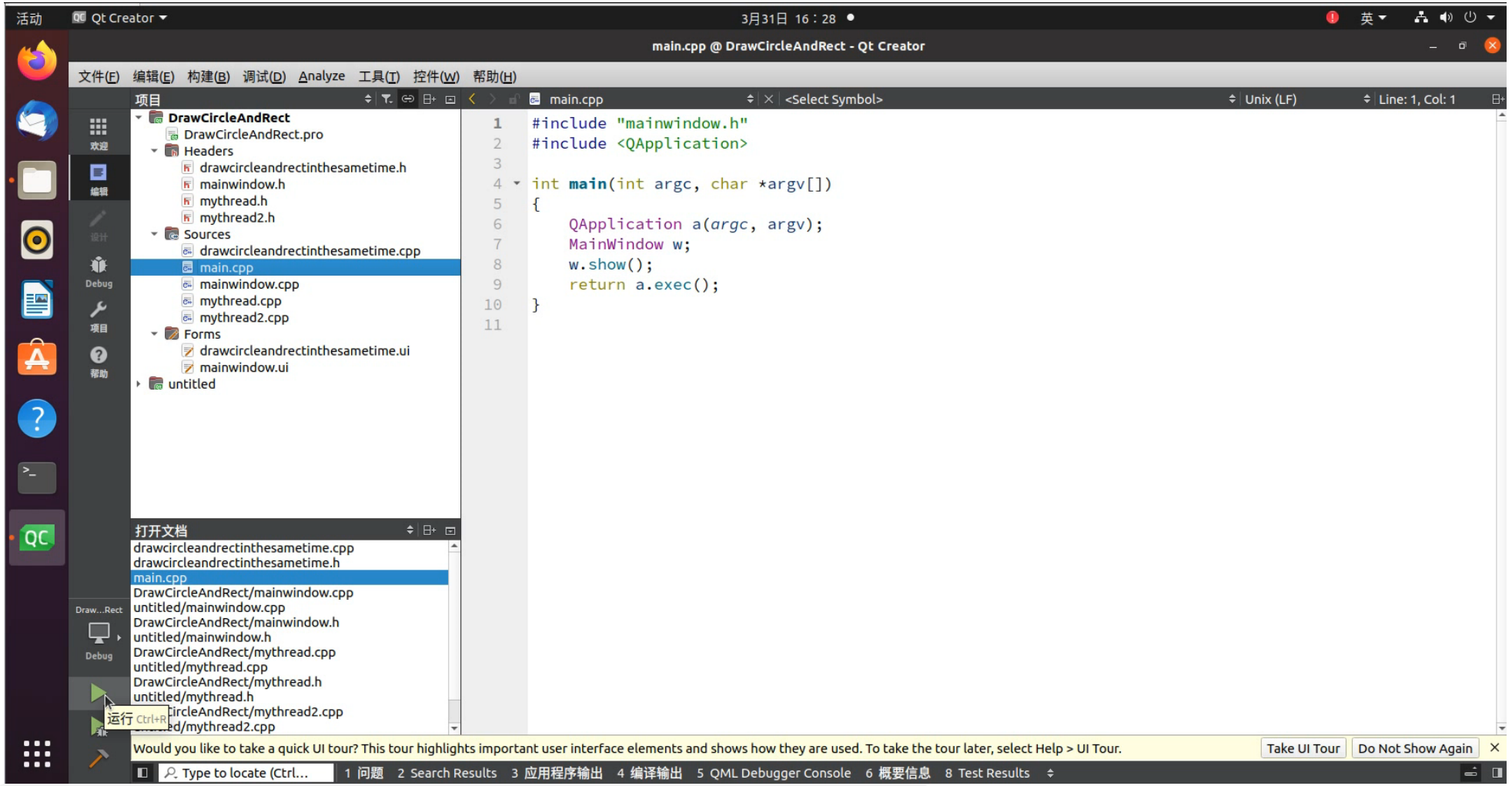
效果展示

03

03

明德　厚学　求是　创新

效果展示


活动
Qt Creator
3月31日 16：28
英
main.cpp @ DrawCircleAndRect - Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) Analyze 工具(T) 控件(W) 帮助(H)
项目
DrawCircleAndRect
DrawCircleAndRect.pro
Headers
drawcircleandrectinthesametime.h
mainwindow.h
mythread.h
mythread2.h
Sources
drawcircleandrectinthesametime.cpp
main.cpp
mainwindow.cpp
mythread.cpp
mythread2.cpp
Forms
drawcircleandrectinthesametime.ui
mainwindow.ui
untitled
main.cpp
<Select Symbol>
Unix (LF)
Line: 1, Col: 1
#include "mainwindow.h"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.show();
return a.exec();
}
打开文档
drawcircleandrectinthesametime.cpp
drawcircleandrectinthesametime.h
main.cpp
DrawCircleAndRect/mainwindow.cpp
untitled/mainwindow.cpp
DrawCircleAndRect/mainwindow.h
untitled/mainwindow.h
DrawCircleAndRect/mythread.cpp
untitled/mythread.cpp
DrawCircleAndRect/mythread.h
untitled/mythread.h
运行 Ctrl+R
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again
Type to locate (Ctrl...
1 问题 2 Search Results 3 应用程序输出 4 编译输出 5 QML Debugger Console 6 概要信息 8 Test Results

04

实验流程

04

04

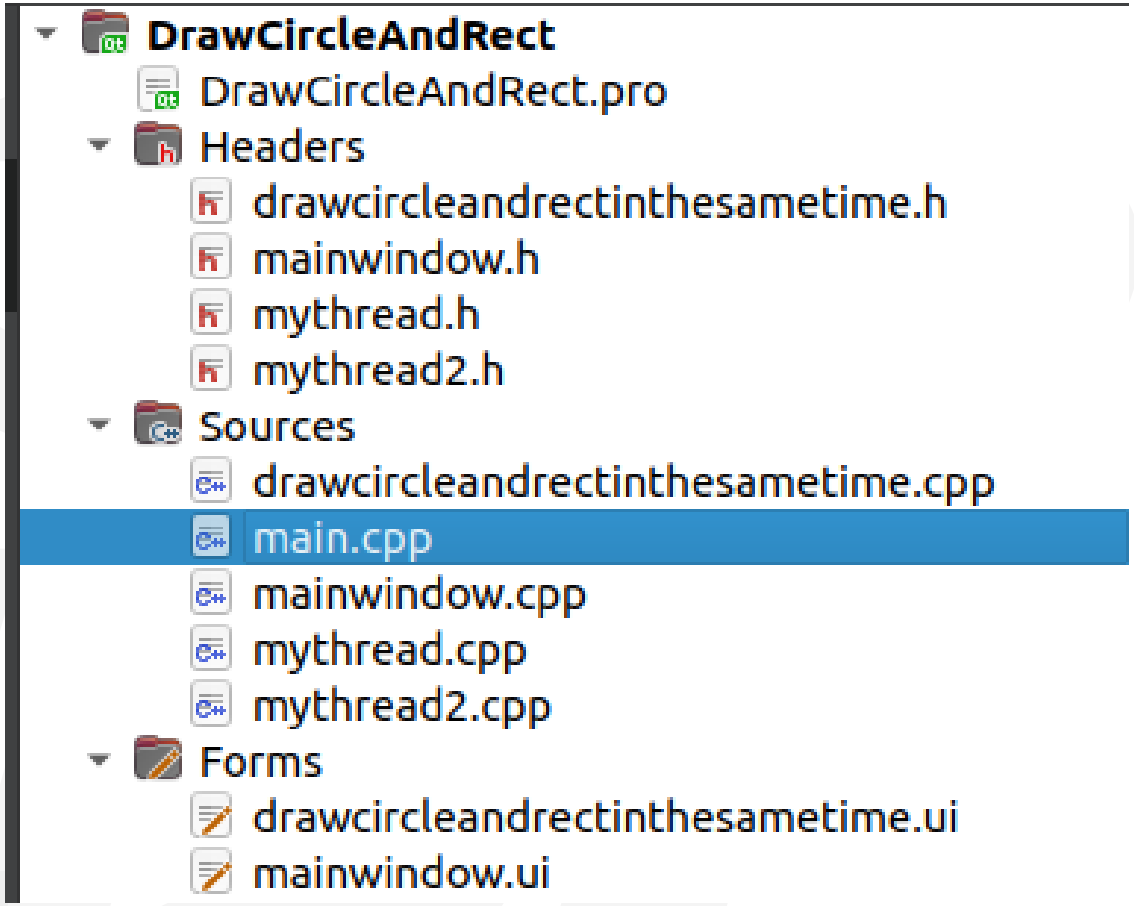
明德　厚学　求是　创新

准备工作

- 按照教程，下载Qt5.9.9并安装。
- 设置环境变量。

代码展示

- Qt项目包含的文件



核心代码展示：页面跳转

- Mainwindow.cpp

```
21  void MainWindow::on_btn_clicked()
22  {
23      DrawCircleAndRectInTheSameTime *draw = new DrawCircleAndRectInTheSameTime();
24      draw->show();
25      this->close();
26  }
27
```

核心代码展示：线程1实现

- mythread.cpp

```
11  void myThread::run(){
12      for(int i = 1; i <= 200; i++) {
13          mutex.lock();
14
15          QPainter* painter = new QPainter(pixmap);//需要动态分配painter，以保证可以释放，保证不会同时画图。
16
17          QColor red(0xFF,0,0);//设置颜色
18          QPen pen(red);//定义画笔
19          pen.setWidth(5);//
20          painter->setPen(pen);
21
22          if(1 <= i && i <= 50)
23              painter->drawLine(100+8*(i-1),100,100+8*i,100);
24          else if(51 <= i && i <= 100)
25              painter->drawLine(500,100+8*(i-51),500,100+8*(i-50));
26          else if(101 <= i && i <= 150)
27              painter->drawLine(500-8*(i-101),500,500-8*(i-100),500);
28          else
29              painter->drawLine(100,500-8*(i-151),100,500-8*(i-150));
30          delete painter;
31
32          mutex.unlock();
33
34          QThread::msleep(SLEEP_TIME_MS);
35          //emit rep();
36      }
37  }
```

核心代码展示：线程2实现

```
6   extern QMutex mutex;
7   myThread2::myThread2(QPixmap *pixmap):
8       pixmap(pixmap)
9   {
10  }
11  void myThread2::run(){
12      for(int i = 1; i <= 180; i++) {
13          mutex.lock();
14
15          QPainter* painter = new QPainter(pixmap);//需要动态分配painter，以保证可以释放，保证不会同时画图。
16
17          QColor blue(0,0,0xFF);//设置颜色
18          QPen pen(blue);//定义画笔
19          pen.setWidth(5);
20          painter->setPen(pen);
21
22          QRectF rect(700, 100, 400, 400);
23
24          painter->drawArc(rect,(i-1)*2*16,2*16);
25
26          delete painter;
27
28          mutex.unlock();
29
30          QThread::msleep(SLEEP_TIME_MS);
31          //emit rep();
32      }
33  }
```

核心代码展示：主进程实现

学在华中大

drawcircleandrectinthesameti... | DrawCircleAndRectInTheSameTime::DrawCircleAndRectInTheSameTi... | Unix (LF)

```
#include "drawcircleandrectinthesametime.h"
#include "ui_drawcircleandrectinthesametime.h"
#include "mythread.h"
#include "mythread2.h"
#include <QThreadPool>
#include <QPainter>
#include <QTimer>
#include <QPixmap>
#define REPAINT_TIME 200

DrawCircleAndRectInTheSameTime::DrawCircleAndRectInTheSameTime(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::DrawCircleAndRectInTheSameTime)
{
    ui->setupUi(this);
    pixmap=new QPixmap(this->size());
    pixmap->fill(Qt::white);
    myThread *m0=new myThread(pixmap);
    QThreadPool::globalInstance()->start(m0);

    myThread2 *m1=new myThread2(pixmap);
    QThreadPool::globalInstance()->start(m1);

    //connect(m0,SIGNAL(rep()),this,SLOT(repaint()));
    //connect(m1,SIGNAL(rep2()),this,SLOT(repaint()));

    QTimer *qt0=new QTimer();
    qt0->setInterval(REPAINT_TIME);
    qt0->start();
    connect(qt0,SIGNAL(timeout()),this,SLOT(onTimeOut()));
}
```

核心代码展示：定时重绘页面

学在華中大

- drawcircleandrectinthesametime.cpp

```
38  void DrawCircleAndRectInTheSameTime::paintEvent(QPaintEvent *){
39      QPainter painter(this);
40      painter.drawPixmap(0,0,*pixmap,0,0,0,0);
41  }
42
```

05

参考资料

05

05

明德　　厚学　　求是　　创新

参考资料

- Linux环境下搭建Qt5.9开发环境 - spfanlost - 博客园 (cnblogs.com)
- (125条消息) Qt：一个按钮实现打开和关闭新窗口_Jingtong--的博客-CSDN博客_qt点击按钮关闭窗口
- (125条消息) Qt多线程基础（一）线程同步之互斥锁同步_CtrlTurtle的博客-CSDN博客_qt 互斥锁
- c++ - 如何销毁QPainter对象/摆脱drawText()内存泄漏？ - IT工具网 (coder.work)
- qt 中画线时如何设置笔的颜色和填充 - 未完代码 - 博客园 (cnblogs.com)
- 28.QT-QPainter介绍 - 诺谦 - 博客园 (cnblogs.com)
- Qt 5.15
- 第04章 进程管理和通信.pdf 华中科技大学苏曙光老师

- 特别感谢silence_heart的博客
- (125条消息) Qt多线程绘图并实时刷新画面的方法_silence_heart的博客-CSDN博客_qt多线程绘图