

C 博客

登录 | 注册

Q

≡

于@IT 学习笔记

≡ 目录视图

≡ 摘要视图

RSS 订阅

个人资料



ychnw

+ 加关注

发私信

访问：19201次

积分：305

等级：BLOG等级2

排名：千里之外

原创：9篇

转载：4篇

译文：0篇

评论：10条

文章搜索

文章分类

C/C++ (8)

Linux (1)

Python (4)

其他 (0)

文章存档

2009年12月 (2)

2009年08月 (1)

2009年07月 (5)

2009年06月 (5)

阅读排行

C++中的静态变量和静态 (3732)

Qt编写的软件在线升级系 (3217)

Qt编写的软件在线升级系 (2359)

Qt在windows下在控制台 (1294)

python中使用自然字符串 (1113)

在arm+linux平台编译Qt4 (959)

Python的os和os.path模块 (904)

Qt特征定义文件（裁剪定 (721)

Python的运算符优先级 (704)


有奖征文：那些年我们一起写过的代码

【博乐】点评美文，得C币

相聚上海，大话iOS开发

恭喜博主出书《

跳吧



CSDN 移动客户端

快速回复

原

Qt编写的软件在线升级系统（服务器端）

分类：C/C++

2009-07-01 14:24

2360人阅读

评论(5)

收藏

举报

服务器

qt

server

file

signal

stream

给公司开发的软件，每次更新后都要一个个的去帮使用的同事重新安装，很麻烦，所以放下几个bug没有解决，先写了一个升级的小程序。

程序入口(main.cpp):

```
[cpp]
01. #include <QtCore>
02. #include <QCoreApplication>
03. #include <QDebug>
04. #include <QTextCodec>
05. #include "Server.h"
06. int main(int argc, char *argv[])
07. {
08.     QCoreApplication a(argc, argv);
09.     QTextCodec::setCodecForLocale(QTextCodec::codecForName("UTF-8"));
10.     QTextCodec::setCodecForCStrings(QTextCodec::codecForName("UTF-8"));
11.     qDebug()
12.     <<"CR_MRP_update:Welcome to ChuangRen MRP/nCR_MRP_Update:Update Server starting...";
13.     Server *server=new Server();
14.     server->listenConnect();
15.     return a.exec();
16. }
```

在主程序中先设置了软件的编码，然后启动一个服务器并监听连接。服务器类的定义如下(Server.h):

```
[cpp]
01. /*
02.  * Server.h
03.  *
04.  * Created on: 2009-5-7
05.  * Author: 于成文
06.  */
07. #ifndef SERVER_H_
08. #define SERVER_H_
09. #include <QDebug>
10. #include <QTcpServer>
11. #include <QTcpSocket>
12. #include <QHostAddress>
13. #include <QFile>
14. #include <QByteArray>
15. #include "ConnectThread.h"
16. class Server : public QTcpServer{
17.     Q_OBJECT
18. public:
19.     Server(QObject *parent=0);
```

C++中使用全局变量 (457)

评论排行

- Qt编写的软件在线升级系 (5)
- Qt编写的软件在线升级系 (4)
- Qt在windows下在控制台 (1)
- 在arm+linux平台编译Qt4 (0)
- Python的os和os.path模块 (0)
- 第一个Python程序 (0)
- Python的运算符优先级 (0)
- python中使用自然字符串 (0)
- fedora 10下安装nvidia显 (0)
- C++中使用全局变量 (0)

推荐文章

最新评论

- Qt编写的软件在线升级系统（客kamio\_misuzu: 正要着手学习这个，谢谢楼主的经验！
- Qt编写的软件在线升级系统（服kamio\_misuzu: 来学习了，顶楼主！
- Qt编写的软件在线升级系统（客ychwn: @fantasysunshinewzp: 谢谢. 刚学Qt时写这玩的, 很多不足, 呵呵.
- Qt编写的软件在线升级系统（客fantasysunshinewzp: 谢谢，很不错
- Qt编写的软件在线升级系统（服maizitian: Thank you！
- Qt编写的软件在线升级系统（客l270378034: 嗯不错
- Qt编写的软件在线升级系统（服匿名用户: 不错，学习！
- Qt编写的软件在线升级系统（服ychwn: Thanks.
- Qt编写的软件在线升级系统（服lanmanck: great!
- Qt在windows下在控制台用qDebalan2888: 学习了!

技术

南华人网站

```
20. ~Server();
21. bool listenConnect();
22. private slots:
23. void incomingConnection(int socketId);
24. private:
25. QString version; //服务器上的最新版本
26. QByteArray file1,file2; //保存客户端要更新的文件
27. };
28. #endif /* SERVER_H_ */
```

下面是Server类的实现(Server.cpp):

```
[cpp]
01. /*
02. * Server.cpp
03. *
04. * Created on: 2009-5-7
05. * Author: 于成文
06. */
07. #include "Server.h"
08. Server::Server(QObject *parent):QTcpServer(parent)
09. {
10.     QFile file("version.dat");
11.     if (file.open(QIODevice::ReadOnly))
12.     {
13.         QTextStream stream(&file);
14.         version = stream.readLine();
15.         file.close();
16.     }
17.     QFile upfile1("MRP.exe");
18.     QFile upfile2("common.dll");
19.     if (upfile1.open(QIODevice::ReadOnly)&&upfile2.open(QIODevice::ReadOnly))
20.     {
21.         file1=upfile1.readAll();
22.         file2=upfile2.readAll();
23.     }
24. }
25. Server::~Server() {
26. }
27. /**
28. * 开始监听
29. */
30. bool Server::listenConnect()
31. {
32.     if(!this->listen(QHostAddress::Any, 3327))
33.     {
34.         qDebug()<<"CR_MRP_Update:listen on port 3327 failed!";
35.         this->close();
36.         return false;
37.     }
38.     qDebug()<<"CR_MRP_Update:listen on port 3327,waiting connection...";
39.     return true;
40. }
41. /**
42. * 当有客户端连接时运行此函数
43. */
44. void Server::incomingConnection(int socketId)
45. {
46.     qDebug()<<"nCR_MRP_Update:a client try to connect...";
47.     //连接客户端
48.     ConnectThread * thread = new ConnectThread(this, socketId, version, file1, file2);
49. }
```



CSDN 移动客户端

快速回复

在Server类的构造函数中读取了当前服务器上的客户端软件的版本信息，并将客户端读入内存。当有客户端连接是会自动运行函数incomingConnection(int socketId)，这个函数中初始化了一个ConnectThread类的对象，并将各种信息通过构造函数传递给它。

下面是类ConnectThread的定义和实现():

[cpp]

```
01.  /*
02.  * ConnectThread.h
03.  *
04.  * Created on: 2009-5-7
05.  * Author: 于成文
06.  */
07. #ifndef CONNECTTHREAD_H_
08. #define CONNECTTHREAD_H_
09. #include <QDebug>
10. #include <QTcpSocket>
11. #include <QDataStream>
12. #include <QDate>
13. #include <QByteArray>
14. #include <QFile>
15. class ConnectThread : public QObject
16. {
17.     Q_OBJECT
18. public:
19.     ConnectThread(QObject * parent = 0 ,int socketId=0,const QString &version="",QByteArray
20.     ~ConnectThread();
21. private:
22.     int socketId;
23.     QTcpSocket *clientSocket;
24.     quint64 nextBlockSize;
25.     QString version;
26.     QByteArray file1,file2;
27. private slots:
28.     void threadOver();
29.     void communicate();
30. };
31. #endif /* CONNECTTHREAD_H_ */
```



CSDN 移动客户端

快速回复

[cpp]

```
01.  /*
02.  * ConnectThread.cpp
03.  *
04.  * Created on: 2009-5-7
05.  * Author: 于成文
06.  */
07. #include "ConnectThread.h"
08. ConnectThread::ConnectThread(QObject * parent,int socketId,const QString &version,QByteAr
09. {
10.     nextBlockSize=0;
11.     clientSocket=new QTcpSocket(this);
12.     clientSocket->setSocketDescriptor(socketId);
13.     this->version=version;
14.     this->file1=file1;
15.     this->file2=file2;
16.     connect(clientSocket, SIGNAL(disconnected()),this, SLOT(threadOver()));
17.     connect(clientSocket,SIGNAL(readyRead()),this,SLOT(communicate()));
18.     qDebug()<<"CR_MRP_Update:connect to server successful,client id is"<<socketId;
19.     //发送当前服务器上的客户端版本信息给升级客户端
20.     QByteArray block;
21.     QDataStream out(&block, QIODevice::WriteOnly);
22.     out.setVersion(QDataStream::Qt_4_5);
23.     out<<quint64(0)<<quint16(0x0001)<<version;
24.     out.device()->seek(0);
25.     out<<quint64(block.size()-sizeof(quint16));
26.     clientSocket->write(block);
27. }
28. ConnectThread::~ConnectThread() {
29. }
30. /**
31.  * 和客户端通信
32.  */
33. void ConnectThread::communicate()
34. {
35.     //创建数据流
```

```
36.     QDataStream in(clientSocket);
37.     in.setVersion(QDataStream::Qt_4_5);
38.     while(1)
39.     {
40.         if (nextBlockSize == 0)
41.         {
42.             if (clientSocket->bytesAvailable() < sizeof(quint64))
43.             {
44.                 break;
45.             }
46.             in >> nextBlockSize;
47.         }
48.         if ((quint64)(clientSocket->bytesAvailable())<nextBlockSize)
49.         {
50.             break;;
51.         }
52.         quint16 protocol;
53.         in>>protocol;
54.         qDebug()<<"CR_MRP_Update:protocol:"<<QString::number(protocol,16);
55.         if(protocol==1)
56.         {
57.             //升级
58.             QByteArray fblock;
59.             QDataStream fout(&fblock, QIODevice::WriteOnly);
60.             fout.setVersion(QDataStream::Qt_4_5);
61.             fout<<quint64(0)<<quint16(0x0002)<<file1<<file2;
62.             fout.device()->seek(0);
63.             fout<<quint64(fblock.size()-sizeof(quint64));
64.             clientSocket->write(fblock);
65.             qDebug()<<"CR_MRP_Update:size:"<<QString::number(fblock.size()-
sizeof(quint64));
66.         }
67.         nextBlockSize=0;
68.     }
69. }
70. /**
71.  * 释放网络连接资源
72.  */
73. void ConnectThread::threadOver()
74. {
75.     qDebug()<<"CR_MRP_Update:a client connection closed.../n";
76.     clientSocket->deleteLater();
77.     this->deleteLater();
78. }
```



版权声明：本文为博主原创文章，未经博主允许不得转载。



- ^ 上一篇

C++中使用全局变量
- v 下一篇

Qt编写的软件在线升级系统（客户端）

主题推荐

- bug
- qt
- 服务器
- 软件
- 安装
- 开发

猜你在找

- Qt基础与Qt on Android入门

基于Qt的二维码生成器
- JavaScript for Qt Quick(QML)

QML中webview元素的JavaScriptWindowObjects的使用
- Python自动化开发基础 分布式监控系统开发实战 day8

Qt编译mysql驱动
- iOS即时通讯(IM)开发实战篇-基于XMPP的聊天软件开发

如何在QML中使用camera API来拍照
- Winform项目实战在线考试系统开发

LINUX总结第13篇LINUX下动态库及版本号控制

准备好了么？！

更多职位尽在 CSDN JOB

<div>服务器端PHP工程师/PHP程序员</div> <div>我要跳槽</div>	<div>服务器端工程师（PHP）</div> <div>我要跳槽</div>
<div>深圳市迈盛达信息技术有限公司</div> <div>10-20K/月</div>	<div>北京华泰中盛科技有限公司</div> <div>12-20K/月</div>

我要跳槽

二

科

我要跳槽

高级研发经理/研发总监（服务器端）

武汉悦然心动网络科技有限公司

8-16K/月

TCL通讯科技控股有限公司

20-30K/月

查看评论

5楼

kamio\_misuzu

2013-05-27 14:55发表

来学习了，顶楼主！

4楼

maizitian

2012-07-25 14:25发表

Thank you !

CSDN 移动客户端

快速回复

3楼

匿名用户

2010-02-21 21:22发表

不错，学习！

2楼

ychwn

2009-12-07 10:35发表

Thanks.

1楼

lanmanck

2009-10-11 15:40发表

great!

您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptch

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap