

# Qt 构建北京市精品课程平台文件上传系统

周恕义<sup>1</sup>, 徐晓亮<sup>2</sup>

(1. 北京工业大学 现代教育技术中心, 北京 100124;

2. 北京工业大学 计算机学院, 北京 100124)

**摘 要:**文中分析了目前北京市精品课程平台文件上传部分的不足,有针对性地设计了基于 Qt 框架的 FTP 上传系统。系统分为服务器端与客户端。服务器端分为三个功能模块,模块之间相互独立,并发运行,通过 OCI 的方式实现与数据库之间的交互。客户端设计的界面友好,简单易用。客户端与服务器端之间通过 Socket 通讯,并将数据进行加密处理,保证了系统的安全性。系统整体运行高效稳定,方便了教师用户以及管理员,并且设计的系统解决了 Qt 框架中无法实现断点续传的功能。

**关键词:**Qt;RFC765;FTP;断点续传

**中图分类号:**G434

**文献标识码:**A

**文章编号:**1673-629X(2011)05-0202-04

## FTP Upload System for Beijing Quality Course Based on Qt

ZHOU Shu-yi<sup>1</sup>, XU Xiao-liang<sup>2</sup>

(1. Modern Technological Center in Education, Beijing University of Technology, Beijing 100124, China;

2. Department of Computer Science, Beijing University of Technology, Beijing 100124, China)

**Abstract:** Analyse the current disadvantage of the file upload function of Beijing Quality Course, and then introduce the designed FTP upload system based on Qt. The system is insisted of three independent parts which running parallel and using OCI to exchange data with database. It is friendly and easily to use the client. Socket is used to communicate between client and server. The data will be encrypting when they are transfer in network. It will ensure the system safety. The whole system facilitates the users and managers and runs stably and effect. The designed system solved Qt which does not include the function of uploading file from break point.

**Key words:** Qt;RFC765;FTP;upload from break point

## 0 引 言

教师在使用北京市精品课程平台时,需要上传视频文件并添加相应的视频信息,目前所采用的方法是教师用户向管理员申请开通 FTP 空间,获得 FTP 账号之后使用第三方 FTP 上传工具登陆 FTP 服务器,进行文件上传,完成之后,登陆网站进行视频信息填写,完成之后视频信息才会在平台中呈现。目前这种方式过程繁琐,教师需要分别使用 FTP 上传账号,网站访问账号才能完成上传视频以及填写视频信息的工作,而管理员也需要参与其中,处理教师用户的 FTP 空间申请。

因此一套方便教师用户使用,减轻管理员负担,并集成文件上传与信息填写功能的文件上传系统显得很有必要。Qt 作为流行的跨平台开源 C++ 设计框架,已

经成功应用于多个领域,其高效简便的开发方式获得了大量的好评<sup>[1]</sup>。因此,笔者设计了基于 Qt 框架的文件上传系统。

## 1 系统架构

教师通过运行软件,使用课程平台的用户名与密码,完成登陆验证,进而上传文件与填写信息。系统架构为客户端/服务器模式。通讯采用异步 Socket<sup>[2]</sup>。网络架构如图 1 所示。

### 1.1 客户端

教师用户在网站下载客户端,使用网站登陆信息登陆到服务器,即可完成上传,添加信息,无需再向管理员申请开通 FTP。流程如图 2 所示。

客户端通过 Socket,发送加密过的用户名密码到 Socket 服务器,服务器获取之后进行解密,验证成功之后,返回 FTP 服务器登录名、密码以及课程信息。客户端连接 FTP 服务器,检测 FTP 中是否有以课程 ID 命名的文件夹,有则进入课程 ID 文件夹目录,否则创

收稿日期:2010-09-18;修回日期:2010-12-24

基金项目:北京市重点计划项目(JC310999200901)

作者简介:周恕义(1955-),男,黑龙江人,教授,在网络多媒体、教育技术领域从事教学、科研和管理工作。

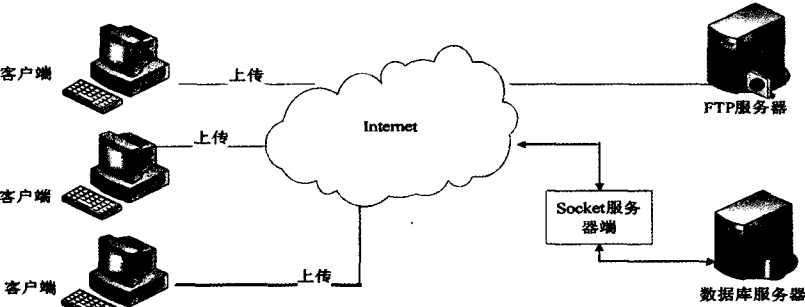


图 1 系统架构

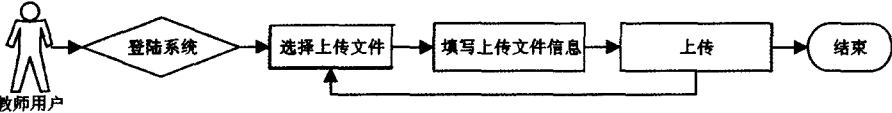


图 2 客户端用户流程

建文件夹,这种方式虽然所有的客户端登陆 FTP 服务器都是用同样的用户名与密码,但却被限制在以课程 ID 命名的文件夹当中,教师用户只可以对其负责的课程文件夹进行操作,并不影响其他用户,同时在网络传输中对账户信息采用加密方式保证了安全性。流程如图 3 所示。

功能描述:用户登陆;显示课程信息;选择文件;填写视频信息;FTP 上传、断点续传;显示上传任务列表、保存列表;获取 FTP 目录并显示。

1.2 服务器端

服务器端分为 FTP 服务器端与 Socket 服务器端两个部分。

2 系统实现

系统基于 Qt4.6.2 for Windows 进行构建,IDE 使用 Qt creator。

1.2.1 FTP 服务器端  
采用目前使用的 Serv-U ftp 服务器。

1.2.2 Socket 服务器端  
作为客户端发送数据的接收方,负责验证用户登陆信息,将视频信息写入数据库,以及监听上传状态的作用。服务器功能流程如图 4 所示。

服务器主要分为三个功能模块。第一,接收客户端发送的用户名和密码信息,向数据库服务器进行验证,验证成功,发送课程信息;否则发送验证错误信息。第二,服务器接收客户端发送的视频信息,并将其写入数据库。第三,获取上传状态信息,当客户端上传完成时,发送指令给服务器,服务器接收并将数据库中的上传状态置为完成。

功能描述:验证用户登陆并发送响应课程信息;获取视频信息并写入数据库;获取上传状态指令并更新数据库。

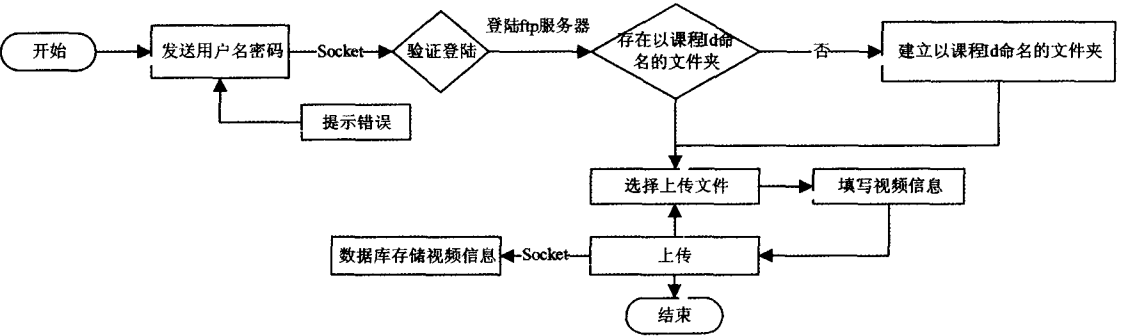


图 3 客户端后台流程

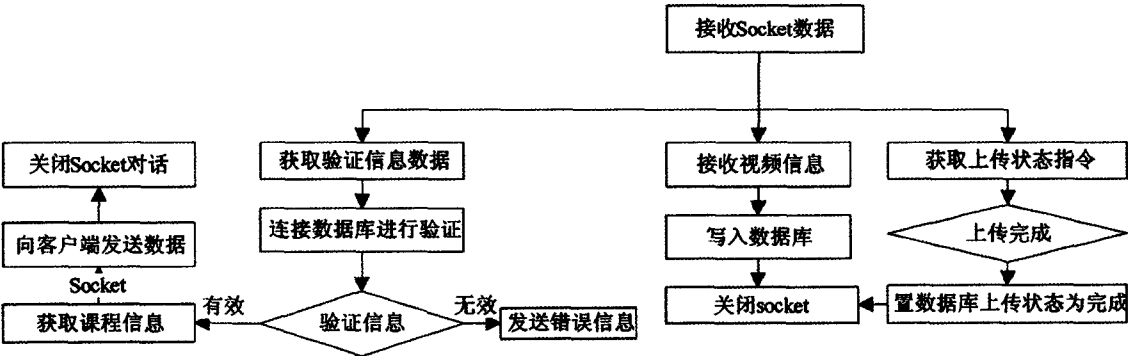


图 4 Socket 服务器端流程

## 2.1 客户端实现

客户端的核心内容主要包括两个模块:Socket 通讯模块;FTP 上传与断点续传模块。

### 2.1.1 Socket 通讯模块

本模块主要实现的功能为:将用户账户信息发送到 Socket 服务器端并接收验证信息;发送视频信息给服务器;发送上传完成信息。

Qt 中提供的 QTcpSocket<sup>[3]</sup> 完全使用了类的封装机制,采用 Tcp 协议进行通讯,不需要接触底层的各种结构体操作,简化了操作。

类 socketClient,功能为创建 Socket 连接,读取/写入 Socket。

其中主要包括成员变量 QTcpSocket client,作为 Socket 连接,以及如下几个功能函数:

槽函数: void startTransfer(), 此函数与 client 的 connected() 信号连接,当 Socket 连接建立后,被触发。使用 QDataStream 序列化数据后用 write 写入 Socket,进行发送。

槽函数: void startRead(), 与 client 的 readyRead() 信号连接,当 Socket 可读时,被触发,使用 QDataStream 对数据进行反序列化,读取 Socket 中的数据。

信号: void StringListRead( QStringList SocketInfo), 将读出的 Socket 信息以信号方式发送,以供其他函数进行数据处理。

具体流程如图 5 所示。

### 2.1.2 FTP 上传与断点续传

FTP 协议的数据传输分为主动模式(PORT)<sup>[4]</sup> 和被动模式(PASV)<sup>[5]</sup>。主动模式下,服务器端使用固定的 20 端口与客户端建立数据连接,数据连接是由服务器发起的。在被动模式下,数据连接是由客户端程序发起的,服务器端发送给客户端一个临时的 Socket 端口号,使用此端口号客户端建立数据连接通道,进行数据传输。由于主动模式下是服务器向客户端发起的连接,常会被客户端的防火墙软件阻挡,导致通讯失败。因此本案设计的上传系统采用被动模式进行通讯。

Qt 提供的 QFtp 类与 FTP<sup>[6]</sup> 协议配合使用,让文件的上传和下载变得更加容易。QFtp 类是异步工

作的。若调用一个像 get() 或者 put() 这样的函数,它会立即返回并且仅在控制权回到 Qt 的事件循环式才发生数据传输。这样就确保了在执行 FTP 指令时,用户界面可以保持响应。

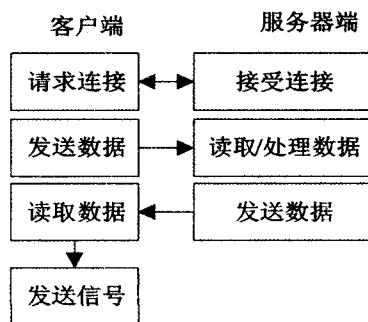


图 5 Socket 通讯

QFtp<sup>[7]</sup> 提供了一些 FTP 指令,包括 connectToHost()、login()、close()、list()、cd()、get()、put()、remove()、mkdir()、和 rename()。所有的这些函数都可以调度一个 FTP 指令,并且可以返回一个标识这个指令的 ID<sup>[8]</sup> 号。其中,rawCommand() 可以执行任意的 FTP 指令。

但在断点续传方面,QFtp 并没有给出接口进行可以直接调用,因此,笔者根据 RFC765<sup>[9]</sup> 中规定的 FTP 协议命令,配合使用 QFtp 的 rawCommand() 函数,实现了断点续传的功能。

主要流程如图 6 所示。

类 ftpclient,内有方法 upload 与 reupload,分别为上传与断点续传的函数接口。

直接上传方面,QFtp 封装了接口 put(文件指针,文件名),函数 upload 进行调用便可实现上传的功能。

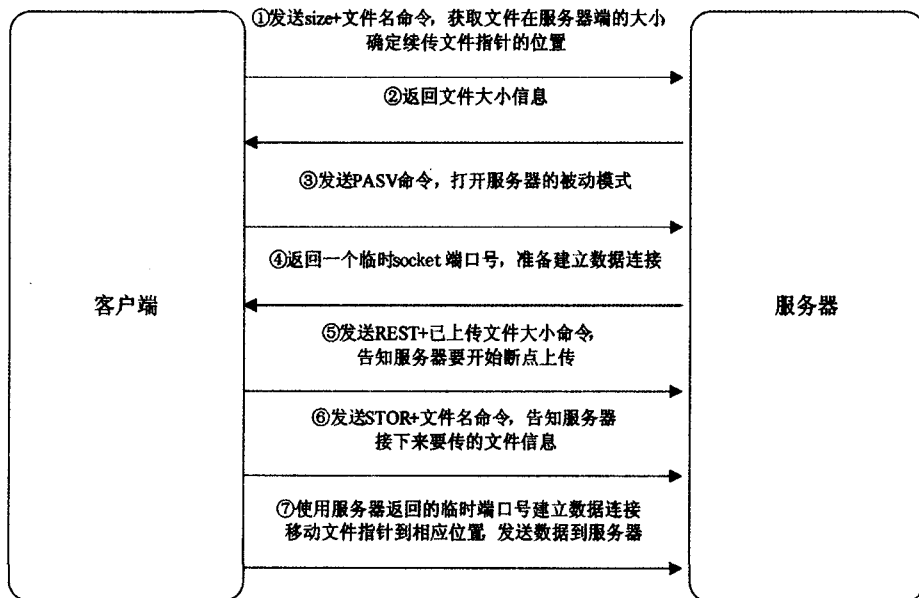


图 6 断点续传实现过程

断点续传方面,槽函数 `rawCommandReplyMessage()` 与 `QFtp::rawCommandReply` 信号相连接,用于接收服务器响应 `rawCommand()` 返回的信息并进而对此行判断,发送信号,触发相应的解析消息函数。建立函数 `handleSIZEMessage()` 解析服务器端的文件大小, `handlePASVMessage()` 用来解析被动模式下服务器发送的临时端口。服务器返回的临时端口消息格式如下:

Message=Entering Passive Mode (a,b,c,d,e,f)

其中,服务器 IP 地址:a.b.c.d,计算端口号公式:  
端口号 =  $e * 2^8 + f$

客户端发送完成命令,解析得到数据通讯端口之后,使用 `QTcpSocket` 类 `_datachannel` 与服务器建立 `Socket` 数据通道。准备发送续传文件。

设定发送缓存窗口大小为 `BUFSIZE`,每次发送窗口大小的文件段,发送成功后窗口移动一个 `BUFSIZE`,直到文件指针到底文件末尾,发送完毕,并向 `Socket` 服务器发送完成信息。设计的缓存窗口机制可以避免发送大文件时一次性读取文件可能发生的内存溢出问题,并且可靠的 `TCP` 连接保证了文件传输过程中的数据完整性。

## 2.2 Socket 服务器端实现

`Socket` 服务器主要负责接收客户端发送的数据,并根据数据调用相应的模块进行处理。主要包括:账户验证模块;视频信息处理模块;上传状态模块。

服务器程序在运行之前,读取存储在配置文档中的数据库信息以及 `FTP` 用户名密码,这种设计方式,使得即使数据库地址或者 `FTP` 账户信息由改变的时候,管理员只需要修改配置文档中的数据,无需对程序进行修改。

服务器类 `Server` 的主要功能包括 `Socket`<sup>[10]</sup> 侦听与数据库操作。

### 2.2.1 Socket 侦听

主要包含两个类: `SocketServer` 和 `SocketClient`。 `SocketServer` 派生自 `QTcpServer`,这是一个允许接收来访的 `TCP`<sup>[11]</sup> 连接。 `SocketClient` 重新实现了 `QTcpSocket`,处理一个单独的连接。在任何时候,在内存中 `SocketClient` 对象的数量和正在被服务的客户端数量都是一样多的。

`SocketServer` 类通过 `QTcpServer` 重新实现了 `incomingConnection()` 函数。只要有一个客户端试图连接到服务器证件听的端口,这个函数就会被调用。

```
void Server::incomingConnection(int socketId)
```

```
{
```

```
    ClientSocket * client = new ClientSocket(this);
```

```
    client->setDBPara ( database_host, database_SID, database_
port, database_username, database_password, _ftpusername, _ftppass-
```

```
word);
```

```
    client->setSocketDescriptor( socketId );
```

```
}
```

在 `incomingConnection()` 中,创建了一个 `ClientSocket` 对象作为 `SocketServer` 对象的子对象,并且将它的套接字描述符设置成提供给服务器的数字。当连接终止时, `ClientSocket` 对象将自动删除。函数 `setDBPara()` 为 `ClientSocket` 访问数据库的初始化工作。

```
class ClientSocket:public QTcpSocket
```

```
{
```

```
    Q_OBJECT
```

```
public:
```

```
    ClientSocket( QObject * parent = 0 );
```

```
private slots:
```

```
    void ReadWriteClient();
```

```
};
```

`ClientSocket` 类派生自 `QTcpSocket` 并且封装了一个单独客户端的状态。

```
ClientSocket::ClientSocket( QObject * parent)
```

```
{
```

```
    connect( this, SIGNAL( readyRead() ), this, SLOT( Read-
```

```
WriteClient() );
    connect ( this, SIGNAL ( disconnected ( ) ), this, SLOT
```

```
( deleteLater() );
```

```
}
```

构造函数中建立了必要的信号-槽连接, `disconned()` 信号被连接到 `deleteLater()`,确保当关闭套接字连接时, `ClientSocket` 对象会被删除。 `ReadWriteClient()` 函数与 `QTcpSocket` 的信号 `readyRead()` 相连接,实现了对 `Socket` 的读取与写入操作。

在 `SocketServer` 的 `main()` 函数中,使用 `QTcpServer::listen()`;来启动服务器,并设置想要接受的连接的 IP 地址和端口号。

### 2.2.2 数据库操作

`QtSql` 模块提供了与平台以及数据库无关的访问 `SQL` 数据库的接口。因北京市精品课程平台采用 `Oracle` 公司的数据库产品,故本案采用 `OCI`<sup>[12]</sup> 的方式对数据库进行连接操作。在进行数据库操作之前,需要进行初始化工作。

```
QSqlDatabase db = QSqlDatabase::addDatabase( "OCI" );
```

```
db.setHostName( database_host );
```

```
db.setDatabaseName( database_SID );
```

```
db.setUserName( database_username );
```

```
db.setPassword( database_password );
```

```
db.setPort( database_port.toInt() );
```

其中 `database_host`, `database_SID`, `database_username`, `database_password`, `database_port` 为从配置文件中

(下转第 209 页)

与计算的信标个数开始减少,致使正确率开始下降。

表2 接收机数量与正确率

接收机数量	1	2	3	4	5	6
正确率	40%	51%	64%	78%	61%	52%

最后,接收移动的速度也是影响定位精度的一个重要因素,当接收机移动速度过快时,位置计算正确率明显大幅下降,当接收机移动较为缓慢时,位置计算的正确率会较高。

## 5 结束语

基于超声波技术和射频技术嵌入式<sup>[11]</sup>的 Cricket 定位系统能够满足一般定位要求,同时由接收机计算自身的位置,保证了使用者的隐私。当然,系统仍然存在着一些需要完善的地方,文中将节能机制引入 Cricket 系统,在一定程度上降低了系统的能耗和成本;对信道的分配适当改进,有效地避免信号之间冲突,防止计算出错误的距离,并且保证节点之间的公平性<sup>[12]</sup>。文中所述的系统定位精度较为理想,节点利用率高、适用范围广。

### 参考文献:

- [1] 张小红. 动态精度单点定位 (PPP) 的精度分析[J]. 全球定位系统, 2006(1): 7-11.
- [2] 韩 强, 陈天滋. UML 在关系型 GIS 空间数据库设计中的

(上接第 205 页)

读取的数据库信息。

初始化工作完成之后,调用 QSqlDatabase::open()  
( ) 打开数据库连接,使用 QSqlQuery 执行 SQL 操作。

## 3 结束语

文中详细地说明了基于 Qt 框架构建的 FTP 上传系统的客户端与服务端的设计与实现,描述了如何在现有 Qt 框架的基础之上,使用 FTP 协议进行文件上传,并且,根据 RFC765 中对于 FTP 命令的规定,补充实现了 Qt 中 QFtp 模块的断点续传的功能。应用证明,设计的上传系统,稳定可靠,方便精品课程平台教师用户上传视频文件以及添加视频信息,同时也减轻了系统管理员的工作。

### 参考文献:

- [1] Blanchette J. C++ GUI Programming with Qt 4 [M]. Second Edition. 北京: 电子工业出版社, 2008: 270-286.
- [2] 晏 宁, 薛继东, 徐学渊. 基于 Linux 集群的 socket 迁移技术[J]. 微机发展, 2004, 14(6): 95-98.
- [3] 黄宇东, 胡跃明, 陈 安. 基于 Qt 的多线程技术应用与研

应用与研究[J]. 江苏大学学报(自然科学版), 2002, 23(1): 82-86.

- [3] Ververides C, Polyzos G C. Mobile Marketing Using A Location Based Service [EB/OL]. [2005-06-15]. <http://mm.aueb.gr>.
- [4] Savvides A, Han C C, Srivastava M B. Dynamic fine grained localization in ad-hoc networks of sensors [C]//Proc of MobiCOM. [s. l.]: [s. n.], 2001: 166-179.
- [5] Priyantha N, Chakraborty A, Balakrishnan H. The cricket location support system [C]//Proceedings of MobiCom. Boston, MA, USA: [s. n.], 2000: 32-43.
- [6] 吴永祥. 射频识别 (RFID) 技术研究现状及发展展望[J]. 微计算机信息, 2006(11Z): 234-236.
- [7] 孙 瑜, 范平志. 射频识别技术及其在室内定位中的应用[J]. 计算机应用, 2005, 25(5): 1205-1208.
- [8] Zhao Yiyang, Liu Yunhao, Ni L M. VIRE: Active RFID based localization using virtual reference elimination [C]//2007 International Conference on Parallel Processing (ICPP 2007). [s. l.]: IEEE Press, 2007: 56-56.
- [9] 孙立民, 李建中, 陈 渝, 等. 无线传感器网络 [M]. 北京: 清华大学出版社, 2005: 135-136.
- [10] 马玉秋, 龙承志, 沈树群. 长距离移动定位技术与室内定位技术[J]. 数据通信, 2004(5): 39-41.
- [11] 范平志, 邓 平, 刘 林. 蜂窝网无线定位 [M]. 北京: 电子工业出版社, 2002: 33-35.
- [12] Hightower J, Borriello G. Location systems for ubiquitous computing [J]. IEEE Computer, 2001(8): 57-66.

究[J]. 软件导刊, 2009, 8(10): 40-42.

- [4] 陈金阳, 蒋建中, 张良胜. FTP 协议分析及其客户端程序实现[J]. 计算机工程与应用, 2005(32): 130-132.
- [5] 吴 珂, 卢秉亮, 张 磊. 基于 FTP 协议客户端软件的实现[J]. 沈阳航空工业学院学报, 2006, 23(4): 39-42.
- [6] 詹金华, 刘 锋. 基于 FTP 协议的素材迁移服务器的设计与实现[J]. 计算机技术与发展, 2008, 18(3): 56-58.
- [7] Trolltech. Qt - Cross - Platform C++ Development - Troll - tech [EB/OL]. 2007. <http://www.trolltech.com/products/qt/features/index>.
- [8] 龚 斌, 季宏涛. FTP 协议分析及其客户端程序的实现[J]. 小型微型计算机系统, 1997(5): 26-29.
- [9] RFC765 Introduction [S/OL]. 1980-06. <http://www.faqs.org/rfcs/rfc765.html>.
- [10] 朱晓蕾, 余达太, 孙昌国, 等. 基于客户/服务器模式的远程控制 [J]. 计算机工程与设计, 2004(5): 796-799.
- [11] 王 集成. 基于 TCP/IP 协议, 用 Windows Socket API 实现 FTP 服务器的客户端 [J]. 微计算机应用, 2002(5): 305-307.
- [12] 李伍元, 黄 东. 基于 OCI 和 ODBC 异构数据库通用接口设计[J]. 计算机技术与发展, 2007, 17(2): 153-155.