重庆大学大数据与软件学院

上 机 实 验 报 告

|  |  |
| --- | --- |
| 上机实践项目 | TCP Socket Programming |
| 课程名称 | 计算机网络 |

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 刘京京 彭可涵 | 成绩 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 20213440 20220905 | 教师 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 班级 | 22软件4班  22软件3班 | 日期 |  |

**《计算机网络》上机实验报告**

**开课实验室：DS1502 2024 年 5 月 5 日**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | | 刘京京 彭可涵 | 年级、班级 | 22软件工程4班 22软件3班 | 学号 | **20213440 20220905** |
| 上机（项目）名称 | | | TCP Socket Programming | | 指导教师 | 胡海波 |
| 教师评语 | 教师签名：  **年 月 日** | | | | | |

1. **上机目的**

学习如何构建最简单的、基于C/S模式的通信程序

1. **基本原理**

套接字：在主机端，由应用进程创建、由操作系统控制的接口。通过套接字一个应用进程可以向另一个进程发送或接收报文。对应于传输层的服务，有两种类型的套接字：TCP：可靠的，面向字节流的传输，封装为TCP报文段(segment)；UDP：不可靠的传输，封装为UDP用户数据报(user datagram)

1. **使用的软件、硬件**

ATM开发：VS Code

服务器开发：

1. **上机操作步骤**

* 编写ATM界面
* 编写ATM发送、处理报文代码
* 编写服务器接收报文、处理报文信息、返回报文代码
* 测试本项目内ATM和服务器连接情况（包括登陆账号、取款、查询余额、断开连接四项）
* 测试项目间连接情况，调试代码

1. **项目介绍**

本项目分为客户端和服务器两个部分，刘京京负责开发客户端，彭可涵负责开发服务器。本项目代码共计1205行，其中客户端887行，服务器端318行。我们采用Socket编程、基于C/S架构、依据标准文档RFC-20222022构建了ATM和银行服务器的通信程序。本文将从客户端、服务器两方面介绍项目开发过程和连接调试过程中遇到的问题和解决方案。

* 客户端介绍

本项目客户端采用JAVA Swing开发，共五个界面，分别为用户登录界面、功能选择界面、取款界面、查询余额界面、用户退出界面，当用户取款完成后（不论取款成功或失败），自动弹出查询余额界面，分别在Login.java、Menu.java、Withdrawl.java、Balance.java、Bye.java文件中实现，文件执行入口为Main.java文件，在该文件中创建Login()类并结合收到的报文信息、用户点击的按钮来显示各个界面。具体界面展示如下：

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

针对报文的收发，封装在Link.java文件中。首先创建一个套接字 clienSocket，并指定连接的服务器的 IP 地址和端口号。创建一个输出流对象DataOutputStream out并利用它的writeBytes()函数进行报文的发送；创建一个输入流对象BufferedReader input并利用它的readLine()函数进行报文的读取。

* 服务器介绍

服务器端通过TCP Socket编程与客户端建立连接，并通过读取处理客户端发送的请求，根据不同的请求类型调用相应的处理方法进行业务逻辑处理。服务器需要与数据库进行交互，以获取和更新客户账户信息。

ATM服务器主要由以下部分组成：

1.TCP Server：该部分实现了服务器的Socket通信。端口号2525，监听一个指定的端口，并等待客户端连接。当客户端连接，就会创建一个新的线程来处理与该客户端的通信。这个线程会接收客户端请求，处理请求并返回响应结果。

2.数据库：该部分存储了所有的客户账户信息及其余额情况。服务器在处理客户请求时需要从数据库中获取数据，并在处理完请求后更新数据库。

3.业务逻辑处理：该部分实现了各种ATM业务逻辑，实现账户及密码验证、查询余额、取款等操作，具体功能如下：

（1）处理HELO请求：当客户端发送以“HELO”开头的请求时，服务器会调用handleHELORequest方法进行处理。该方法验证输入的账户是否正确，并向客户端返回相应的响应信息。

（2）处理PASS请求：当客户端发送以“PASS”开头的请求时，服务器会调用handlePASSRequest方法进行处理。该方法将验证客户端输入的密码是否正确，并向客户端返回相应的结果，例如成功登录或密码错误的提示。

（3）处理取款请求：当客户端发送以“WDRA”开头的请求时，服务器会调用handleWithdrawRequest方法进行处理。该方法将检查客户账户余额是否足够进行取款操作，并在取款成功或失败后返回相应的结果给客户端。

（4） 处理余额查询请求：当客户端发送“BALA”请求，服务器会调用getBalance方法进行处理。该方法将获取客户账户的当前余额，并将其发送回给客户端。

（5） 处理退出请求：当客户端发送“BYE”请求时，服务器会向客户端发送“BYE”响应，并断开与客户端的连接，结束通信。

* 项目遇到的困难：

1. 客户端开发初期，在每个界面都创建Link类，导致用户只能登录并不能进行取款等操作；解决方案：在Login.java文件中创建的Link类对象作为参数传递给他的下一界面的构造函数中并且每当创建一个新的界面的时候都应将该对象传递下去，这样保证连接不变。
2. 客户端开发初期无法查询余额；解决方案：经协商，得知报文发送格式中多了“<>”，更改发送报文和处理报文格式，成功解决问题。
3. 客户端测试连接时，针对python组的同学，连接成功但发送HELO报文没有任何响应；暂无解决方案。
4. 出现网络通信问题：在建立TCP Socket连接和进行数据传输的过程中，可能会出现网络通信问题，例如连接超时、断开连接等，导致客户端和服务器之间的通信中断，需要仔细检查网络配置和异常情况代码。
5. 数据库操作问题：服务器需要与数据库进行交互以获取账户信息并进行相应的验证和更新操作。在处理数据库查询和更新时，SQL语句编写错误。应当确保数据库操作的正确性和高效性。
6. 并发处理问题：ATM服务器需要处理多个客户端的并发请求，导致线程安全性问题和资源竞争。该问题暂未解决，无法实现多个客户端同时连接。
7. **结果及分析**

* 本实验在组内调试完成后，进行组间测试发现仍有很多问题，组间调试可以更好地完善代码。
* 本实验实现了双向通信，客户端能够正确接收到服务器的回复。在测试期间服务器需要一直运行，以保证及时给予客户端响应。
* 服务器的IP需要通过手动设置来固定，以防客户端连接不上。
* 采用标准文档规定报文的格式能大大减小通信时因报文格式不一样而更改代码消耗的工作量。

1. **上机实验总结**

* 了解了如何构建最简单的、基于C/S模式的通信程序，更好地理解了TCP传输服务
* 在Socket套接字编程中，通过创建输入流对象和输出流对象并利用其发送和接收报文函数可以进行通信
* 在进行组内调试和组间调试的过程中，一步步调试代码，提高了调试通信程序的能力。
* 网络编程注重细节，编写代码时绝对不能马虎。