

Socket Programming

UDP &FTP Report

软件52 许乾 2015080124

一 . 实验环境

CPU:Inter(R) Core(TM)i5-6300HQ CPU @2.30GHz 2.30GHz

Memory:16GB

Programming language:C

二 . UDP

UDP协议全称是用户数据报协议。用于处理数据包，是一种无连接的协议。

本次实验简单修改了UDPServer和UDPClient。使客户端向服务端发送0到50的数字，服务端接到信息后，向客户端返回序列号加上发送的数字。

UDP服务端首先创建一个Socket->然后接受客户端连接的请求->处理命令->关闭Socket

UDP客户端创建一个Socket->连接到服务端->发送package包->输入信息->发送到客户端->等待客户端处理命令->关闭Socket

(Optional)

1:如何用UDP实现两个客户端的聊天

可以利用上层协议来保证可靠运输，如果客户端使用UDP协议发出信息后，服务端收到该包，然后服务端需要使用UDP协议发回给客户端一个应答包。这样能够保证信息的传输。

2：是否可以利用UDP进行文件传输？如果可以，该如何实现？

UDP是一个不可靠的传输连续数据协议，它容易丢包而且接受包的次序为随机，为了解决丢包问题可以让接收端每次正确地接收到一个包数据后向发送端发出接受请求。为解决接受包次序的问题，可以在每次传输时定义一个序列号，可以让接收端按照正确的顺序接受这些包。但是为了解决这两种问题，会使文件传输变得缓慢，所以文件传输还是利用TCP为多。

三 . FTP

FTP 全名File Transfer Protocol，中文为文件传输协议。用户可以利用FTP通过客户机程序向远程主机上传文件或从远程主机下载文件。

FTP客户端:

创建一个Socket->连接至服务器->输入信息->等待响应->关闭Socket

FTP服务端:

创建一个Socket->监听->接收连接的请求->接受并处理命令->关闭Socket

FTP程序运行方式及说明：

服务端：利用gcc命令首先编译代码，然后可以在终端输入./server连接至默认端口号21，也可以利用./server -port n 指定端口号，此外还可以利用./server -root 命令指定文件根目录。

客户端：利用GCC命令首先编译代码，然后在终端输入./client连接至默认端口21，127.0.0.1,也可以利用形如

./client -port 127.0.0.1 : 6789这种方式连接到指定的端口。

利用QUIT或ABOR命令进行退出。

实现的FTP命令：

在客户端发送命令后，服务端会进行相应的响应，并发送回客户端一个相应数字+字符串。

本次实验实现的命令：USER,PASS,

TYPE,RETR,STOR,QUIT,PORT, PASV,MKD,CWD,LIST,RMD，这些指令均需要登录才可以使用，利用USER anonymous命令和PASS 命令登录

客户端均能发送这些命令，服务端都能正确进行响应。

服务端利用我自己写的客户端和autograde.py进行了测试，可以正常运行。

客户端利用了166.111.80.127 : 5050进行了测试。

四 . 总结

刚得到这次实验作业时，首先由于文档是全英文，一开始阅读时花费了一些时间，也有一些地方由于个人翻译不太准确，理解得有些错误。经过一段时间阅读后，对于UDP和TCP也有了一些基本的理解。TCP是一种面向连接的、可靠的、基于字节流的传输层通信协议。当客户和服务端彼此交换数据前，必须先在双方之间建立一个TCP连接，之后才能传输数据。而udp是用户数据报协议，是一个简单的面向数据报的运输层协议。UDP不提供可靠性，它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。

刚开始进行实验时，遇到了不少困难，如在客户端发送命令，服务端却没有响应，在经过一些更改后，可以在服务端和客户端进行正常通讯，可是在运行autograde.py时发现第一步就没有连接上，后来才知道因为没有遵守协议，从一开始所做得一些信息处理方式都是错误的，经过了很长时间的修改，才得以通过。如果一开始就认真按照协议要求编写代码的话就能避免花费很多不必要的时间。这次实验我学到了很多有用的知识，也给了我一次很好的教训。