

FTP 实验

文件传输是计算机网络的基本功能，文件传输协议（File Transfer Protocol，FTP）是一个基本的应用层协议。本实验需要学生在 Linux 系统上使用 Socket 编程技术实现简化的 FTP 服务器和客户端的程序。

1. 实验目的

本实验要求在 Linux 系统上完成一个文件传输协议（FTP）的简单实现。通过本实验，学生不仅可以深入理解 FTP 的原理和协议细节，还可以学会利用 Socket 接口设计实现简单应用层协议，掌握 TCP/IP 网络应用程序的基本设计方法和实现技巧。

2. 实验说明

FTP 是 File Transfer Protocol 的简称，即文件传输协议的缩写。该协议用于在两台计算机之间传送文件。FTP 会话包含了两个通道，一个是控制通道，一个是数据通道。控制通道是和 FTP 服务器进行沟通的通道，连接 FTP 服务器，发送 FTP 指令；数据通道则是和 FTP 服务器进行文件传输或者获取文件列表的通道。

FTP 中，控制连接的各种指令均由客户端主动发起，而数据连接有两种工作方式：主动方式（PORT 方式）和被动方式（PASV 方式）。主动方式下，FTP 客户端首先和 FTP 服务器的控制通道对应端口（一般为 21）建立连接，通过控制通道发送命令，客户端需要接收数据的时候在这个通道上发送 PORT 命令。PORT 命令包含了客户端用什么端口（一个大于 1024 的端口）接收数据。在传输数据的时候，FTP 服务器必须和客户端建立一个新的连接。被动方式下，建立控制通道的过程和主动方式类似，当客户端通过这个通道发送 PASV 命令的时候，FTP Server 打开一个位于 1024~5000 之间的随机端口并且通知客户端，然后客户端与服务器之间将通过这个端口进行数据的传送。

具体的 FTP 规范请参考 RFC959。

3. 实验内容

本实验要求学生在 Linux 系统上使用 C/C++ 编程语言利用 Socket 接口实现 FTP 客户端和服务器的程序，使客户端可以连接至服务器，并且可以进行一些 FTP 的基本操作，如列出目录、下载文件等。从 FTP 的实现角度来看，客户端与服务器的命令通道和数据通道需要分离，同时应该支持包括但不限于以下一些 FTP 命令：

- 1) get: 取远方的一个文件。
- 2) put: 传给远方一个文件。
- 3) pwd: 显示远方当前目录。
- 4) dir: 列出远方当前目录下的子目录和文件列表。
- 5) cd: 改变远方当前目录。
- 6) ?: 显示你提供的命令列表。
- 7) quit: 退出返回。

4. 思考问题

请在完成了 FTP 客户端与服务器程序之后，思考如下的问题：

- 1) 在 FTP 中，为什么要建立两个 TCP 连接来分别传送命令和数据？
- 2) 主动方式和被动方式的主要区别是什么？为何要设计这两种方式？
- 3) 当使用 FTP 下载大量小文件的时候，速度会很慢，这是什么缘故？可以怎样改进？

5. 实验要求和评分标准

实验考核包括编程检查和实验报告两部分：

(1) 编程检查（80%）

程序检查时间：2022 年 6 月 17 日。

代码提交时间：2022 年 6 月 19 日。

主要考核程序功能和代码实现，包括但不限于功能演示、代码测试、现场问答等，总分值 100 分，评分标准如下：

	功能要求	分值
基本功能	客户端和服务端之间能够建立 Socket 连接	5
	客户端和服务端之间能够进行数据通讯	5
	实现客户端从服务器下载文件功能	10
	实现客户端向服务器上传文件功能	10
	遵循 RFC959 规范	15
	实现远程文件目录的显示和操作功能	5
	实现命令查询、退出功能	5
扩展功能	实现主动和被动两种工作方式	10
	实现登录验证过程（有安全机制）	10
	服务器支持多客户端连接和数据传输	15
额外功能		10
总分值		100

(2) 实验报告（20%）

报告提交时间：2022 年 6 月 19 日。

报告内容应包括但不限于程序架构、实现方法、实验测试结果、遇到的问题 and 解决办法、思考题解答等，总分值 100 分。