

上海大学

SHANGHAI UNIVERSITY

毕业设计（论文）

UNDERGRADUATE PROJECT (THESIS)

题目：基于 socket 的文件传输的设计与实现

学院 计算机工程与科学

专业 计算机科学与技术

学号 10122060

学生姓名 吕伟彬

指导老师 雷咏梅

起讫日期 2014.02.17—2014.05.30

【目录】

基于 socket 的文件传输设计与实现

摘要

本文从先讨论了 socket 网络编程的方法,然后讨论了文件传输协议的设计与实现,主要包括 HTTP、FTP 和 BitTorrent 协议。涉及到以上几种文件传输协议的协议规范,以及开源实现的实现细节,包括 libcurl (HTTP 和 FTP) 以及 libtransmission (BitTorrent)。同时描述了我本人实现的文件传输管理器 wdl 以及一个简单的 HTTP 服务器 wdl's 的设计与实现。wdl 支持 FTP、HTTP 和 BitTorrent 协议,多任务多线程下载,支持断点续传;可以从互联网上下载文件。wdl's 是一个非常简单的 HTTP 服务器,实现 HTTP200, HTTP404 和 HTTP206,支持断点续传,可以与 wdl 很好地配合。wdl 和 wdl's 都是在 Linux 完全采用 C 语言实现,wdl's 不依赖任何第三方库。

关键词: Socket,linux,文件传输协议, HTTP 服务器

ABSTRACT

This Document describes the design and implement of file transfer protocol using Socket, including HTTP, FTP and BitTorrent Protocol. The specifications and open source implements of those protocols are involved. Libcurl is introduced as HTTP and FTP implement, while libtransmission as BitTorrent implement. Also, my file transfer manager named wdl and simple HTTP server named wdl's are described in the document. Wdl supports FTP, HTTP and BitTorrent Protocol, multi-task, multi-thread and breakpoint-resume. It's OK to download files from the Internet using wdl. And wdl's is a very simple HTTP server, which only implements HTTP200, HTTP206 and HTTP404, and breakpoint-resume too. Wdl and wdl's work well together. Wdl and wdl's are both written in C language in Linux System. In addition, wdl's does not depend on any third-party library.

Keywords:Socket,linux,file transfer protocol,HTTP server

绪论

Socket 最早作为 BSD UNIX 的进程通信机制, 随着网络的出现, 它也开始支持网络通信; 事实上, 在 UNIX 系统下, 网络通信被认为是一种特殊的进程间通信。网络的快速发展, 使得 Socket 网络编程越来越流行, 以至于现在很多人都忽略了 Socket 作为本机进程间通信的功能。本文中所谈到的 Socket, 除非特别说明, 都是只指网络套接字。而且 Windows 最早的 IP/TCP 实现就来源于 BSD UNIX 的代码, 所以 Windows 下的网络编程接口也是 Socket 兼容的。

Socket 实际上就是操作系统内核 (主要是 IP/TCP 协议栈) 对外提供的网络编程接口, 它属于系统调用层面, 而不是特定语言的标准库。因为大部分操作系统都是 C 语言实现的, 因此, Socket 接口一般都是以 C 函数的形式提供。而像 C++, Java, Python 这些语言的网络支持, 要么是对 C 语言的封装, 如 C++; 要么就是通过虚拟机 (如 Java) 或者解释器 (如 Python) 间接调用 Socket 接口; 总之, 它们最底层还是要调用操作系统提供的 Socket 接口。因此, 了解熟悉 Socket 接口对理解网络编程, 无论哪个层面的, 都非常有益。

既然 Socket 是作为操作系统网络协议栈对外的接口, 那么了解相应的网络协议 (主要是 TCP/IP) 就很有必要了。本文的前一部分会简要描述 TCP/IP 协议的规范, 但不会很深入, 对这方面感兴趣的读者可以参考 IP/TCP 详解^[1]。接着, 我们讨论 Socket 网络编程接口, 从系统调用的层面展示网络编程的方法, 这要求读者对 UNIX 系统下的 C 语言编程有一定基础, 但这些并不在本文的讨论范围内; 对这方面感兴趣的读者可以参考 UNIX 环境高级编程^[2]。

Socket 套接字其实不仅仅局限于网络编程, 它有多种协议类型, 最典型的就 AF_INET 和 AF_INET6, 分别对应的是 IPv4 和 IPv6 套接字。AF_UNIX 或者 AF_LOCAL 是 UNIX 域套接字, 用于本地进程间通信。还有其他协议特定的类型, 比如 AF_IPX 是 IPX 协议套接字, AF_X25 是 ITU-T X.25 / ISO-8208 协议套接字, 以及内核的 netlink 套接字 AF_NETLINK。本文只讨论 AF_INET, IPv6 套接字与 IPv4 类似, 主要是地址结构不同。

而 AF_INET 协议下又可以指定 SOCK_STREAM 和 SOCK_DGRAM, 分别对应 TCP 和 UDP 协议; 甚至可以指定 SOCK_RAW 使用原始套接字, 原始套接字可以接受到 IP 的数据报文, 由程序员手动构造和解析 TCP 或者 UDP 协议首部字段。本文只讨论 SOCK_STREAM。也就是说只讨论 TCP 套接字, 同时也只描述 TCP 和 IP 协议规范。

接着我们讨论基于 Socket 的网络文件传输协议, FTP、HTTP 和 BitTorrent 协议。FTP 协议是流行的文件传输协议, 主要用在托管大量文件的服务器上; HTTP 最初为了互联网浏览网页而设计, 虽然协议中有很多与网页服务相关的字段, 但本质就是文件传输的协议; BitTorrent 近年来也越来越流行, 主要用在文件的传输上。

最后结合上述的内容来描述我所完成的文件下载器 wdl 和一个简单的 HTTP 服务器 wdl_s, 主要从网络编程的角度描述, 而不是从软件工程的角度。wdl 和 wdl_s 的源代码托管在 GitHub 上, 地址分别是 <https://github.com/wiiky/wdl> 和 https://github.com/wiiky/wdl_s。在 Fedora 20 下完成, 依赖 libgtk+-3.0、libcurl 和 libtransmission。

第一章 IP/TCP

第一节

参考文献

[1] W.Richard Stevens. TCP/IP Illustrated Volume 1: The Protocol. Addison Wesley, 1994.

[2] W.Richard Stevens, Stephen A Rago. 尤晋元, 张亚英, 戚正伟 译。 Advanced Programming in the UNIX Environment. UNIX 环境高级编程。人民邮电出版社, 2006 年 5 月 1 日。