

实验七 TCP/IP 协议的应用

【实验目的】

了解和掌握基于 socket 的网络编程技术，开发基于 socket 的网络应用。

【实验内容】

了解应用编程接口 API，掌握基于 socket 的网络编程的原理，开发利用 socket 的 TCP 文件传输应用程序。该应用需要具备的功能以及实现的要点描述如下：

1. 该程序应该包括服务器应用程序以及客户应用程序。
2. 用户需要身份验证。即对于客户端来说，无论是上传文件，还是下载文件，首先需要做的事情是登陆服务器，得到服务器的验证。若验证成功，即可与服务器之间开始传输文件。若验证失败，服务器则返回错误信息。错误信息包括，用户名错误，密码错误等。
3. 理解文件传输的原理。所谓文件传输，对于发送端来说，实质是将数据读入发送缓存再将其发送。对于接收端来说，实质是从接收缓存里读取数据并将其写入到指定的位置。
4. 请注意理解文件传输的含义。文件，指的是一切可以传输的信息，包括文本文件、图片文件、视频文件等。传输，指的是服务器和客户端都可以成为文件的发送者。从客户端的角度来说，客户端向服务器传送文件称为上传，服务器向客户端传送文件称为下载。

【实验步骤】

1. 阅读老师提供的示例程序，了解和掌握 socket 网络编程的原理
2. 根据实验要求，开发基于 TCP 的文件传输服务器端程序
3. 开发基于 TCP 的文件传输客户端程序
4. 编译程序、调试程序、运行程序
5. 写实验报告，>800 字，提交设计程序

【实验准备知识】

大多数操作系统使用系统调用(system call)的机制在应用程序和操作系统之间传递控制权。对程序员来说，每一个系统调用和一般程序设计中的函数调用非常相似，只是系统调用是将控制权传递给了操作系统。

当某个应用进程启动系统调用时，控制权就从应用进程传递给了系统调用接口。此接口再将控制权传递给计算机的操作系统。操作系统将此调用转给某个内部过程，并执行所请求的操作。内部过程一旦执行完毕，控制权就又通过系统调用接口返回给应用进程。系统调用接口实际上就是应用进程的控制权和操作系统的控制权进行转换的一个接口，即应用编程接口 API。

当应用进程需要使用网络进行通信时就发出 socket 系统调用，请求操作系统为其创建“套接字”，以便把网络通信所需要的系统资源（存储器空间、CPU 时间、网络带宽等）分配给该应用进程。操作系统为这些资源的总和用一个叫做套接字描述符的号码来表示，并把此号码返回给应用进程。应用进程所进行的网络操作都必须使用这个号码。通信完毕后，应用进程通过一个关闭套接字的系统调用通知操作系统回收与该“号码”相关的所有资源。

当套接字被创建后，它的端口号和 IP 地址都是空的，因此应用进程要调用 bind

(绑定)来指明套接字的本地地址。在服务器端调用 `bind` 时就是把熟知端口号和本地 IP 地址填写到已创建的套接字中。这就叫做把本地地址绑定到套接字。服务器在调用 `bind` 后,还必须调用 `listen` (收听)把套接字设置为被动方式,以便随时接受客户的服务请求。UDP 服务器由于只提供无连接服务,不使用 `listen` 系统调用。服务器紧接着就调用 `accept` (接受),以便把远地客户进程发来的连接请求提取出来。系统调用 `accept` 的一个变量就是要指明从哪一个套接字发起的连接。调用 `accept` 要完成的动作较多。这是因为一个服务器必须能够同时处理多个连接。同时处理多个连接的服务器称为并发方式工作的服务器。客户端在调用 `socket` 创建了套接字后,客户进程就调用 `connect`,来和远地服务器建立连接。在 `connect` 系统调用中,客户必须指明远地端点(即远地服务器的 IP 地址和端口号)。客户和服务器都在 TCP 连接上使用 `send` 系统调用来传送数据,使用 `recv` 系统调用接收数据。调用 `send` 需要的变量:套接字描述符、要发送的数据的地址、数据的长度。调用 `recv` 需要的变量:套接字描述符、缓存地址、缓存空间的长度。一旦客户或者服务器结束使用套接字,就把套接字撤销。这时就调用 `close` 释放连接和撤销套接字。UDP 服务器由于只提供无连接服务,因此,不使用 `listen` 和 `accept` 系统调用。系统调用的使用顺序如下图所示。

