

五级项目 8：基于 TCP 的基本网络通信

1 学时

- 1 学时

2 实验目的

- 理解、掌握、应用 Linux 下基于 TCP 协议进行基本的网络通信过程。

3 实验内容

- 本项目编写一个基于 TCP 协议的网络通信，要求客户端通过命令行参数向指定服务器发起连接，服务器向请求连接的客户端发送欢迎信息，客户端接收欢迎信息并显示。

4 实验原理

(1) 面向连接的通信

网络中两台主机之间的数据传输方式主要有两种：面向连接和面向无连接。在TCP/IP协议簇中，传输层的TCP协议和UDP协议分别实现的是面向连接和面向无连接的数据传输。

所谓面向连接，是指通信双方在进行通信之前，要事先在双方之间建立起一个完整的可以彼此沟通的通道。这个通道也就是连接，在通信过程中，整个连接的情况一直可以被实时地监控和管理。比如打电话，必须等线路接通了，对方拿起话筒才能相互通话，最后还要释放连接——挂电话。

面向连接服务具有连接建立，数据传输和连接释放这三个阶段，在网络层中又称为虚电路服务。

面向连接数据传输过程中数据按序传送，可靠性高，适合于在一定期间内向同一目的地发送许多报情况。

TCP（Transmission Control Protocol，传输控制协议）是基于连接的协议，也就是说，在正式收发数据前，必须和对方建立可靠的连接。一个TCP连接必须要经过三次“对话”才能建立起来。

建立连接的过程一般需要三次握手，如图 1 所示。释放连接的过程一般需要四次握手，如图 2 所示。

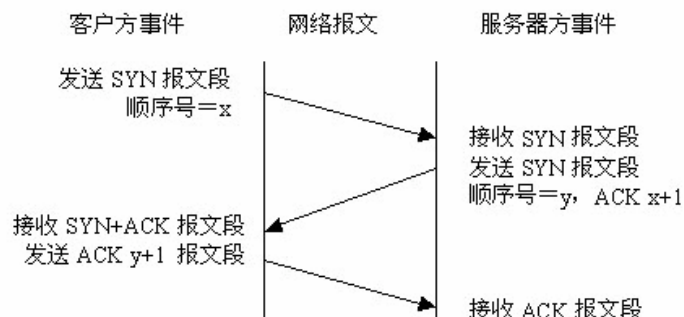


图 1 TCP 建立连接过程

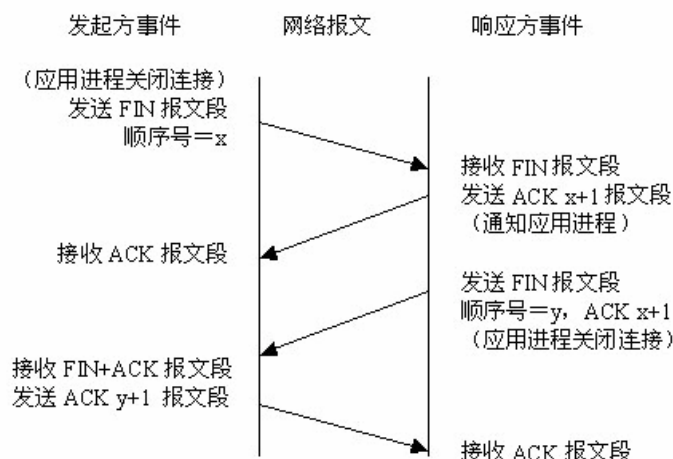


图 2 TCP 释放连接过程

UNIX/Linux 都是计算机使用的主流操作系统，TCP/IP 是广为应用的互连网协议，UNIX/Linux 为 TCP/IP 网络编程提供了一种网络进程通信机制：套接字接口（Socket）。

（2）套接字简介

套接字（Socket）是网络通信的基本操作单元，它提供了不同主机间进程双向通信的端点，这些进程在通信前各自建立一个 Socket，并通过对 Socket 的读/写操作实现网络通信功能，因此套接字也是进程间通信的一种方式，其通信过程如图 3 所示。

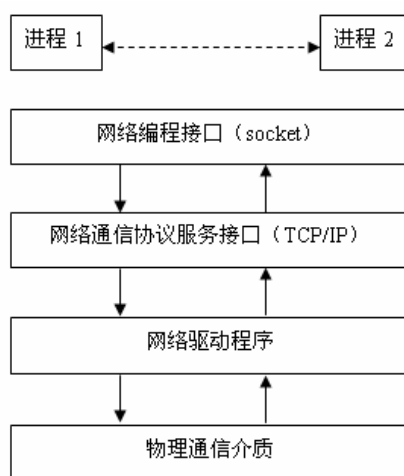


图 3 Socket 实现网络进程间通信

套接字主要分为以下 3 种类型。

● 字节流套接字

这是最常用的套接字类型，TCP/IP 协议簇中的 TCP（Transport Control Protocol）协议使用此类接口，它提供面向连接的（建立虚电路）、无差错的、发送先后顺序一致的、包长度不限和非重复的网络信包传输。

● 数据报套接字

TCP/IP 协议簇中的 UDP（User Datagram Protocol）协议使用此类接口，它是无连接的服务，以独立的信包进行网络传输，传输不保证顺序性、可靠性和无重复性，它通常用于单个报文传输或可靠性不重要的场合。

● 原始套接字

提供对网络下层通讯协议（如IP协议）的直接访问，它一般不是提供给普通用户的，主要用于开发新的协议或用于提取协议较隐蔽的功能。

（3）面向连接套接字通信过程

基于TCP协议进行通信的方式是面向连接的通信过程，通信进程双方之间在通信之前必须要建立连接从而通过此连接进行数据通信，通信结束后释放连接，是可靠的传输。但是建立连接和释放连接需要一定过程，因此速度有所降低。流式套接字就是使用TCP协议进行网络进程间通信的套接字方式。

使用流式套接字建立连接及通信的步骤如图 4 所示。

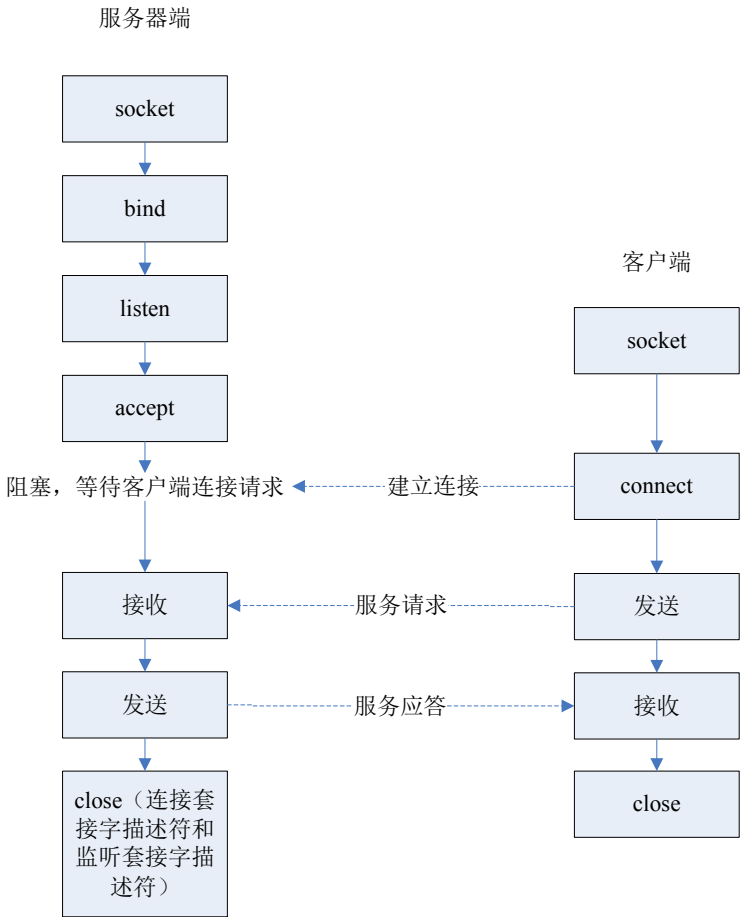


图 4 流式套接字通信过程

服务器端先启动，进行socket、bind、listen、accept一系列准备工作，然后如果没有客户端请求建立连接，则服务器端处于阻塞状态，等待客户端请求建立连接。

客户端启动后，首先向服务器端发起建立连接请求，得到服务器同意后，连接建立成功，客户端和服务端开始进行数据通信（即数据的发送和接收过程），客户机完成通信过程后释放连接，关闭套接字。

服务器关闭与某客户的连接，并根据情况决定是否结束整个服务器的服务，一般大型服务器是不间断服务的，除非进行系统维护和升级，所以一般服务器将会使用循环方式，不断接受客户请求以及与客户进行通信。另外服务器可以选择一次只处理一个客户请求（重复性服务器），也可以采用多进程或多线程等方式同时处理多个客户请求（并发服务器）。

5 预习要求和技术准备工作

- 掌握 Linux 基本操作

- 掌握 C 语言开发工具的使用
- 掌握基于 TCP 的套接字通信流程

6 实验环境

- PC 机
- 在 Windows 环境中的 VMware 虚拟机上运行 Ubuntu 操作系统或者独立的 Ubuntu 操作系统
- 基于 Linux 的 vi 编辑器和 gcc 编译器

7 实验设计及操作步骤

7.1 以 root 身份登录系统，在/home 目录中创建目录 exp58

```
cd /home
```

```
mkdir exp58
```

7.2 进入刚创建的目录

```
cd exp58
```

7.3 使用 vi 编辑文件，文件名分别是 nettcpserver.c 和 nettcpclient.c。

7.4 编写程序，实现要求的功能

7.5 编译可执行文件。

```
gcc -o nettcpserver nettcpserver.c
```

```
gcc -o nettcpclient nettcpclient.c
```

7.6 运行程序

在一个终端中运行服务器端nettcpserver:

```
./nettcpserver
```

在另一个终端中运行客户端nettcpclient:

```
./nettcpclient
```

服务器端运行结果:

```
received a connection from 127.0.0.1
```

客户端运行结果:

```
Received: Hello, you are connected!
```

8 实验报告提交要求:

将实验操作每个步骤中的命令、源程序以及截图写入实验报告，实验报告命名为“学号姓名-实验 58.doc”，交给指定人员。

9 项目思考

请在本例基础上增加，服务器处理多个客户端的功能，可以使用多进程或多线程。