

网络编程套接字

网络编程

预备知识

H₃ 源IP地址和目的IP地址

在IP数据包头部中，有两个IP地址，分别叫做(☆ 源IP地址)和(☆ 目的IP地址) --> 始终不变的

- 与之对应的还有(☆ 源MAC地址)和(☆ 目的MAC地址) --> 是会不断变化的

H₃ 认识端口号

(☆ 端口号(port))是传输层协议的内容

- 端口号是一个2字节16位的数据
- 端口号用来标识一个进程，告诉操作系统，当前的这个数据要交给哪一个进程来处理
- IP地址+端口号能够标识网络上的某一台主机的某一个进程 --> 称之为socket通信
- 一个端口号只能被一个进程占用

socket通信，本质上是进程间通信，跨网络的进程间通讯

任何的网路服务与网络客服端，如果要进行正常的数据通讯，必须要用端口号，来唯一标识自身。

H₃ 端口号与进程ID

一个进程可以绑定多个端口号，一个端口号只能绑定一个进程

一台机器上，可能存在大量的进程，但不是所有的进程都要对外进行网络数据请求

H₃ 源端口号与目的端口号

传输层协议(TCP和UDP)的数据段中有两个端口号，分别叫做源端口号和目的端口号，就是在描述“数据是谁发的，要发给谁？”

H₃ TCP协议

☆ TCP，Transmission Control Protocol传输控制协议

- 传输层协议
- 有连接
- 可靠传输
- 传输速度慢
- 面向字节流

H₃ UDP协议

☆ UDP，User Datagram Protocol用户数据报协议

- 传输层协议
- 无连接
- 不可靠传输
- 传输速度快
- 面向数据报

H₃ 网络字节序

- 发送主机通常将发送缓冲区中的数据按内存地址从低到高的顺序发出
- 接收主机把从网络上接到的字节依次保存在接收缓冲区中，也是按内存地址从低到高的顺序保存
- 因此，网络数据流的地址应这样规定：**先发出的数据是低地址，后发出的数据是高地址**
- TCP/IP协议规定，**网络数据流应采用大端字节序，即低地址高字节**

为了网络程序具有可移植性，可以调用以下的库函数进行网络字节序和主机字节序的转换

```
1  #include <arpa/inet.h>
2
3  uint32_t htonl(uint32_t hostlong);
4  uint16_t htons(uint16_t hostshort);
5  uint32_t ntohl(uint32_t netlong);
6  uint16_t ntohs(uint16_t netshort);
```

- h表示host，n表示network，表示32位长整数，s表示16位短整数
- htonl表示将32位的长整数从主机字节序转换为网络字节序
- 如果主机是小端字节序，这些函数将参数做相应的大小端转换后返回
- 如果主机是大端字节序，这些函数不做转换，将参数原封不动返回

H₃ socket编程接口