

Linux 下 socket 编程基本知识

1 概念

1.1 套接字 (socket)

Socket (套接字) 是一种通讯机制, 它包含一整套的调用接口和数据结构的定义, 它给应用进程提供了使用如 TCP/UDP 等网络协议进行网络通讯的手段。

Linux 中的网络编程通过 socket 接口实现, socket 既是一种特殊的 IO, 提供对应的文件描述符。一个完整的 socket 都有一个相关描述, 简称五元组 (协议、本地地址、本地端口、远程地址、远程端口), 每一个 socket 有一个本地的唯一的 socket, 由操作系统分配。

1.2 字节序

字节序分为大端字节序和小端字节序, 网络协议使用网络字节序即大端字节序。



内部字节表示顺序和网络字节顺序不一样的机器, 要对数据进行转换。

1.2.1 字节序转换常用函数

<code>uint32_t htonl(uint32_t hostlong);</code>	将一个 32 位整数由主机字节序转换为网络字节序
<code>uint16_t htons(uint16_t hostshort);</code>	将一个 16 位整数由主机字节序转换成网络字节序

uint32_t htonl(uint32_t netlong); 将一个 32 位整数由网络字节序转成主机字节序
uint16_t ntohs(uint16_t netshort); 将一个 16 位整数由网络字节序转换成主机字节序

1.3 数据结构

1.3.1 通用地址结构（一般不使用，但是内核中是使用的，所以需要强制转换为此种数据结构）

```
#include <sys/socket.h>
```

```
struct sockaddr{  
  
    unsigned short sa_family;  
  
    char sa_data[14];  
  
};
```

sa_data: 包含了一些远程电脑的地址、端口和套接字的数目，它里面的数据是杂糅在一起的。

sa_family: 一般来说 IPv4 使用 AF_INET。

在传递给需要地址结构的函数时，把指向该结构的指针转换成 struct sockaddr *传递进去。

1.3.2 因特网地址结构

```
struct in_addr{  
  
    in_addr_t s_addr    //IPv4 地址  
  
};
```

```
struct sockaddr_in{  
  
    short int sin_family;              //Internet 地址族如 AF_INET（主机字节序）  
  
    unsigned short int sin_port;      //端口号，16bit（网络字节序）  
  
    struct in_addr sin_addr;          //Internet 地址，32bit IPv4 地址（网络字节序）
```

```
    unsigned char sin_zero[8];    //添 0（为了格式对其的填充位）  
};
```

通用地址结构和因特网地址结构数据类型是等效的，可以相互转换，通常使用 `sockaddr_in` 更为方便。

2 基本函数

2.1 创建 socket

```
#include <sys/socket.h>
```

```
int socket(int domain, int type, int protocol)
```

返回：成功返回文件描述符，出错返回-1。

socket 创建在内核中，若创建成功返回内核文件描述表中的 socket 描述符。

参数：

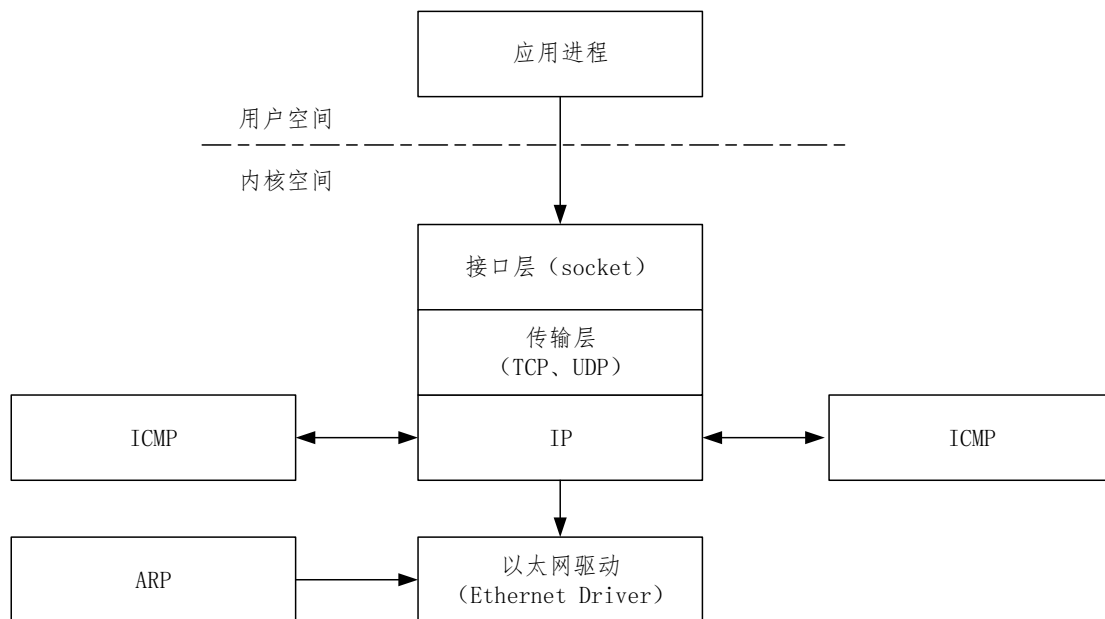
domain: **AF_INET** IPv4 因特网域、**AF_INET6** IPv6 因特网域、**AF_UNIX** unix 域、**AF_UNSPEC** 未指定。

protocol: 通常为 0，表示按给定的域和套接字类型选择默认协议

type: **SOCK_STREAM** 流式套接字，可以提供可靠的、面向连接的通讯流，它使用 TCP 协议，TCP 保证了数据传输的正确性和顺序性。

SOCK_DGRAM 数据报套接字，定义了一种无连接的服务，数据通过相互独立的报文进行传输，是无序的，并且不保证可靠、无差错，使用数据报协议 UDP 协议。**SOCK_RAW** 原始套接字，允许对低层协议如 IP 或 ICMP 直接访问，主要用于新的网络协议实现的测试等。

SOCK_SEQPACKET 长度固定的、有序、可靠的面向连接报文传递。



2.2 IPv4 地址族和字符地址间的转换函数

```
#include <arp/inet.h>
```

功能：将网络字节序转换位点分十进制

```
const char* inet_ntop(int domain, const void *restrict_addr, char *restrict_str,
socklen_t size);
```

功能：将点分十进制转换为网络字节序

```
const char* inet_pton(int domain, const void *restrict_str, char *restrict_addr);
```

参数：

domain: Internet 地址族，如 AF_INET

addr: Internet 地址，32 位 IPv4 地址（网络字节序）

str: 地址字符串（点分十进制）指针

size: 地址字符串大小

2.3 填写 IPv4 地址族结构案例

```
struct sockaddr_in sin; //定义一个 sockaddr_in 结构体
```

```
char buf[16];
```

```
memset(&sin, 0, sizeof(sin));
```

```
sin.sin_family = AF_INET;    //填写 Internet 地址族

sin.sin_port = htons((short)3001);    //填写端口号（网络字节序）

//填写 sin_addr

if (inet_pton(AF_INET, "192.168.2.1", &sin.sin_addr.s_addr) <=0)

{

    //错误处理

}

printf("%s\n", inet_ntop(AF_INET, &sin.sin_addr.s_addr, buf, sizeof(buf)));
```