



Lab 1 Ping程序的实现

总体流程

获取命令行 → 检验参数正确性 → socket连接 → 构造ICMP报文 → 等待接收报文 → 输出结果

ICMP报文



- 类型：1-127的差错报文，或128以上的信息报文。
- 代码：标志对应ICMP报文的代码。
- 校验和： 检验传输过程中是否出现差错。
- 标志： 仅适用于回显请求和应答ICMP报文，对于目标不可达/超时ICMP报文，该字段值为0。

Socket编程函数集

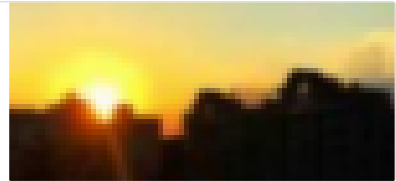
sockaddr_in, sockaddr, in_addr区别Socket编程函数集（非常有用）_maopig的专栏-CSDN博客
struct sockaddr { unsigned short sa_family; //2 char sa_data[14]; //14 }; 上面是通用的socket地址，具体到Internet socket，用下面的结构，二者可以进行类型转换 struct sockaddr_in { short int sin_family; //2 unsigned short int sin_port; //2 struct in_addr sin_addr; //4 unsigned char sin_zero[8]; //8 }; struct in_addr就是32位IP地址
<https://blog.csdn.net/maopig/article/details/17193021>

- struct sockaddr_in

```
struct    sockaddr_in
{
    short int          sin_family; /* 地址族,形如AF_xxx,通常用的是AF_INET, 2字节 */
    unsigned short int sin_port;   /* 端口号 (使用网络字节顺序) 2字节 */
    struct in_addr     sin_addr;   /* 存储IP地址, 4字节, 就是32位的ip地址 */
    unsigned char      sin_zero[8]; /* 总共8个字节, 实际上没有什么用, 只是为了和struct
                                     sockaddr保持一样的长度 */
};
```


- `int socket (int domain, int type, int protocol);`
domain: 取参数/ type: 套接字类型 / protocol: 协议类型

`int socket(int domain, int type, int protocol)`参数详解_聆听风雨的博客-CSDN博客
raw_socket(原始套接字)以及普通socket使用终极总结
<https://blog.csdn.net/luchengtao11/article/details/76635669> 网上对于socket的资料很多, 但是很凌乱。笔者这里整理一下socket的各种参数。可参考本人博客: Linux下的raw Socket (原
 <https://blog.csdn.net/luchengtao11/article/details/74689660>



```
if ( (sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)) < 0 )
//参数: AF_INET: TCP/IP协议家族; SOCK_RAW: 原始套接字;
//IPPROTO_ICMP: 传输层协议
{
    perror("socket_icmp");
    exit(0);
}
```

- `inet_pton()`

`inet_pton()`和`inet_ntop()`函数详解_夕小瑶的卖萌屋-CSDN博客
`int inet_aton(const char *cp, struct in_addr *inp);` `inet_aton()` 转换网络主机地址ip(如192.168.1.10)为二进制数值, 并存储在`struct in_addr`结构中, 即第二个参数*inp, 函数返回非0表示cp主机地址有效, 返回0表示主机地址无效。(这个转换完后不能用于网络传输, 还需要调用`htons`或`htonl`函数才能将主
 <https://blog.csdn.net/zyy617532750/article/details/58595700>

原创

函数中p和n分别代表表达 (presentation)和数值 (numeric)。地址的格式通常是ASCII字符串, 数值格式则是存放套接字地址结构的二进制值。

```
#include <arpa/inet.h>
int inet_pton(int family, const char *strptr, void *addrptr);    //将点分十进制的ip地址转化为用于网络传输的数值格式
    返回值: 若成功则为1, 若输入不是有效的表达式则为0, 若出错则为-1

const char * inet_ntop(int family, const void *addrptr, char *strptr, size_t len);    //将数值格式转化为点分十进制的ip地址格式
    返回值: 若成功则为指向结构的指针, 若出错则为NULL
```

- `sigaction` //信号发生
- `Struct icmp` 表示ICMP头部的数据结构
- `sendto()`函数

原型: `int sendto(int s, const void * msg, int len, unsigned int flags, const struct sockaddr * to, int tolen);`

: `sendto()` 用来将数据由指定的socket 传给对方主机. 参数s 为已建好连线的socket, 如果利用UDP协议则不需经过连线操作. 参数msg 指向欲连线的的数据内容, 参数flags 一般设0, 详细描述请参考`send()`. 参数to 用来指定欲传送的网络地址, 结构`sockaddr` 请参考`bind()`. 参数tolen 为`sockaddr` 的结果长度.