华清创客学院
makeru.com.cn

Linux网络开发课程

易老师

创客引领未来

扫微信二维码 获取更多信息

stephen

昵称：易胖

华清创客学院，金牌讲师

maker

LINUX网络开发-part3

1 广播和组播

2 UNIX域套接字

3 网络总结

LINUX网络开发-part3

华清创客学院
makeru.com.cn

1 广播和组播

2 UNIX域套接字

3 网络总结

广播

- 前面介绍的数据包发送方式只有一个接受方，称为单播
- 如果同时发给局域网中的所有主机，称为广播
- 只有用户数据报(使用UDP协议)套接字才能广播
- 广播地址
  - 以192.168.1.0（255.255.255.0）网段为例，最大的主机地址192.168.1.255代表该网段的广播地址
  - 发到该地址的数据包被所有的主机接收
  - 255.255.255.255在所有网段中都代表广播地址

广播发送

- 创建用户数据报套接字
- 缺省创建的套接字不允许广播数据包，需要设置属性
  - setsockopt可以设置套接字属性
- 接收方地址指定为广播地址
- 指定端口信息
- 发送数据包

setsockopt

- int setsockopt(int s, int level, int optname, const void *optval, socklen_t optlen);
  - 头文件：<sys/socket.h>
  - level：选项级别（例如SOL_SOCKET）
  - optname：选项名（例如SO_BROADCAST）
  - optval：存放选项值的缓冲区的地址
  - optlen：缓冲区长度
  - 返回值：成功返回0　失败返回-1并设置errno

广播发送示例


```
sockfd = socket(,,);
......
int on = 1;
setsockopt(sockfd, SOL_SOCKET, SO_BROADCAST, &on, sizeof(on));
.....
sendto(;;;;;);
```

广播接收

- 创建用户数据报套接字
- 绑定本机IP地址和端口
  - 绑定的端口必须和发送方指定的端口相同
- 等待接收数据

组播

- 单播方式只能发给一个接收方。
- 广播方式发给所有的主机。过多的广播会大量占用网络带宽，造成广播风暴，影响正常的通信。
- 组播(又称为多播)是一种折中的方式。只有加入某个多播组的主机才能收到数据。
- 多播方式既可以发给多个主机，又能避免象广播那样带来过多的负载(每台主机要到传输层才能判断广播包是否要处理)

网络地址

- A类地址
  - 第1字节为网络地址，其他3个字节为主机地址。第1字节的最高位固定为0
  - 1.0.0.1 – 126.255.255.255
- B类地址
  - 第1字节和第2字节是网络地址，其他2个字节是主机地址。第1字节的前两位固定为10
  - 128.0.0.1 – 191.255.255.255
- C类地址
  - 前3个字节是网络地址，最后1个字节是主机地址。第1字节的前3位固定为110
  - 192.0.0.1 – 223.255.255.255
- D类地址（组播地址）
  - 不分网络地址和主机地址，第1字节的前4位固定为1110
  - 224.0.0.1 – 239.255.255.255

组播发送

- 创建用户数据报套接字
- 接收方地址指定为组播地址
- 指定端口信息
- 发送数据包

组播接收

- 创建用户数据报套接字
- 加入多播组
- 绑定本机IP地址和端口
  - 绑定的端口必须和发送方指定的端口相同
- 等待接收数据

加入多播组

```
struct ip_mreq
{
      struct  in_addr  imr_multiaddr;
      struct  in_addr  imr_interface;
};

struct  ip_mreq  mreq;
bzero(&mreq, sizeof(mreq));
mreq.imr_multiaddr.s_addr = inet_addr(“235.10.10.3”);
mreq.imr_interface.s_addr = htonl(INADDR_ANY);

setsockopt(sockfd, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq,
                   sizeof(mreq));
```

LINUX网络开发-part3

1 广播和组播

2 UNIX域套接字

3 网络总结

UNIX域套接字

- socket同样可以用于本地通信
- 创建套接字时使用本地协议PF_UNIX(或PF_LOCAL)。

```
socket(AF_LOCAL, SOCK_STREAM, 0)
socket(AF_LOCAL, SOCK_DGRAM, 0)
```

- 分为流式套接字和用户数据报套接字
- 和其他进程间通信方式相比使用方便、效率更高
- 常用于前后台进程通信

用于进程通信

易用：消队 > UNIX > 管道 > 共内

效率：共内 > UNIX > 管道 > 消队

效率：共内

一般：UNIX...

UNIX域套接字

- 本地地址结构

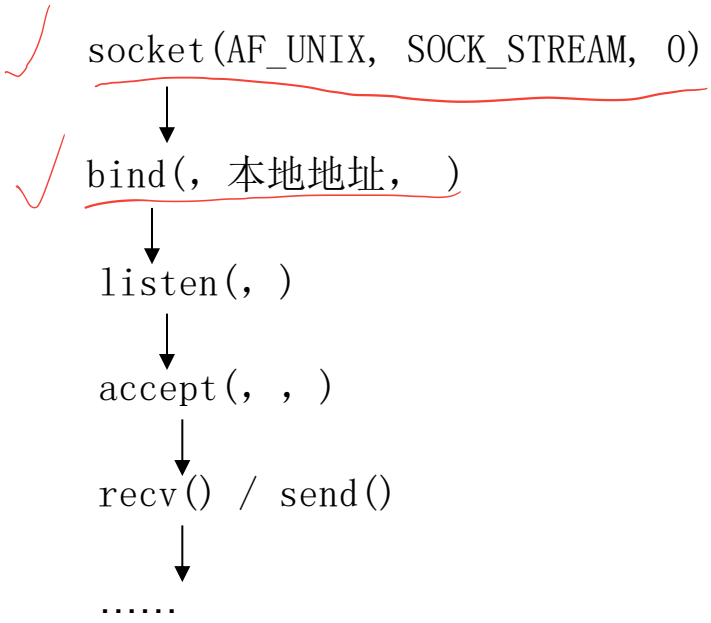
```
struct sockaddr_un          //  <sys/un.h>
{
    sa_family_t  sun_family;
    char  sun_path[108];          // 套接字文件的路径
};
```

- 填充地址结构

```
struct sockaddr_un myaddr;
bzero(&myaddr,  sizeof(myaddr));
myaddr.sun_family = AF_UNIX;
   strcpy(myaddr.sun_path,  “/tmp/mysocket”);
```

UNIX域(流式)套接字

- 服务器端



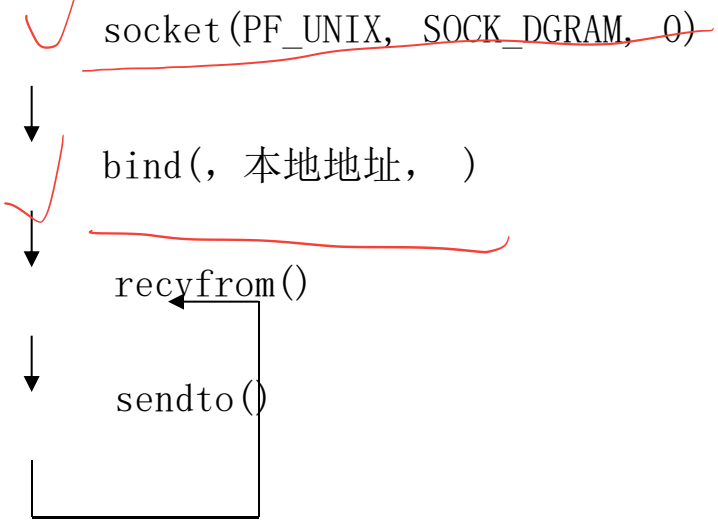
UNIX域(流式)套接字

- 客户端

```
socket(PF_UNIX, SOCK_STREAM, 0)
        ↓
bind(, 本地地址, )   //   可选
        ↓
connect(, , )
        ↓
recv() / send()
        ↓
.....
```

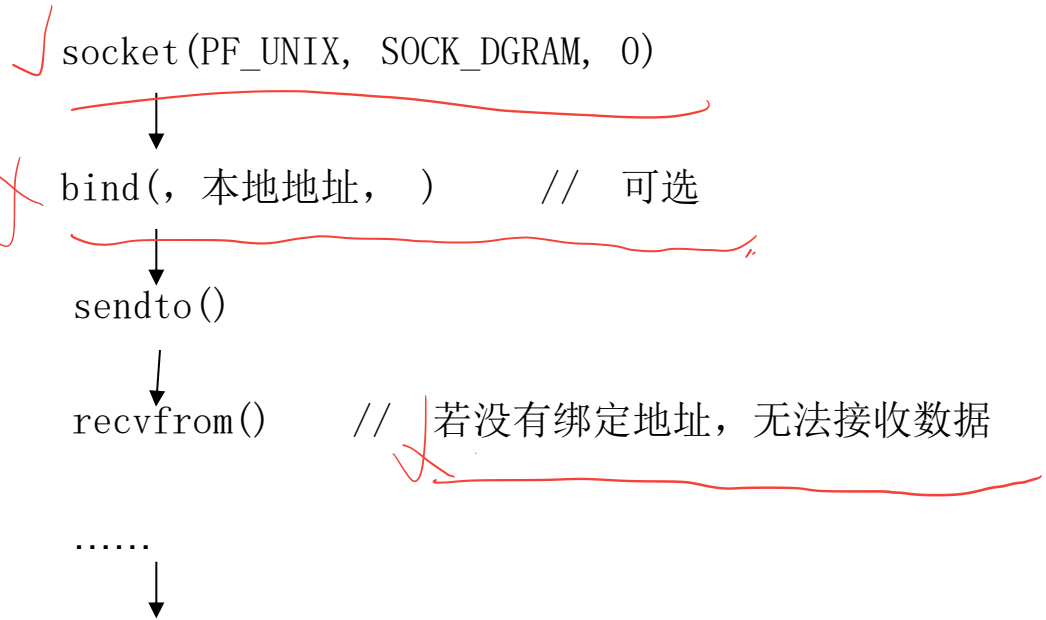
UNIX域(用户数据报)套接字

- 服务器端



UNIX域(用户数据报)套接字

- 客户端



LINUX网络开发-part3

1 广播和组播

2 UNIX域套接字

3 网络总结

扫微信二维码 获取更多信息