**socket之进程间通信篇**

socket：

1. 是一个编程接口；(流式套接字和数据包套接字位于传输层与会话层之间，即内核空间和用户空间之间)
2. 是一个特殊的文件描述符；
3. 是一种进程间通信机制：socket不仅可以用于本地进程间通信；也可用于网络通信。

用于本地进程间通信的socket被称为UNIX域套接字：

1. 在创建socket时，使用本地协议PF\_UNIX/PF\_LOCAL/AF\_UNIX/AF\_LOCAL；
2. 包括两种：流式套接字和用户数据报套接字。

地址结构：

1. internet协议地址结构------struct sockaddr\_in
2. 本地地址结构------struct sockaddr\_un
3. 通用地址结构------struct sockaddr
4. IPV4地址结构------struct in\_addr

注意：

struct sockaddr\_un {

sa\_family\_t sun\_family;//协议族

char sun\_path[108];//套接字文件的路径

};

本地地址结构的填充使用以下代码：

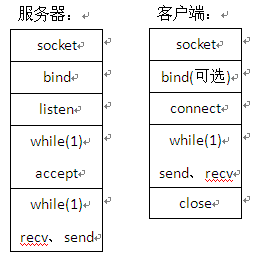
struct sockaddr\_un addr;

memset(&addr, 0, sizeof(addr));

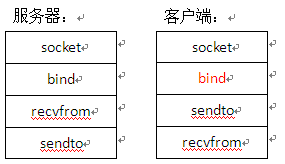
addr.sun\_family = AF\_UNIX;

strcpy(addr.sun\_path, “mysocket”);

UNIX域(流式)套接字



UNIX域(数据报)套接字



注意：

1. 用于本地进程间通信时，bind函数调用成功会在内存中创建出socket文件，且该文件是可见的(有点类似于fifo)；
2. 上面两种通信方式的步骤和TCP、UDP的网络通信步骤基本一致；
3. 在利用流式套接字进行通信时，只需创建一个socket文件；而若采用用户数据报套接字，则需创建两个socket文件，分别属于服务器和客户端；
4. 若指定的socket文件已经存在，则bind函数会报错，故应在bind之前删除同名socket文件，可以通过调用remove函数实现。