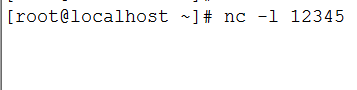
进程间通信首先会想到管道、消息队列、信号量、共享内存以及socket通信。但是一般的进程间通信传递socket句柄，比如使用共享内存的方式，传递的socket文件描述符就被当做普通的整型数值，接收进程拿到这个socket描述符，进行socket通信的时候，底层API就是告知这个描述符是非法的。搜索了一些关于跨进程传递socket句柄的信息(http://bbs.chinaunix.net/thread-4083527-1-1.html,http://blog.163.com/etter\_boy/blog/static/1888803502012392274889/)，得到如下结论：1、两个进程间传递socket句柄有两种方式：管道和UNIX域套接字；2、两个进程间传递socket句柄有两种情况：父子进程间传递（使用非命名管道或者非命名UNIX域套接字）和两个独立进程间传递（使用命名管道或者命名UNIX域套接字）。本次选择命名UNIX域套接字传送文件描述符来实现。

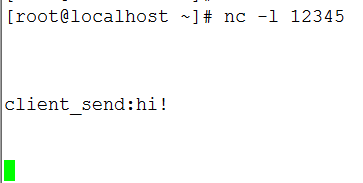
    关于命名UNIX域套接字和传送文件描述符这部分的内容可以参见《UNIX环境高级编程》中17.1-17.4章节。传送文件描述符部分详见17.4.2章，该章节有详细的示例代码(示例代码也可以在网络上查找下载<https://www.cnblogs.com/nufangrensheng/p/3571370.htm>)，下载该示例代码，编译后可以直接运行。命名UNIX域套接字部分详见17..3.1章，该章节也有几段示例代码(<http://www.cnblogs.com/nufangrensheng/p/3569416.html>)，可以整合后进行简单修改，也可以在网上查找其他完整的示例代码（<https://blog.csdn.net/bytxl/article/details/47861469>），进行简单修改后编译运行。（代码详见附件）

    修改示例代码，使其具有如下功能：通过nc命令在本机监听12345端口；进程A通过socket句柄fd与本机12345端口建立连接，并且发送消息“client\_send:hi!”，然后将fd句柄传给进程B后关闭该句柄；进程B收到fd句柄后发送消息“server\_send:hello!”；示例代码运行过程如下：

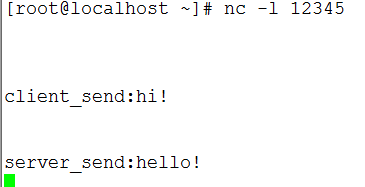
    1)本机监听12345端口，同时运行进程B：



    2) 运行进程A，发送消息后关闭该句柄：



    3) 进程B使用该句柄发送消息：



过程中犯的错误：

1、使用《UNIX环境高级编程》中17.4.2章示例代码传送文件描述符，缺少apu.h，可以安装apu.3e

（https://blog.csdn.net/songshimvp1/article/details/51440545），也可以删除示例代码中apu相关代码；

2、使用网络socket传递文件句柄FILE\* fp：进程B收到的句柄中内容与进程A中创建的句柄内容完全一样，但是进程B无法使用该句柄；

3、使用网络socket通过传送文件描述符的方式传递socket句柄fd：进程B无法收到该socket句柄fd，newfd = \*(int \*)CMSG\_DATA(cmptr);中newfd为0；

4、使用命名UNIX域套接字通过传送文件描述符的方式传递socket句柄fd，达到上述示例代码中效果；

相关参考材料：

通过unix domain socket在多个进程间传递socket套接字：

<https://linux.die.net/man/3/cmsg>

<https://stackoverflow.com/questions/8481138/how-to-use-sendmsg-to-send-a-file-descriptor-via-sockets-between-2-processes>

这个是别人封装的一个库：

[http://www.normalesup.org/~george/comp/libancillary/#s\_2](http://www.normalesup.org/~george/comp/libancillary/" \l "s_2)