

要求屏幕上或者程序上有“LIUYING”目录设置为“LIUYING”

—

分别编一个客户端程序和服务器程序，首先建立客户程序与服务器之间正确的 **socket** 连结，然后利用 **send** 和 **recv** 函数，客户程序将一个较长的文本文件（如>2k 字节）（目的是为了在一次连接测试多次的收/发）中的数据发送给服务器。要求服务器全部正确地接收到所有的数据（一个也不能多、不能少），并将其存入一个文件。

可先传文件的大小（简单的方法是服务器知道文件名和文件的大小）。要修改书上的程序，服务器方生成子进程后要循环地接受数据，直至文件数据全部收到。

TCP 和 UDP 套接字可以任选。

二

编写一个递归的 **shell** 程序(不是习题集中的递归函数) **findit**，该程序搜索参数 2 指定的目录树，查找所有的以 **.c** 和 **.h** 结尾的文件，如文件行中含有参数 1 指定的字符串，显示文件所处的目录名、该行和相应的文件名。如目录参数 2 缺省，则从当前目录中搜索。如：

findit searchstring /home/student/wang

搜索以 **/home/student/wang** 为根的目录树中的 **c** 程序和头文件，查找含有 **searchstring** 字符串的行，显示文件名。

findit searchstring

从当前目录开始搜索。请用含有 **shell** 特殊字符的字符串模式进一步调试该 **shell** 程序。

为了在递归程序中便于查找执行文件 **findit**，例如 **findit** 位于登录主目录下的 **wang** 目录中，可以在当前 **Shell** 中执行：

PATH=.:\$HOME/wang:\$PATH

也可以在递归程序中用绝对路径名执行 **findit**,如：

/home/student/wang/findit

■ 请在主目录下创建 1~n 个子目录和待查的文件

三：

写一段程序实现下面的任意一个内容。

例如：

【存储管理】、或者【进程调度】、或者【死锁】、或者【设备管理】、或者文件系统等仿真实现和 **shell** 程序设计等。

