实验五 简单FTP协议实现

实验内容：

在Unix/Linux系统中，基于socket套接字及进程管理相关系统函数，使用C/C++完成简单FTP的服务器和客户端程序：

具体要求：

1. 基于TCP通信机制，基于守护进程编写一个FTP服务器应答的程序。该服务器进程在收到请求时，能够自动产生新的进程/线程，该进程/线程会完成对FTP客户端请求的做出响应。具体请求-应答内容如下：
   1. 收到“hello”，回复“connected”;
   2. 收到“list”，回复程序当前目录中的文件列表；
   3. 收到“get name.type”，在当前目录中查找文件名为name.type的文件：如果存在，打开该文件并将内容传递回来，客户端将文件存入到自身的当前工作目录中，并以name.type命名保存，待传送完成回复服务器“name.type has been downloaded!”，准备发送新的请求；如果不存在，回复“name.type does not exist in server!”；
   4. 客户端输入“put name.type”后，客户端首先在自身当前工作目录中查找文件名为name.type的文件：如果存在，发送消息“”服务器在工作目录创建以name.type命名的新文件，创建完成等待接收，并回复客户端“Ready”，客户端收到后，打开该文件并将内容发到服务器，服务器将文件存入到自身的当前工作目录中，并以name.type命名保存，待传送完成回复客户端“name.type has been received!”，等待新的请求；如果不存在，客户端自己显示“name.type does not exist in client!”。
2. 基于TCP通信机制，编写客户端程序，配合FTP服务器程序完成上述功能。
3. 在上述FTP服务器程序的基础上，将其改为守护进程并实现多进程（或多线程）并发服务。
4. 在上述程序的基础上，尝试在客户端和服务器端各用两个进程分别实现FTP的命令传送和数据传送。（此内容为最后一个分组实验，可2-3人/组完成，需做PPT报告讲解思路）

实验报告：

1. FTP服务程序为什么要基于TCP协议完成？要获得服务器的响应服务，客户端需要知道什么信息？

因为ftp服务是在网络间进行通信的，而且TCP是可靠连接传输，所以要基于TCP协议完成。客户端需要知道服务器端的地址。

1. 在对具体客户端进程响应时，采用进程和线程有什么差别和联系？你选择的是哪种方式？为什么要选择这种方式（你认为它的优势在哪里）？‘

进程和线程的差别：进程是资源分配的最小单位，而线程是程序执行的最小单位。进程有自己独立的地址空间，每启动一个进程，系统就会为它分配地址空间，建立数据段，栈段，这是十分复杂的操作。而线程不同，线程共享进程的所有资源，它创建的成本比进程小很多。同时，线程之间的通信更加方便。但是进程比线程更加健壮，多线程程序只要有一个线程死掉，整个进程也就没了。而进程死掉对另一个进程不会有影响。

我选择的是多进程，因为对线程的操作不太了解。

1. 在客户端建立连接发出用户请求后，如果服务器断线，是否会出现客户端程序处于死等的情况？为什么？

个人认为应该是会的，因为客户端程序一直在等待服务器回应消息，属于阻塞状态。

1. 以该程序为基础，如何实现基于用户登陆的FTP服务？

在建立连接之后，不能进行任何的操作，否则服务器不予反应，直到客户端输入账户密码之后，禁忌解除。