

# 文件传输服务端设计说明

## 协议设计：

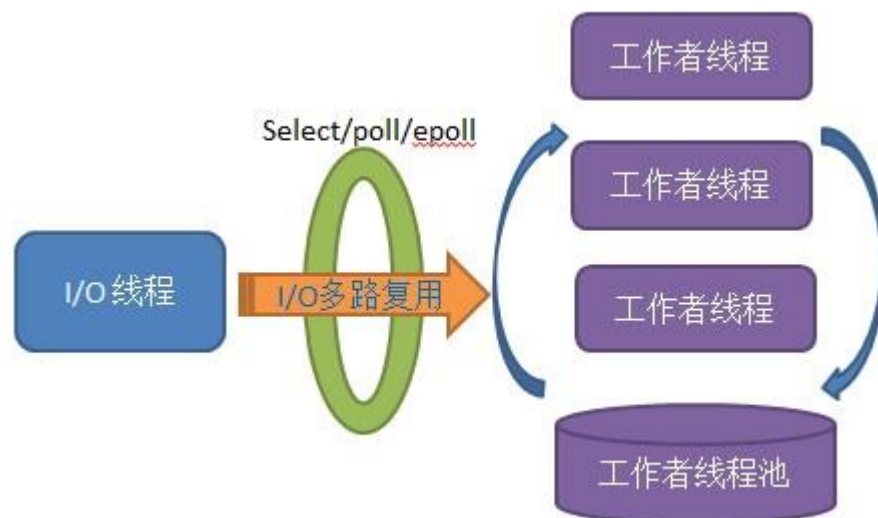
客户端通过运行客户端程序执行 get 或 put 命令向服务器上传文件或下载文件。客户端建立连接后，首先向服务器说明请求的服务类型，即上传还是下载，和文件的名称。服务器收到请求后尝试打开文件，若成功则向客户端发出信号，说明文件的上传或下载可以开始进行，否则出现错误，向客户端发出错误信息后断开与客户端的连接。可能的错误是客户端请求下载的文件在服务器上并不存在。传输完成后服务器向客户端发出确认信后后随即断开连接，客户端收到反馈信息后也将断开连接。

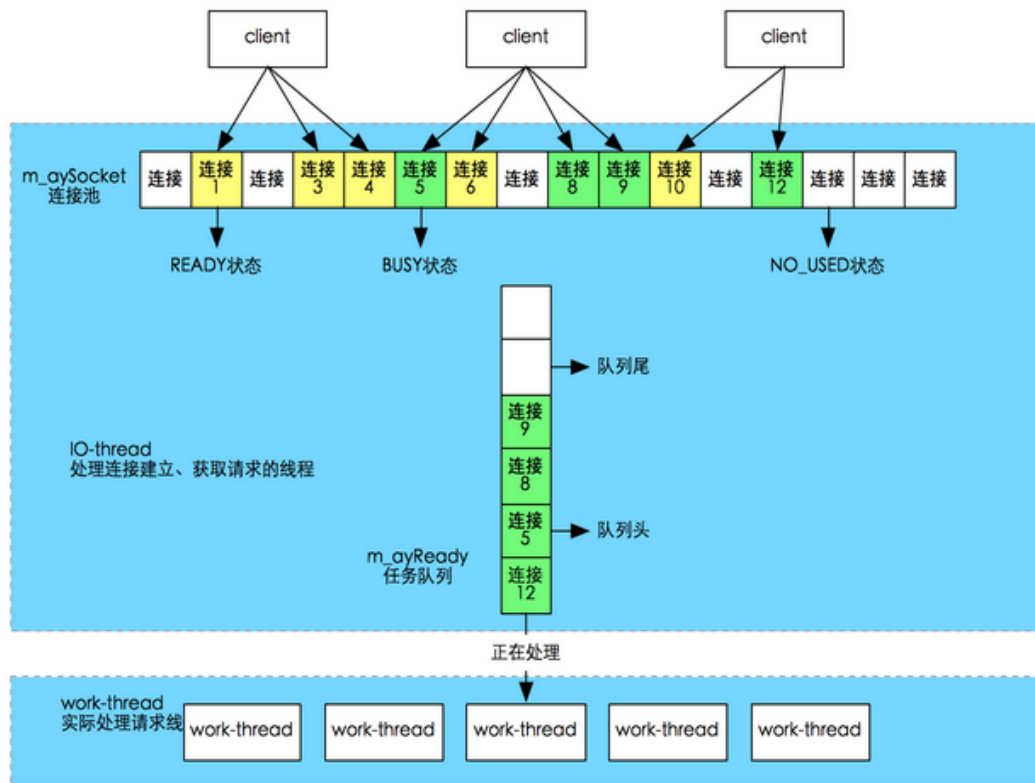
客户端与服务端之间的一次连接仅完成一项上传或下载服务，服务成功完成或者异常退出都将与客户端断开连接。

为简单，本程序不进行不同客户端的账号验证和区分，所有客户端共享服务端同一文件目录，不对同名文件的互斥进行检查。服务端的所有文件对所有客户端同一可读可写。因此当不同客户端试图同时上传或下载同一文件时，可能会出现错误。

## 服务端程序架构设计：

服务端采用 epoll I/O 多路复用模型与线程池的组合。主线程运行 epoll 模型处理 I/O 操作，每次 accept 一个新的连接都新建一个服务并加入工作队列，发出就绪信号通知线程池。而线程池中的工作线程则不断从工作队列中取出等待中的服务并完成。当工作队列为空时，线程池中的工作线程因为等待就绪条件而阻塞。





### 安全性考虑：

设置 SO\_LINGER 选项，避免 FIN\_WAIT\_2 攻击。

设置接收超时时间，避免发呆连接。