

FtpServer 编写

- 1.FTP 是应用层协议，用于实现可靠的文件传输。
- 2.FTP 占用两个端口，21 用于控制连接，20 用于数据连接
- 3.在工作时，FTP 监听 21 端口，每当接收一个连接就创建一对父子进程。父进程为 nobody 进程，负责数据连接的建立、断开，子进程为用户登录进程，负责数据的收发。
- 4.nobody 用户是一个没有任何特殊权限的用户，使用 nobody 主要防止一些破坏性操作。如果需要特殊权限，可以采用 capacities 赋予特殊权限。
- 5.今天的任务：先建立 sysutil 中的函数，然后实现基本的框架：必须以 root 启动程序（在程序中检测），每当接受连接就创建一对父子进程，子进程回发一个“Hello”。
- 6.文件描述符的传递：因为进程空间是相互独立的，所以一个进程打开的 fd 不能直接为另一个进程使用，此时传递描述符需要使用 sendmsg 以及 recvmsg，实现的关键是正确填充 struct msghdr 这个结构体。
- 7.超时 IO：希望给 read、recv、accept 此类的函数加上一个超时时间。期望实现这种效果：IO 可用就立刻返回，如果不可用就最多阻塞某时间。
- 8.实现方式：可以使用 socket 选项来设置 fd 的超时时间。另外就是采用 select 模型。实现思路是：让 select 模型只监听

单个 fd，同时设置相应的超时时间，这样一旦 select 超时就意味着系统调用超时。read、write 以及 accept 的实现思路就是如此。

9.对于 connect 的超时实现：因为连接建立需要一定的时间，所以要把 connect 设置为非阻塞。这样 connect 会立刻返回，如果为 0 代表 connect 成功，但是为 -1 并不意味着失败，此时（前提 errno 为 EINPROGRESS）需要用 select 继续监听，当 connect 真正建立的时候，fd 可读，select 返回，否则 select 最多阻塞用户设置的时间，同时判定连接失败。

10.一个主机可能有多个网卡，这意味着同一台主机可以存在多个 IP 地址，INADDR_ANY 代表当 accept 一个新连接时，可以绑定在任何一块网卡上。在实际中，可能需要设置 server 的 IP 地址，然后 bind 这个地址，于是之后建立的任何连接都是使用的这个 IP 地址，例如“192.168.44.133”，同时意味着只有“192.168.44.X”网段的用户才可以连接这台主机。

11.readn、readLine、writen 全部是基于阻塞 IO 的。

12.在网络编程中使用非阻塞 fd，一定要具有缓冲区。

13.以发送数据为例，用户把数据放入 Buffer 里面，然后在 buffer 调用 write 函数，但是未必一次写完，如果有残留数据，此时需要在 epoll 中注册该 fd 的写事件，当 fd 可写时，继续写入。当缓冲区清空时，从 epoll 中移除该 fd 的写事件。

14.在非阻塞 IO 中，如果 connect 返回 -1，且 errno 为

EINPROGRESS, 那么需要向 **epoll** 注册该 **fd** 的读事件。