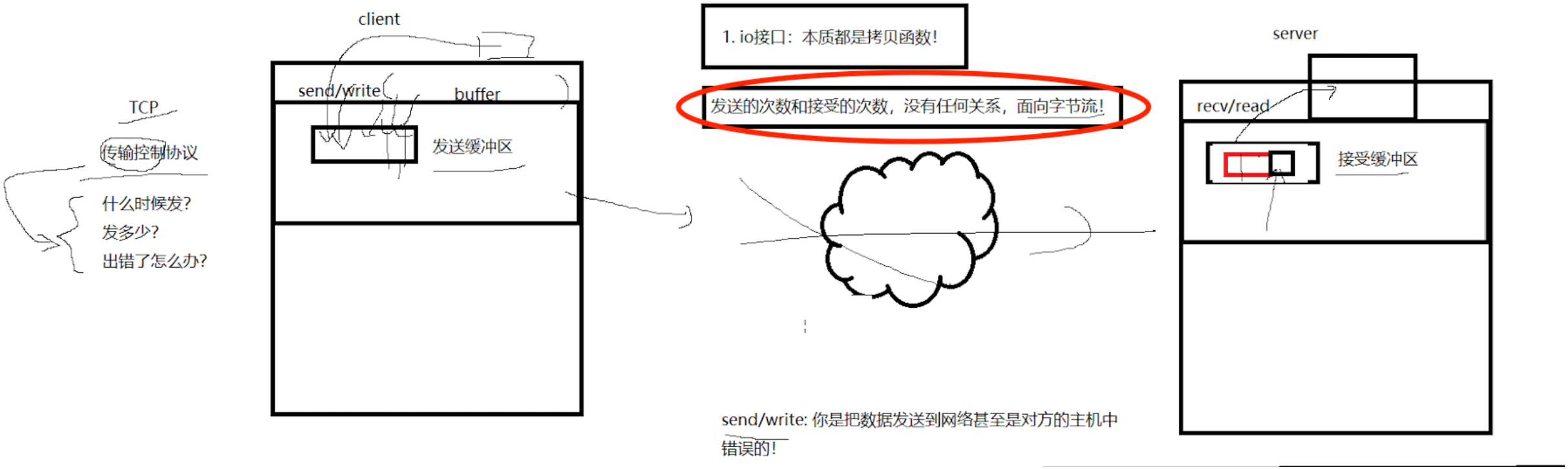
# 0226应用层序列化和反序列化




现在我们已经写了很多网络服务器了
流程也很熟悉了
现在我们要来弄一个操作 --- 守护进程!

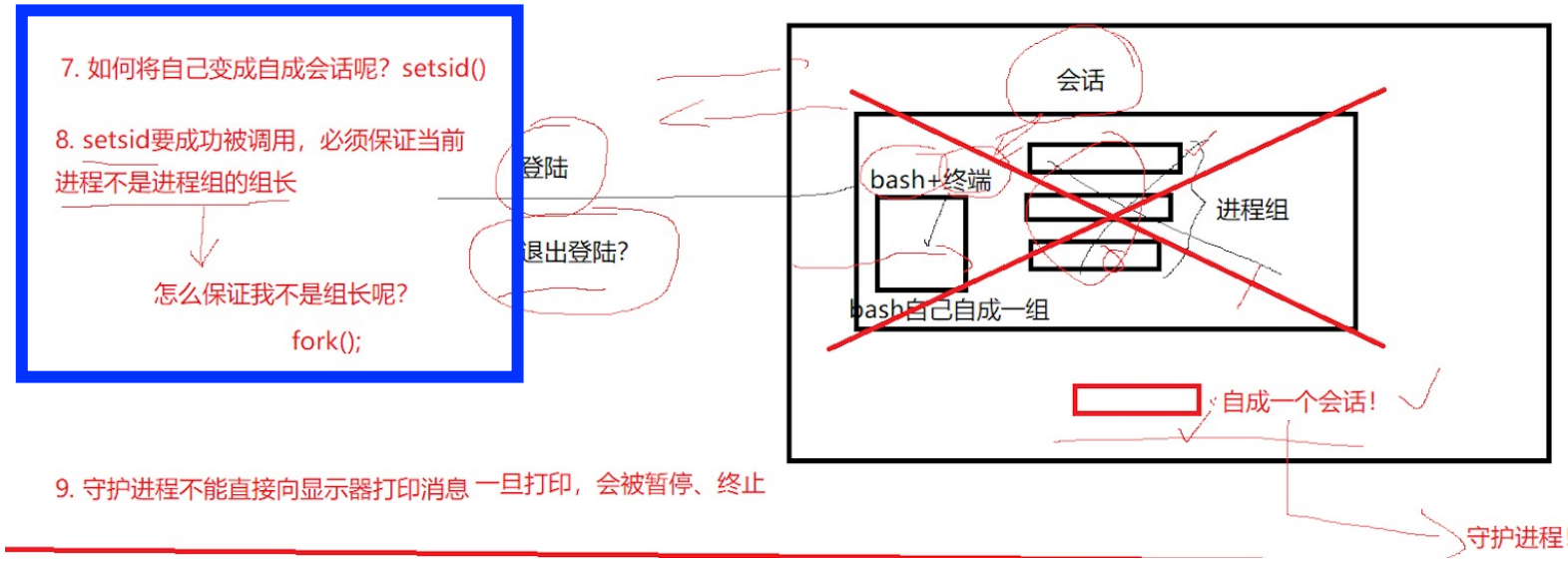
# 守护进程

现在我们看到的很多进程，大部分都是在前台运行的

1. 前台进程：和终端关联的进程
2. 任何终端登录，只允许一个前台进程和多个后台进程
3. 进程除了有自己的pid，也有组id
4. 在命令行中，同时用管道启动的多个进程，他们是兄弟关系，父进程都是bash -> 可以用匿名管道来通信
5. 而同时被创建的多个进程，可以成为一个进程组的概念，组长一般是第一个进程
6. 任何一次登录，登陆的用户都需要有多个进程（组），来给这个用户提供这个服务的

所以当终端关掉的时候，
那么这个会话的的东西就会释放掉

但是现在我们需要一个，关掉终端，
服务器也不要退出！



如何在Linux中正确的写一个让进程守护进程化的一个代码?

我想通过自己写一个函数，让我们的进程调用这个函数，自动变成守护进程

```
[whb@bite-alicloud NetCal]$ echo "hello" > /dev/null
[whb@bite-alicloud NetCal]$ cat < /dev/null
```

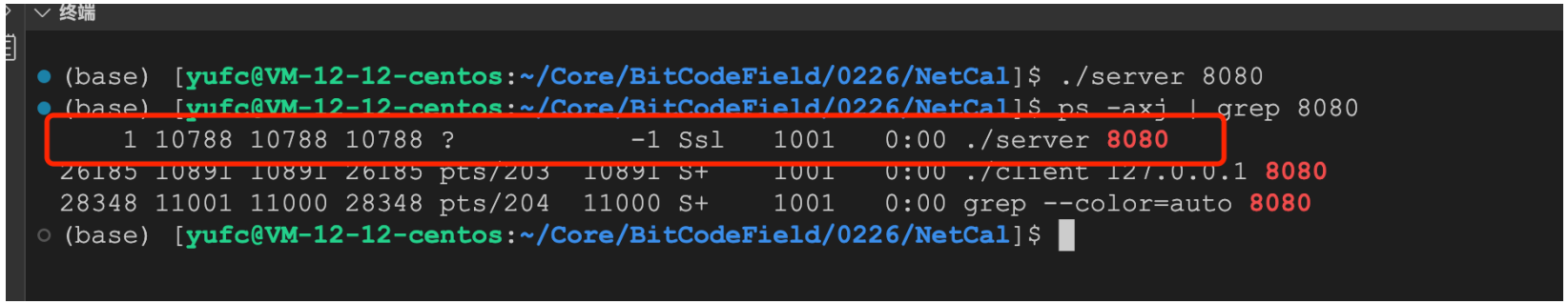
Linux操作系统一般都有一个叫做/dev/null的文件
他就是一个类似于文件黑洞的东西

我们往里面写东西，不管你
读东西，可以读，但是你什么也读不到

```
(base) [yufc@VM-12-12-centos:~/Core/BitCodeField/0226/NetCal]$ ll /dev/null
crw-rw-rw- 1 root root 1, 3 Jan  7 19:39 /dev/null
(base) [yufc@VM-12-12-centos:~/Core/BitCodeField/0226/NetCal]$ echo "hello" > /dev/null
(base) [yufc@VM-12-12-centos:~/Core/BitCodeField/0226/NetCal]$ cat /dev/null
(base) [yufc@VM-12-12-centos:~/Core/BitCodeField/0226/NetCal]$
```

```
96      */
97
98      std::unique_ptr<yufc_tcpServer::TcpServer> server(new yufc_tcpServer::TcpS
99      server->BindService(Calculator);
100     MyDaemon();
101     server->Start();
102     return 0;
103 }
```

此时server就是守护进程了
是后台的
运行就和当前bash没有关系了




守护进程的父进程是1
所以
守护进程就是孤儿进程的一种
只不过是孤儿进程可能还属于某个会话
但是守护进程不属于任何一个会话

# 以上，就是我们自己写的应用层！