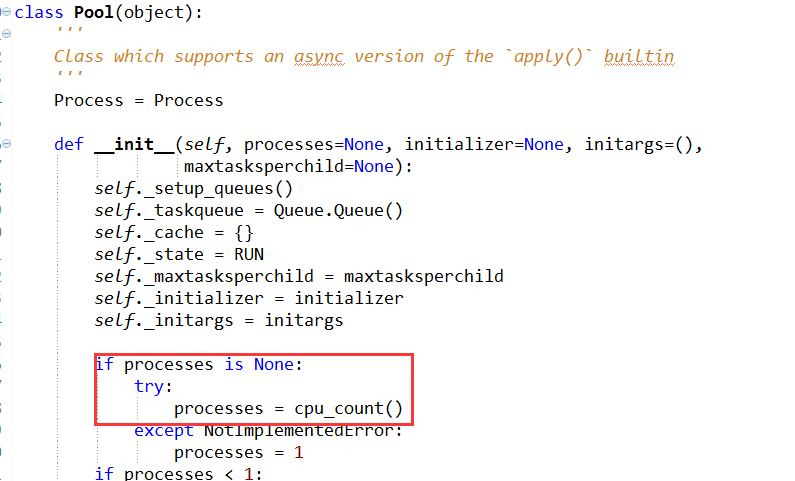
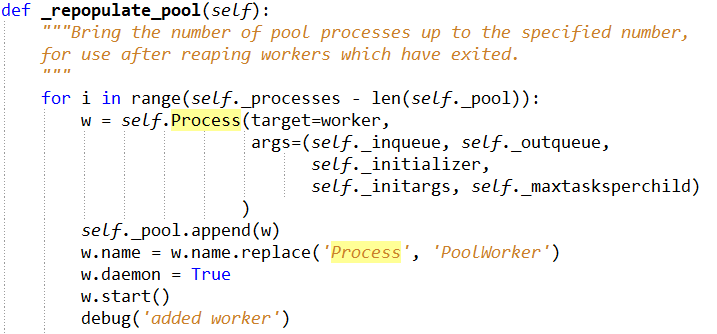
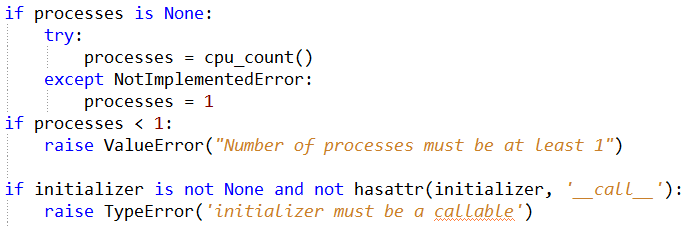
1.不指定进程数，默认就是CPU个数：



2.初始化的时候，用multiprocessing.Process创建进程：



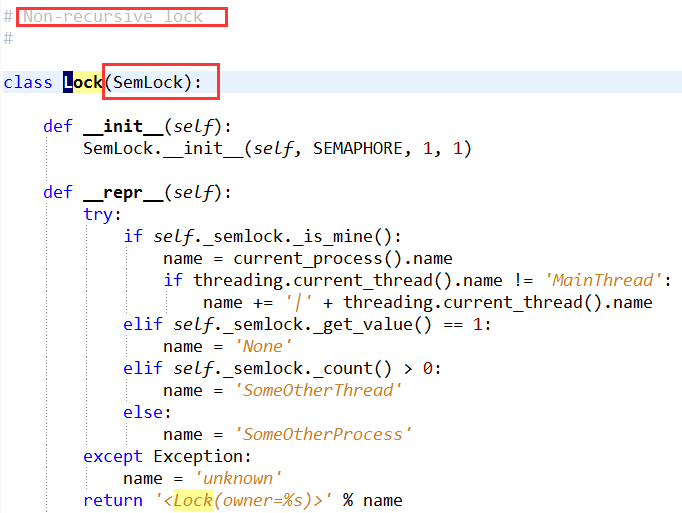
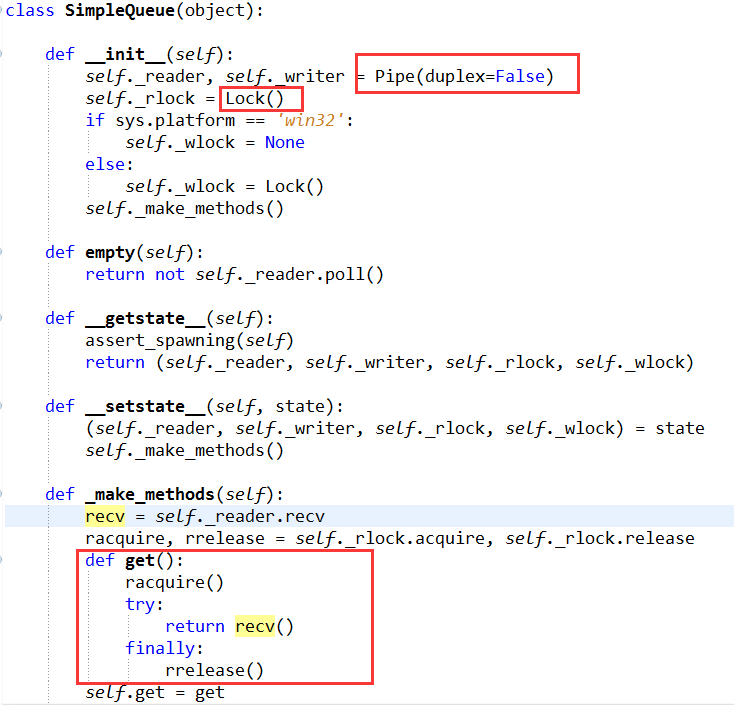
3.对于异常的使用，可以学一下：



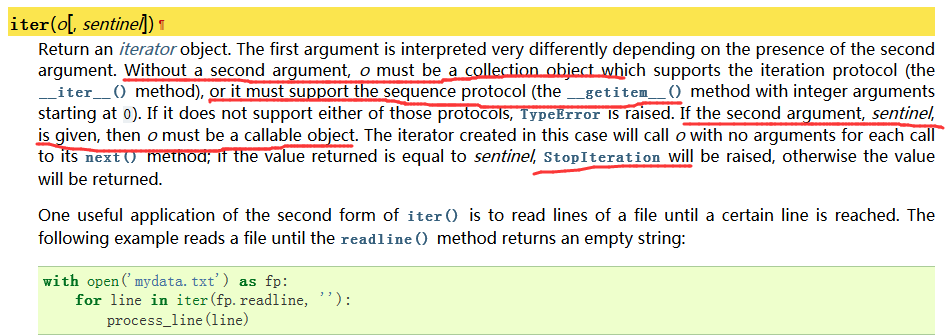
4.使用SimpleQuque作为进程间通信：

a.内部实现为管道

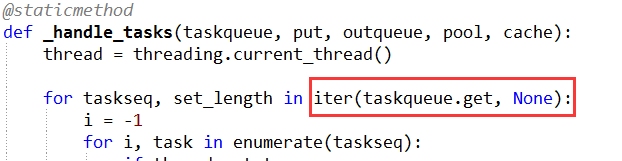
b.加上互斥锁（非递归锁）保护

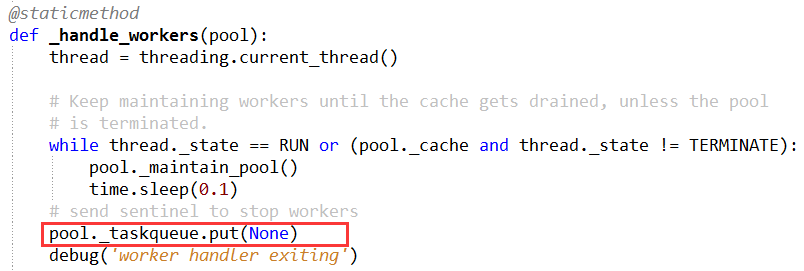


5.iter的使用：

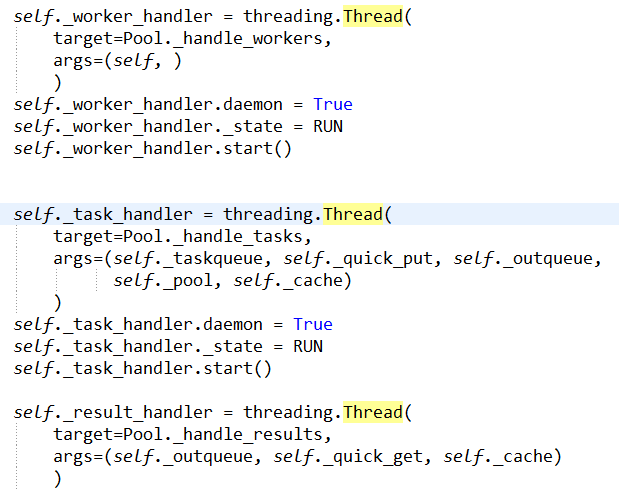


在\_handle\_tasks用来判断是否结束（队列有个None）：

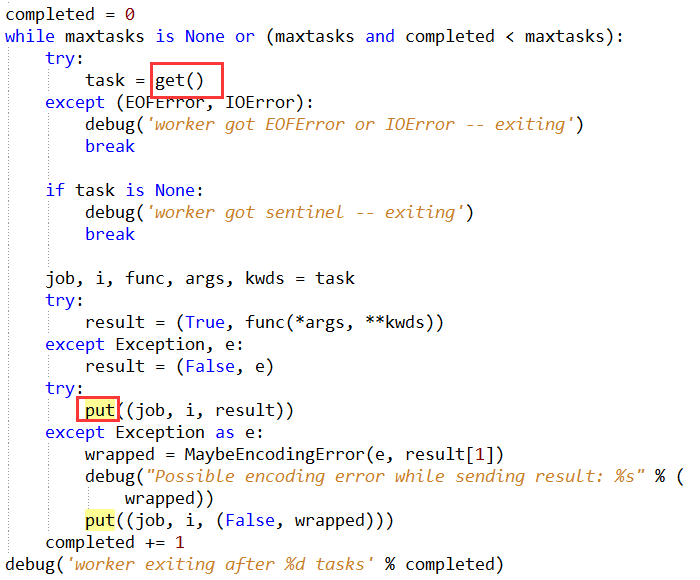




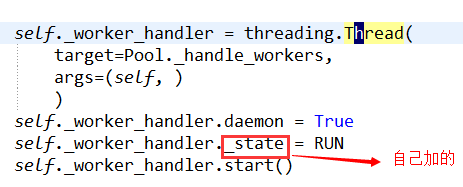
6.开了多个线程来管理：



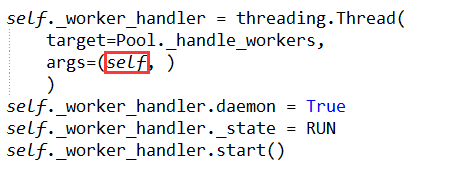
7.可以看出，pool创建的子进程，都是简单的生产者消费者模型（不可不说，这种代码风格不错，看起来好舒服，而且一目了然，跟我的一样，吼吼）：



8.python的对象可以自己随便加个字段：



9.为啥传个self给线程：



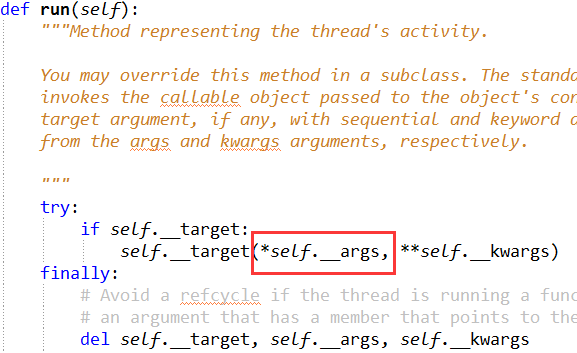
因为：

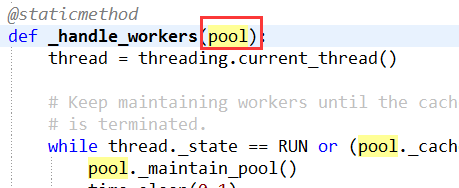
Thread.start

-> \_\_bootstrap

-> \_\_bootstrap\_inner

-> run





10.根据代码，猜测3个状态值（pool.\_state）的意义：

1) RUN

运行中

2) CLOSE

正在关闭

3) TERMINATE

已经关闭

11.Process -> Popen

Popen的作用:

a.初始化stdou和stderr

b.执行fork

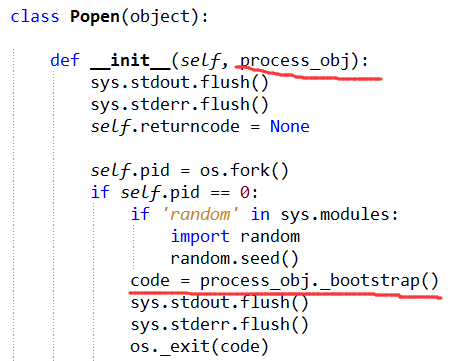
1）子进程

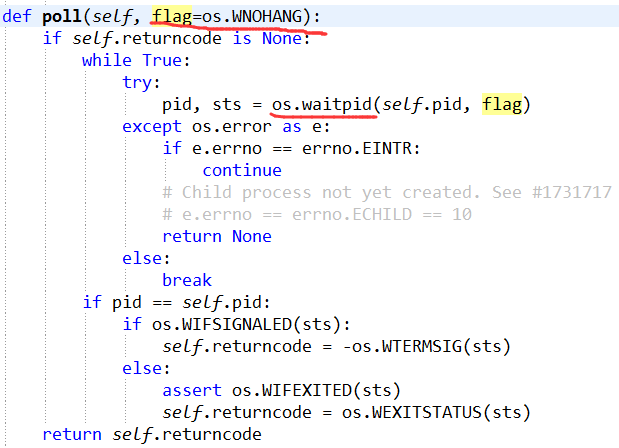
a.执行 Process的\_bootstrap

b.获取renturncode

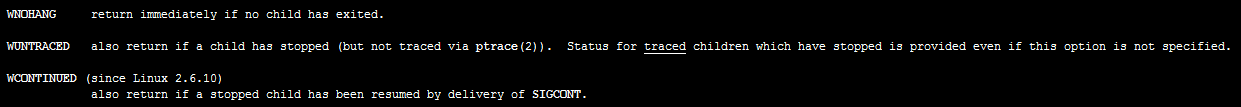
2）父进程

a.提供wait和poll和terminate









EINTR means "This call did not succeed because it was interrupted. However, if you try again, it will probably work."  
  
In other words, EINTR is not a fatal error -- it just means you should retry whatever you were attempting.

http://man7.org/linux/man-pages/man7/signal.7.html

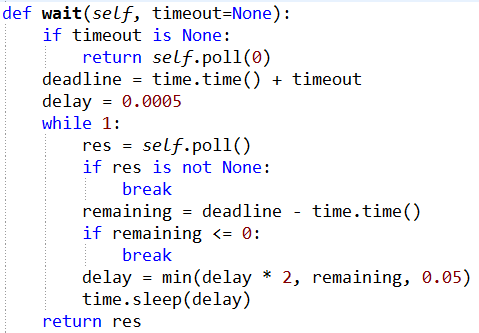
On Linux, even in the absence of signal handlers, certain blocking

interfaces can fail with the error **EINTR** after the process is stopped

by one of the stop signals and then resumed via **SIGCONT**. This

behavior is not sanctioned by POSIX.1, and doesn't occur on other

systems.



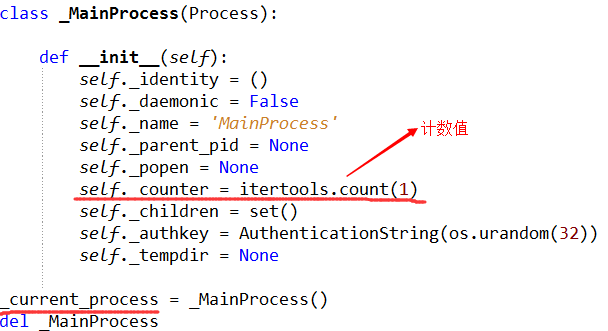
注意：这里的wait()调用的是poll(0)，poll(0)表示的是一直阻塞（没有使用WNOHANG），但是又没有对SIGCHILD做处理，因此会一直阻塞，等到子进程结束了，就会抛异常了ECHILD，导致返回的returncode是None，但是其实子进程已经结束了。

ECHILD

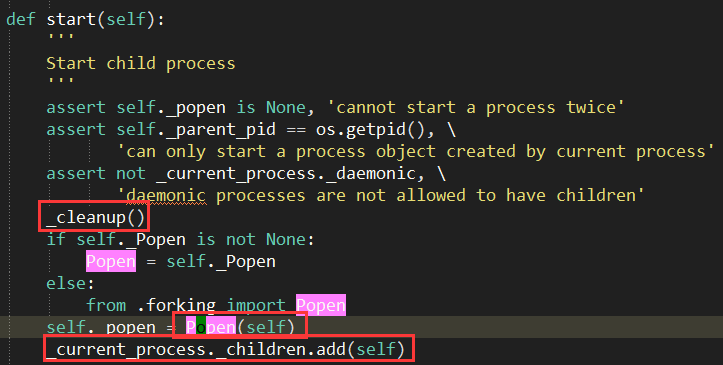
(for waitpid() or waitid()) The process specified by pid (waitpid()) or idtype and id (waitid()) does not exist or is not a child of the calling process. (This can happen for one's own child if the action for SIGCHLD is set to SIG\_IGN. See also the Linux Notes section about threads.)

If the signal SIGCHLD is ignored (which is the default) and you skip WNOHANG as mentioned by Jonathan, waitpid will hang until the child exits and then fail with code ECHILD.

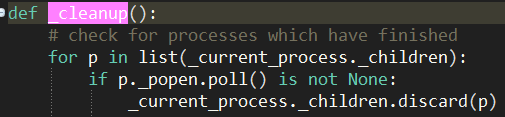
12.子进程的process类



import process的时候，就会生成一个进程对象（不是一个真正的进程，只是一个对象），名字是\_current\_process，作为管理进程（记录进程个数之类的）。然后还可以把类给删掉，这个倒是有点意思啊。



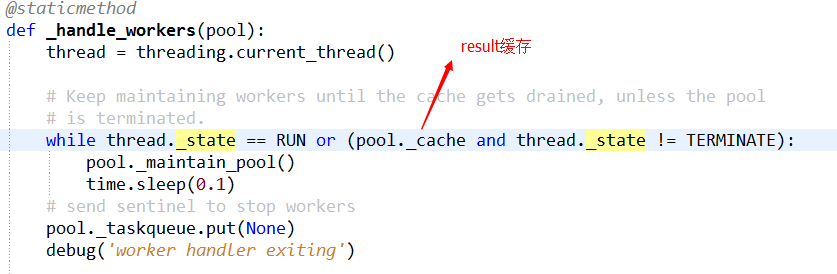
a) 先清理：



b) 实际生成进程是用Popen

c) 加入管理全局对象

11.进程管理thread:



12.python的弱引用：

https://pymotw.com/2/weakref/