

在Linux中执行python程序中一般都是python3 xxx.py

而前面的python3也是一个程序，是一个解释器程序，由C语言写的解释器叫cpython，java写的解释器的jpython，由于历史原因，默认采用的是cpython，而cpython无法充分的发挥多线程的功能，而GIL在官方中不易移除，因此有了GIL锁，但使用jpython解释器可以解决GIL问题

因此，多进程比多线程更浪费资源的前提是使用jpython解释器的前提下，倘若在cpython的解释器下执行，如果想发挥多核cpu的能力，可以使用多进程

计算密集型：计算、没有延时的操作（recv，sleep等等）这个时候不适于线程，建议用进程

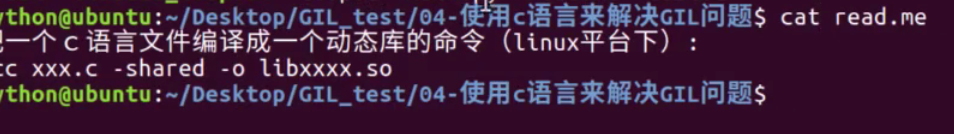
Io密集型（input、output）：有延时的操作，可以应用线程

协程适用于io密集型，因为gevent也是充分利用等待时间，计算密集型没有等待时间

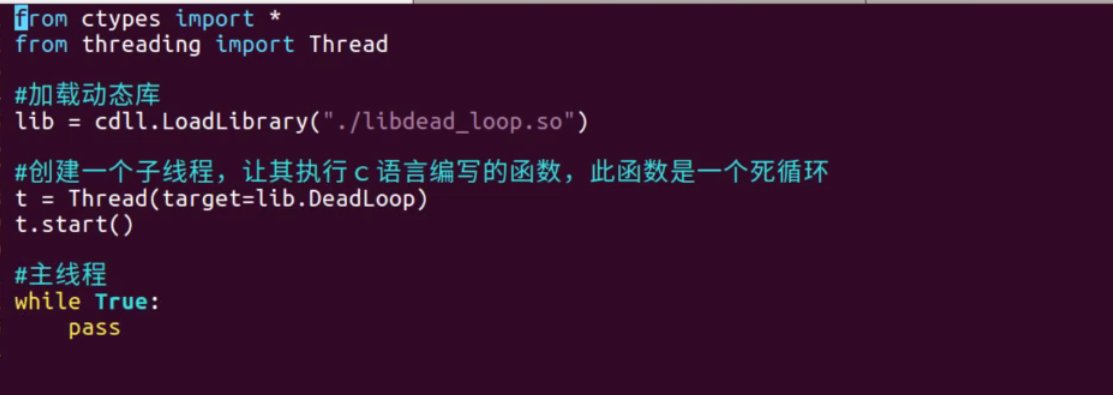
计算密集型：进程

Io密集型：线程、协程

Python是一门胶合型语言，python可以直接调用其他语言写的代码（包括C语言，c++，java，JavaScript等等）



通过用特殊的命令来编译，把这个文件编译成 .so 文件，这就叫做动态库，简而言之就是一个二进制文件



通过以上方法来调用其他语言的代码

如何解决GIL问题？

1. 换一种解释器，把python解释器换成其他语言写的解释器，避免C语言写的解释器
2. 用其他的语言来替代将来线程里面做的事情