fork编程思路

1、父进程只负责生成子进程

2、子进程做具体的工作，做完后务必结束退出

僵尸进程：

1、父进程还在

2、子进程没有任何可执行代码了

3、父进程没有处理僵尸进程的代码

fork网络编程思路：

1、父进程只负责接收新的客户端，生成子进程与客户端通信

2、子进程与客户端收发数据，进行通信

时分复用、时间片

GIL：全局解释器锁。它限制某一时刻只能有一个线程在使用解释器，多个线程需要轮流使用

这样，导致某一时刻只有一个线程在工作

计算密集型应用(CPU密集)：多进程可以提升效率

IO密集型应用：多进程和多线程都可以提升效率

exit() 在for重可以杀死进程，continue是不行的

(‘172.1.1.%d’ % i for i in range(1-255)) 是生成器

父进程的os.fork返回值是子进程的pid