

死锁是指两个或两个以上的进程在执行过程中，因争夺资源而造成的一种互相等待的现象，若无外力作用，它们都将无法推进下去。

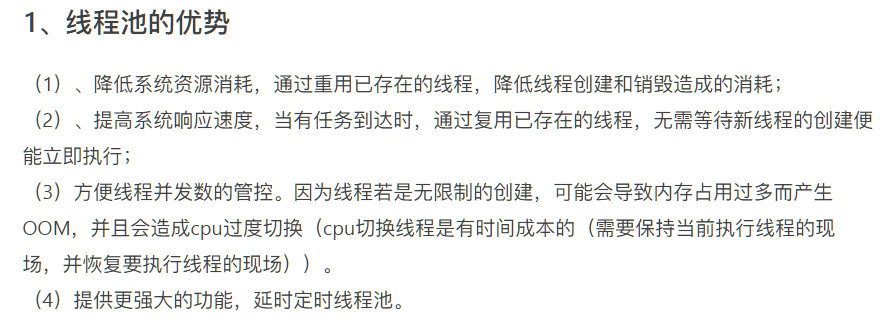
死锁的发生必须满足以下四个条件：

- 互斥条件：一个资源每次只能被一个进程使用。

- 请求与保持条件：一个进程因请求资源而阻塞时，对已获得的资源保持不放。

- 不剥夺条件：进程已获得的资源，在末使用完之前，不能强行剥夺。

- 循环等待条件：若干进程之间形成一种头尾相接的循环等待资源关系。



* **newFixedThreadPool**创建的线程池corePoolSize和maximumPoolSize值是相等的，它使用的LinkedBlockingQueue；
* **newSingleThreadExecutor**将corePoolSize和maximumPoolSize都设置为1，也使用的LinkedBlockingQueue；可以保证任务的顺序执行。
* **newCachedThreadPool**将corePoolSize设置为0，将maximumPoolSize设置为Integer.MAX\_VALUE，使用的SynchronousQueue，也就是说来了任务就创建线程运行，当线程空闲超过60秒，就销毁线程。
* **newScheduledThreadPool**()
* **newWorkStealingPool**() （后台线程）工作窃取 每个线程维护一个线程队列，当自己队列任务执行完会去主动偷取别的线程队列中的任务。



