线程的优雅关闭实践

平时开发中，大家更多地关注的是线程池的创建、任务的提交和执行。往往会忽略线程池的关闭，甚至忘记调用shutdown()方法，导致内存溢出。大多只知道需要调用shutdown()关闭线程池，却少研究其真正的关闭过程。

首先看源码中的一句注释：

A pool that is no longer referenced in a program and has no remaining threads will be shutdown automatically.  
如果程序中不再持有线程池的引用，并且线程池中没有线程时，线程池将会自动关闭。

线程池自动关闭有两个条件：

1. 线程池的引用不可达；
2. 线程池中没有线程；

这里对于条件2解释一下，线程池中没有线程是指线程池中的所有线程都已运行完并自动消亡。然而我们常用的FixedThreadPool的核心线程没有超时策略，所以并不会自动关闭。

展示两种不同线程池不关闭的情况：

1. FixedThreadPool示例
2. CachedThreadPool示例