

除了使用ThreadPoolExecutor，还有没有其他方式创建线程池？

有的，可以通过Executors去进行创建，分别有：

newCachedThreadPool：创建一个可以根据需要创建新线程的线程池，如果有空闲线程，优先使用空闲的线程

newFixedThreadPool：创建一个固定大小的线程池，在任何时候，最多只有N个线程在处理任务

newScheduledThreadPool：能延迟执行、定时执行的线程池

newWorkStealingPool：工作窃取，使用多个队列来减少竞争

newSingleThreadExecutor：单一线程的线程池，只会使用唯一一个线程来执行任务，即使提交再多的任务，也都是会放到等待队列里进行等待

newSingleThreadScheduledExecutor：单线程能延迟执行、定时执行的线程池

上面这几种方式，底层还是使用ThreadPoolExecutor创建线程池，且如果使用不当，容易造成oom

newFixedThreadPool newSingleThreadExecutor

允许的请求队列长度为 Integer.MAX_VALUE，可能会堆积大量的请求，从而导致 OOM。

newCachedThreadPool newScheduledThreadPool

允许的创建线程数量为 Integer.MAX_VALUE，可能会创建大量的线程，从而导致 OOM

此时，如果能结合某个具体例子进行说明，效果会更加好，比如：

在某次线上发生OOM,通过分析程序因为无法创建新的线程而OOM，初步判断可能是线程过多导致的OOM，整个应用存在大量线程，再从代码分析，是由于使用了newCachedThreadPool创建线程池，在该时刻提交的任务数过大，导致的OOM。后面改为new ThreadPoolExecutor，显示指定拒绝策略