概述

使用

预想的用法

而线程池是没有办法按照池化的思想来设计的。

实际的用法

生产者-消费者模式

也就是说在使用线程池时，我们编写的代码充当的角色类似于生产者，要往线程池中放任务即可，线程池自己会获取任务并执行。

底层实现

Executor框架

1. 对任务的抽象（Runnable/Callable）

2. 对任务执行的抽象(Executor)

3. 异步计算结果的抽象(Future)

异步

Executor的执行流程

ThreadPoolExecutor 类

阻塞类队列

线程池会按照定义的饱和策略来拒绝任务

既不抛弃任务也不抛出异常

其他线程池方案

FixedThreadPool

不足之处：

需要注意

OOM。

SingleThreadExecutor

不足之处：

CachedThreadPool

因此CachedThreadPoll线程在遇到新任务时会不断创建新线程来执行。

不足之处：

总结

阻塞类队列

线程池会按照定义的饱和策略来拒绝任务

既不抛弃任务也不抛出异常

引用