# 1 线程池

## 1.1 线程封装

对Linux下的有关线程的接口进行封装

与线程有关的还有互斥锁和条件变量

## 1.2 生产者-消费者问题

一个多线程同步问题的经典案例

该问题描述了两个共享固定大小缓冲区的线程——即所谓的“生产者”和“消费者”——在实际运行时会发生的问题。

生产者的主要作用是生成一定量的数据放到缓冲区中，然后重复此过程。

与此同时，消费者也在缓冲区消耗这些数据。

该问题的关键就是要保证生产者不会在缓冲区满时加入数据，消费者也不会在缓冲区中空时消耗数据。

###### 1.3 线程池

面向对象编程中，创建和销毁对象是很费时的，因为创建一个对象要获取内存资源或者其它更多资源。

所以提高服务程序效率的一个手段就是尽可能减少创建和销毁对象的次数，特别是一些很耗资源的对象创建和销毁。

基本思想还是一种对象池的思想，开辟一块内存空间，里面存放了众多(未死亡)的线程，池中线程执行调度由池管理器来处理。当有线程任务时，从池中取一个，执行完成后线程对象归池，这样可以避免反复创建线程对象所带来的性能开销，节省了系统的资源。

一个简单的线程池实现:

线程池组件：

线程池管理器(ThreadPool):用于创建并管理线程池

工作线程(WorkThread): 线程池中的线程

任务接口(Task):每个任务必须实现的接口，以供工作线程调度任务的执行。

任务队列(Queue):用于存放没有处理的任务。提供一种缓冲机