iOS线程文档

一、定义。  
 什么是进程：在系统中正在运行的一个应用程序。每个进程都是独立的，每个进程均运行在其专用而且受保护的内存空间内。  
 什么是线程：一个进程想要执行任务，必须得有一个线程。进程中的所有任务都在线程中执行。  
 串行：一个线程中的人物都是串行执行的。如果要在一个线程中执行多个任务，那么都是按顺序一个个执行。同一个时间内，一个线程只能执行一个任务。  
 多线程：一个进程中可以开启多条线程，每条线程可以并行执行不通的任务。  
 多线程原理：同一时间，cpu只能处理一条线程，只有一条线程在工作。多线程并发，其实是cpu快速的在多条线程之间切换。如果多线程切换速度特别快，就造成线程并发执行的假象。  
 多线程优缺点：  
 优点：能适当提高程序执行效率。能够适当提高资源利用率。  
 缺点：创建线程是有开销的，iOS成本：内核数据结构，栈空间。创建线程大约需要90毫秒。线程过多cpu消耗大。使用过多会使程序设计更加复杂。  
  
二、实现方式  
2.1、GCD  
 优势：系统提供，纯C语言，系统多和并行运算提高的方案，自动管理线程生命周期。自动利用跟多的CPU。  
 四个概念：同步，异步，并发（DISPATCH\_QUEUE\_CONCURRENT），串行（DISPATCH\_QUEUE\_SERIAL）。  
 同步和异步主要影响能不能开线程。

串行和并行主要影响任务的执行方式。

使用栅栏函数是不能是全局队列，最好是自己创建队列。

GCD各种组合结果：

2.2、NSOperation。  
 配合使用NSOperation和NSOperationQueue也能实现多线程编程。  
 默认情况下，调用了start方法后并不会开一条新线程去执行操作，而是在当前线程同步执行操作。  
 只有NSOperation放到一个NSOperationQueue中，才会异步执行。  
 NSBlockOperation封装的操作数大于1，就会异步执行。  
 如果将NSOperation添加到NSOperationQueue(操作队列)中，系统会自动异步执行NSOperation中的操作。queue.maxConcurrentOperationCount设置最大并发操作数。  
 当队列调用了队列挂起的方法（ self.queue.suspended = YES;），队列里的执行方法立即停止，但是有一点需要注意的是，当block操作中，队列挂起是不起作用的，它是无法停止的，必须操作执行结束后才会生效。  
 当队列调用取消（ [self.queue cancelAllOperations]）就意味着后续队列不再执行，再次启动需要重新加入队列。  
 设置队列监听与依赖：addDependency。  
2.3、NSThread  
 面向对象，简单易用，可以直接操作线程对象。  
 需要自己管理线程生命周期。  
2.4、pthread\_t  
 操作系统级别的线程。  
 基于C语言实现，使用难度大。  
 需要手动管理生命周期。

三、线程安全，锁。  
 锁的方式有以下：@synchronized、 NSLock、NSRecursiveLock、NSCondition、NSConditionLock、pthread\_mutex、dispatch\_semaphore、OSSpinLock、atomic(property) set/ge等等各种方式。