

展 E 宝 周汇第一期

1.iOS同步锁的介绍

2.DNS服务器的简单介绍

3.网络分层的简单介绍

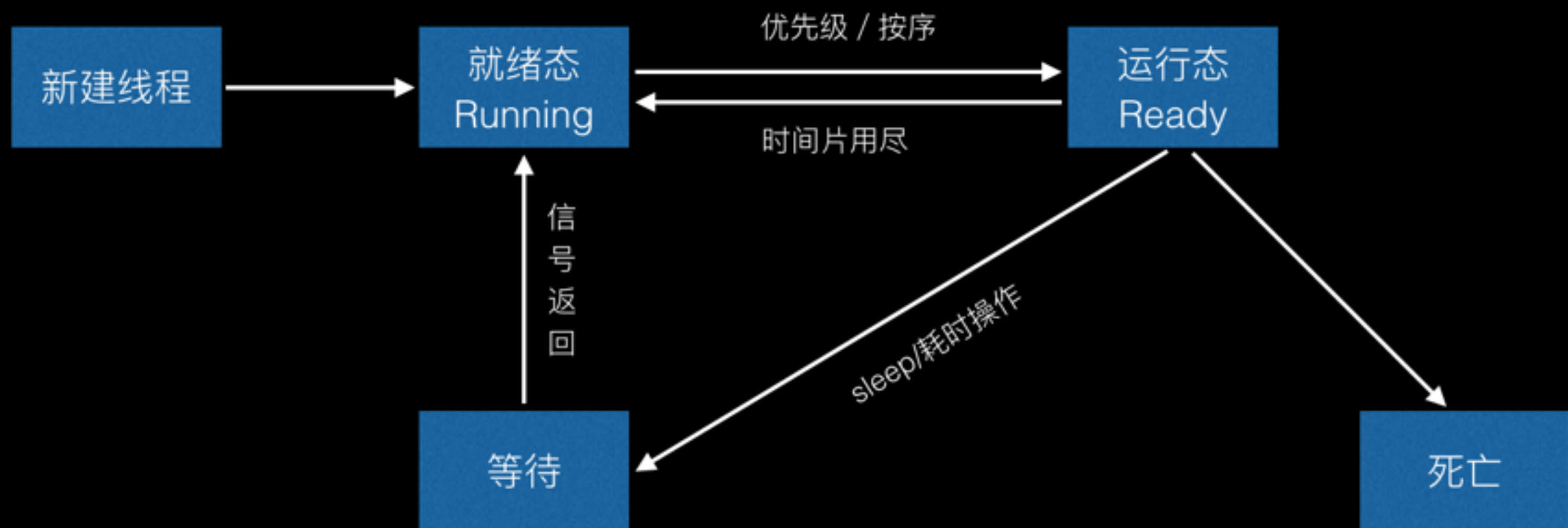
4.URL编码的简单介绍

多线程的概念

- 什么是线程
- 线程，有时被称为轻量级进程（LWP），是程序执行流的最小单元。一个标准的线程由线程ID，当前指令指针，寄存器集合和堆栈组成。
- 大多数的软件应用中，线程的数量都不止一个。多个线程可以互不干扰地并发执行，并共享的全局变量和堆的数据。

多线程的使用场景

- 1.某个操作可能会陷入长时间等待，等待的线程会进入睡眠状态，无法继续执行，多线程执行可以有效利用等待的时间。典型的例子是等待网络响应。
- 2.某个操作（类似有算法的方法）会消耗大量的时间，如果只有一个线程，程序 and 用户之间的交互会中断，多线程可以让主线程负责交互，另一个线程负责计算。
- 3.程序本身逻辑就要并发，例如下载程序
- 4.多核计算机，本身具有同时具有多个线程的能力（充分利用硬件优势）
- 5.多线程主数据共享方面效率要高很多



线程调度

线程调度的准则

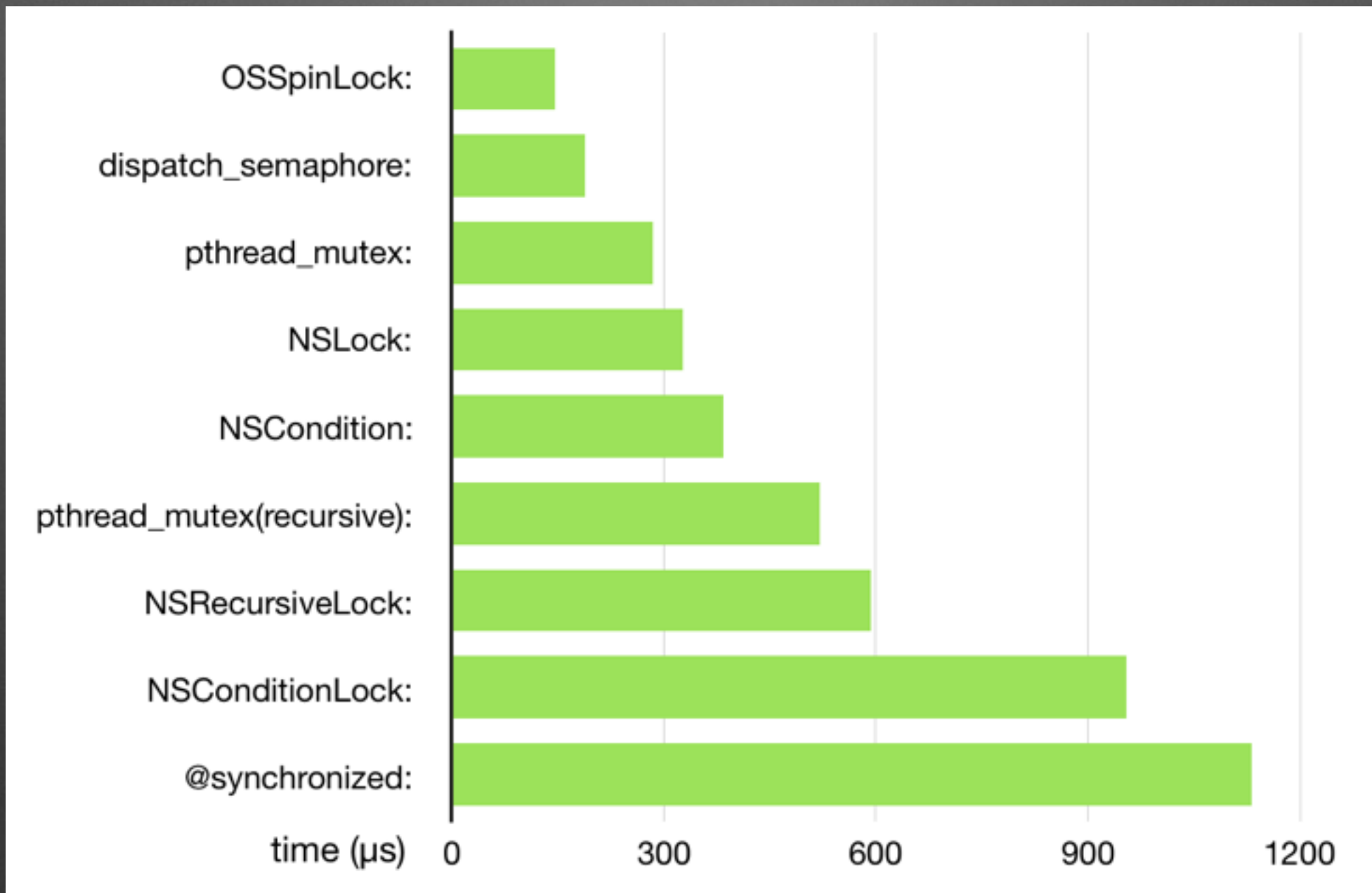
- CPU利用率
- 系统吞吐量
- 周转时间
- 等待时间
- 响应时间

线程的优先级

- 1.用户制定优先级
- 2.根据进入优先级的频繁程度提升或降低优先级
- 3.长时间得不到调用而被提升优先级

锁的类型

- 二元信号量 (Binary Semaphore)
 - 二元信号量是最简单的一种锁，它只有两种状态：占用与非占用。它适合只能被唯一一个线程独占访问的资源。当二元信号量处于非占用状态时，第一个试图获取该二元信号量的线程会获得该锁，并将二元信号量置为占用状态。第二个试图获取该二元信号量的线程将会等待，直到该锁的释放。
- 互斥量 (Mutex)
 - 互斥量与信号量非常类似，区别就在于信号量可以被任意的线程获取并释放，互斥量则要求哪个线程获取了互斥量，哪个线程就要负责释放。
- 临界区 (Critical Section)
 - 与互斥锁类似，但比互斥锁更为严格，临界区的作用范围仅限于本进程，其余线程不可获取该锁。
- 读写锁 (Read-Write Lock)
 - 这里不详细介绍，读写锁三种状态：自由 / 共享 / 独占有两种获取方式：共享和独占
 - 当以共享方式去获取锁时，除了独占状态需等待，其余状态均可获取；
 - 当以独占方式去获取锁时，线程必须等待锁被所有资源释放后，才可以获取；
- 条件变量 (Condition Variable)
 - 该锁类似一个塞子，不同的线程均可以等待相同的条件，但只有满足条件的线程才会被唤醒



**OSSpinLock不安全，（优先级
反转）**

DNS服务器的简单介绍

什么是DNS服务器

- DNS（Domain Name System 的缩写）的作用非常简单，就是根据域名查出IP地址。你可以把它想象成一本巨大的电话本。
- 举例来说，如果你要访问域名 `math.stackexchange.com`，首先要通过DNS查出它的IP地址是 `151.101.129.69`

网络的分层概念

- 应用层 (ftp, 邮件, 网页)
- 传输层 (port; 处理端口到端口; udp / tcp)
- 网络层 (ip协议 / 子网掩码; 处理主机到主机)
- 链接层 (同一个局域网下, 查找; 网卡; 以太网协议)
- 实体层 (实际的物理链接)

URL编码

`http://root:Password1@example.com:8042/over/there/index.dtb?type=animal&name=narwhal#nose`

SCHEME

USERNAME

PASSWORD

HOSTNAME

PORT

PATH

EXTENSION

QUERY

FRAGMENT

相关博客地址：

1.<http://www.cocoachina.com/ios/20160129/15170.html>

2.<https://blog.ibireme.com/2016/01/16/>

[spinlock is unsafe in ios/](#)

3.<http://www.ruanyifeng.com/blog/2016/06/dns.html>

4.<http://www.ruanyifeng.com/blog/2012/05/>

[internet_protocol_suite_part_i.html](#)

Thanks