



逻辑教育  
Logic education

# 大师班第21天

和谐学习，不急不躁

LG\_Cooci



### GCD 简介

什么是GCD?

全称是 Grand Central Dispatch

纯 C 语言，提供了非常多强大的函数

### GCD的优势

GCD 是苹果公司为多核的并行运算提出的解决方案

GCD 会自动利用更多的CPU内核（比如双核、四核）

GCD 会自动管理线程的生命周期（创建线程、调度任务、销毁线程）

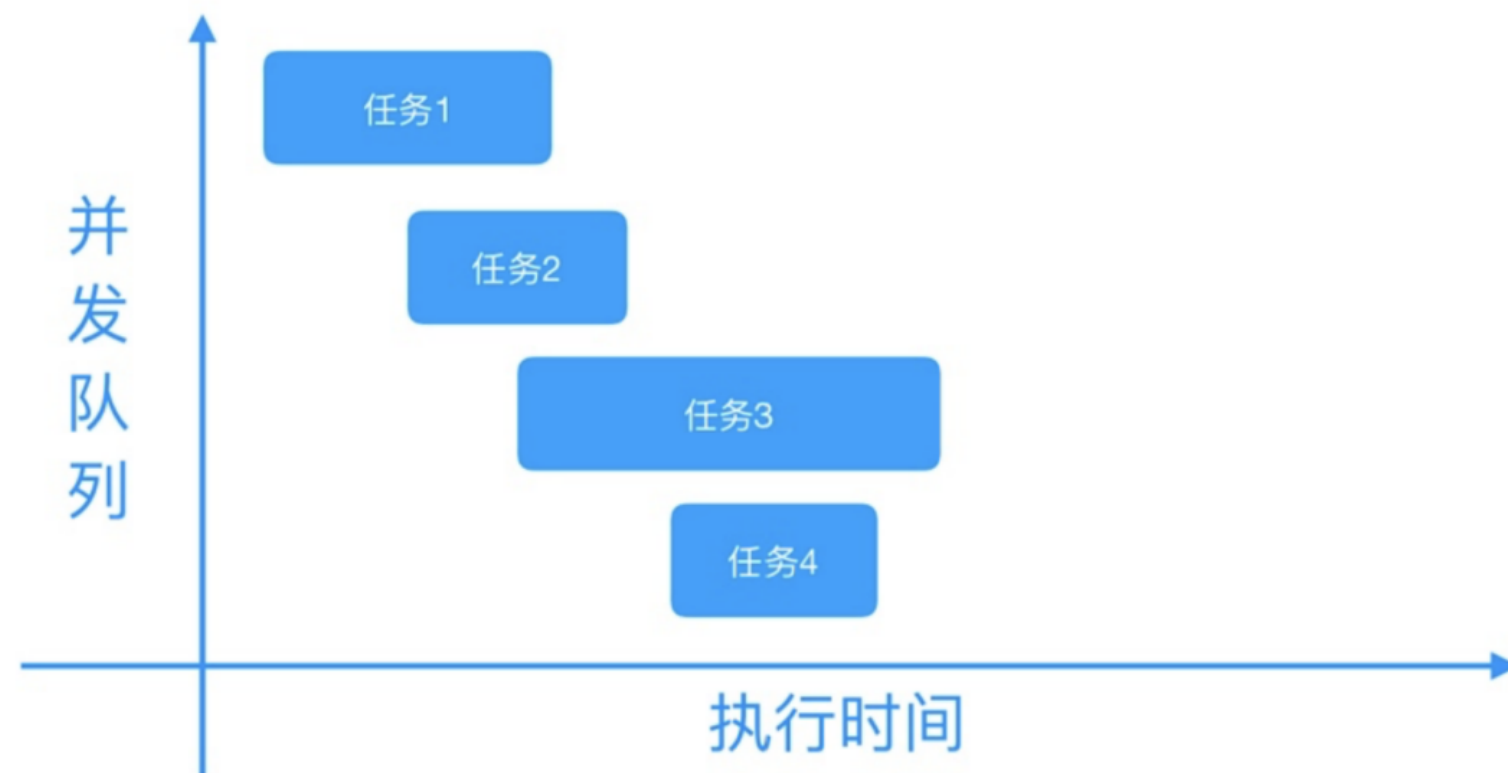
程序员只需要告诉 GCD 想要执行什么任务，不需要编写任何线程管理代码



- \* 任务使用 block 封装
- \* 任务的 block 没有参数也没有返回值
- \* 执行任务的函数
  - \* 异步 `dispatch\_async`
    - \* 不用等待当前语句执行完毕，就可以执行下一条语句
      - \* 会开启线程执行 block 的任务
      - \* 异步是多线程的代名词
  - \* 同步 `dispatch\_sync`
    - \* 必须等待当前语句执行完毕，才会执行下一条语句
    - \* 不会开启线程
    - \* 在当前执行 block 的任务



## 队列



## 函数与队列

### 同步函数串行队列

- 不会开启线程，在当前线程执行任务
- 任务串行执行，任务一个接着一个
- 会产生堵塞

### 同步函数并发队列

- 不会开启线程，在当前线程执行任务
- 任务一个接着一个

### 异步函数串行队列

- 开启线程一条新线程
- 任务一个接着一个

### 异步函数并发队列

- 开启线程，在当前线程执行任务
- 任务异步执行，没有顺序，CPU调度有关



### 主队列

专门用来在主线程上调度任务的串行队列

不会开启线程

如果当前主线程正在有任务执行，那么无论主队列中当前被添加了什么任务，都不会被调度

`dispatch_get_main_queue();`

### 全局并发队列

为了方便程序员的使用，苹果提供了全局队列 `dispatch_get_global_queue(0, 0)`

全局队列是一个并发队列

在使用多线程开发时，如果对队列没有特殊需求，在执行异步任务时，可以直接使用全局队列

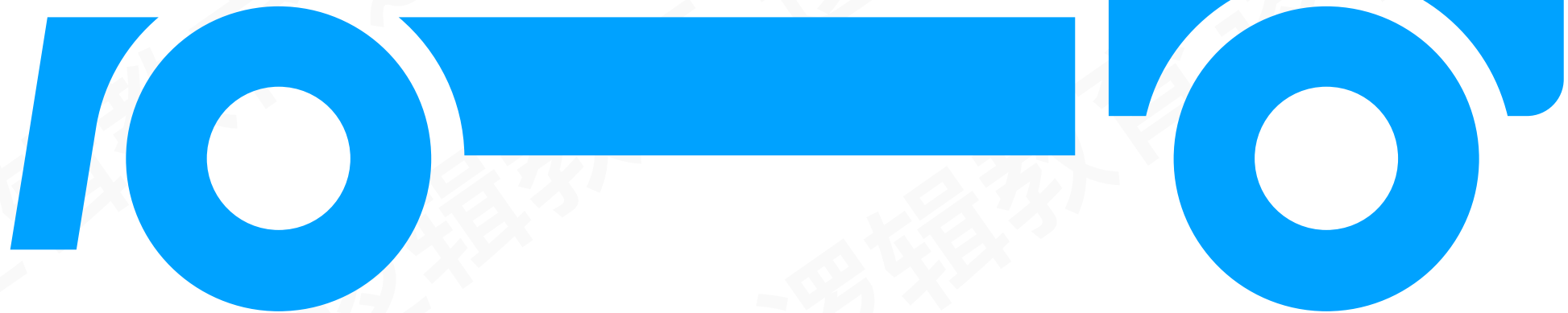


## 死锁现象

主线程因为你同步函数的原因等着先执行任务

主队列等着主线程的任务执行完毕再执行自己的任务

主队列和主线程相互等待会造成死锁





## GCD的作用

单利

同步/异步

dispatchSource

栅栏函数

调度组

延迟

线程通讯

信号量/锁

.....





最直接的作用: 控制任务执行顺序,同步

`dispatch_barrier_async` 前面的任务执行完毕才会来到这里

`dispatch_barrier_sync` 作用相同,但是这个会堵塞线程,影响后面的任务执行

非常重要的一点: 栅栏函数只能控制同一并发队列

### 最直接的作用: 控制任务执行顺序

<code>dispatch_group_create</code>	创建组
<code>dispatch_group_async</code>	进组任务
<code>dispatch_group_notify</code>	进组任务执行完毕通知
<code>dispatch_group_wait</code>	进组任务执行等待时间
<code>dispatch_group_enter</code>	进组
<code>dispatch_group_leave</code>	出组
注意搭配使用	



## 信号量dispatch\_semaphore\_t

dispatch\_semaphore\_create

创建信号量

dispatch\_semaphore\_wait

信号量等待

dispatch\_semaphore\_signal

信号量释放

同步->当锁, 控制GCD最大并发数



- \* 其 CPU 负荷非常小，尽量不占用资源
- \* 联结的优势

在任一线程上调用它的一个函数 `dispatch_source_merge_data` 后，会执行 Dispatch Source 事先定义好的句柄（可以把句柄简单理解为一个 block）  
这个过程叫 Custom event，用户事件。是 dispatch source 支持处理的一种事件

句柄是一种指向指针的指针 它指向的就是一个类或者结构，它和系统有很密切的关系  
HINSTANCE（实例句柄），HBITMAP（位图句柄），HDC（设备表述句柄），HICON（图标句柄）等。这当中还有一个通用的句柄，就是HANDLE



## Dispatch\_Source

dispatch\_source\_create

创建源

dispatch\_source\_set\_event\_handler

设置源事件回调

dispatch\_source\_merge\_data

源事件设置数据

dispatch\_source\_get\_data

获取源事件数据

dispatch\_resume

继续

dispatch\_suspend

挂起



逻辑教育  
Logic education

# *Hello Cooci*

我就是我，颜色不一样的烟火