

RunLoop

概念

- 基于EventLoop, do while循环, 符合条件后退出循环
- 关键: 没有消息处理时候线程休眠, 需要处理消息时立刻唤醒

与线程的关系

一一对应, 使用全局static CFMutableDictionaryRef loopsDic, key为pthread\_t, value 是 CFRunLoopRef。

CFRunLoopRef

- 纯C函数API, 线程安全
- 成员变量
  - CFMutableSetRef \_commonModes;
  - CFMutableSetRef \_commonModelItems;

使用kCFRunLoopCommonModes标记mode为common。  
CFRunLoopAddCommonMode(CFRunLoopRef runloop, CFStringRef modeName);

NSRunLoop

基于CFRunLoopRef封装, 面向对象API, 线程不安全

主要数据结构

CFRunLoopModeRef

- 不对外暴露, 通过CFRunLoopRef接口封装
- 如果需要切换 Mode, 只能退出 Loop, 再重新指定一个 Mode 进入。这样做主要是为了分隔开不同组的 Source/Timer/Observer, 让其互不影响。
- Mode Item, 可以被同时加入多个mode, mode中一个item都没有时, 直接退出。
  - CFMutableSetRef \_sources0;
    - 只包含回调 (函数指针), 不能主动触发事件
    - CFRunLoopWakeUp(runloop),手动唤醒runloop处理事件
    - CFRunLoopSourceSignal(source), 将这个 Source 标记为待处理
  - CFMutableSetRef \_sources1;
    - 包含mach\_port和回调(函数), 可以主动唤醒runloop
  - CFMutableArrayRef \_observers;
    - 监听状态变化
      - kCFRunLoopEntry = (1UL << 0), // 即将进入Loop
      - kCFRunLoopBeforeTimers = (1UL << 1), // 即将处理 Timer
      - kCFRunLoopBeforeSources = (1UL << 2), // 即将处理 Source
      - kCFRunLoopBeforeWaiting = (1UL << 5), // 即将进入休眠
      - kCFRunLoopAfterWaiting = (1UL << 6), // 刚从休眠中唤醒
      - kCFRunLoopExit = (1UL << 7), // 即将退出Loop
  - CFMutableArrayRef \_timers;
    - 包含时间长度和回调, 进入runloop时, 会注册对应的时间点, 到点执行回调
    - 基于时间的触发器, 可以与NSTimer混用

主要流程

- 1、CFRunLoopRunSpecific, 入参runloop, modeName、stopAfterHandle
- 2、根据modeName获取mode, check如果modelItem为空(即没有source、timer、observer), 直接退出runloop
- 3、触发source0回调(非port)
  - 3.1、如果有source1事件, go to 5,跳过线程休眠、执行source1
  - 3.2、如果没有, 顺序执行4.
- 4、线程将进入休眠
  - 调用mach\_msg 等待接受 mach\_port 的消息。
  - 可被以下事件唤醒:
    - 4.1 、一个基于 port 的Source 的事件。(source1)
    - 4.2、一个 Timer 到时间了
    - 4.3、RunLoop 自身的超时时间到了
    - 4.4、被其他什么调用者手动唤醒 (CFRunLoopWakeUp)
- 5、线程被唤醒, 执行唤醒者(4.1 ~ 4.4)事件
- 6、检查runloop是否需要退出, 如果不需要, 返回3, 继续执行

runloop退出条件

- 1、外部传入参数、提示runloop执行完参数就退出 (stopAfterHandle和source0共同决定)
- 2、超出入参标记的runloop时间
- 3、手动强制停止
- 4、modelItem为空

底层实现

- runloop的核心: 调用mach\_msg进入线程休眠, 并被mach\_port唤醒
- mach的概念