

runloop <https://blog.ibireme.com/2015/05/18/runloop/>

RunLoop 对外的接口

在 CoreFoundation 里面关于 RunLoop 有5个类:

CFRunLoopRef

CFRunLoopModeRef

CFRunLoopSourceRef

CFRunLoopTimerRef

CFRunLoopObserverRef

一个 RunLoop 包含若干个 Mode，每个 Mode 又包含若干个

Source/Timer/Observer。**每次调用 RunLoop 的主函数时，只能指定其中一个 Mode**，这个 Mode 被称作 CurrentMode。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

1.CFRunLoopSourceRef 是事件产生的地方。Source 有两个版本:

Source0 和 Source1。

- Source0 只包含了一个回调（函数指针），它并不能主动触发事件。使用时，你需要先调用 `CFRunLoopSourceSignal(source)`，将这个 Source 标记为待处理，然后手动调用 `CFRunLoopWakeUp(runloop)` 来唤醒 RunLoop，让其处理这个事件。
- Source1 包含了一个 `mach_port` 和一个回调（函数指针），被用于通过内核和其他线程相互发送消息。这种 Source 能主动唤醒 RunLoop 的线程，其原理在下面会讲到。

2.CFRunLoopTimerRef 是基于时间的触发器，它和 `NSTimer` 是 toll-free bridged 的，可以混用。其包含一个时间长度和一个回调（函数指

针)。当其加入到 RunLoop 时, RunLoop 会注册对应的时间点, 当时间点到时, RunLoop 会被唤醒以执行那个回调。

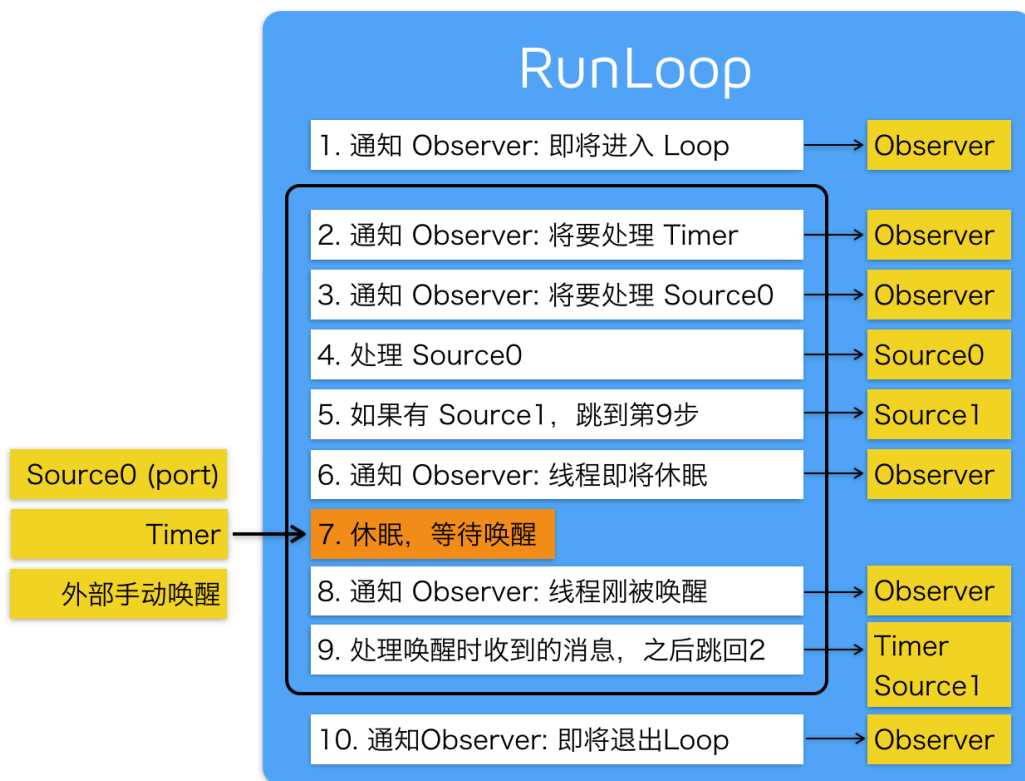
3.CFRunLoopObserverRef 是观察者, 每个 Observer 都包含了一个回调 (函数指针), 当 RunLoop 的状态发生变化时, 观察者就能通过回调接受到这个变化。可以观测的时间点有以下几个:

```
typedef CF_OPTIONS(CFOptionFlags, CFRunLoopActivity) {
    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
}
```

上面的 Source/Timer/Observer 被统称为 **mode item**, 一个 item 可以被同时加入多个 mode。但一个 item 被重复加入同一个 mode 时是不会有效果的。如果一个 mode 中一个 item 都没有, 则 RunLoop 会直接退出, 不进入循环。

应用场景举例: 主线程的 RunLoop 里有两个预置的 Mode:

kCFRunLoopDefaultMode 和 UITrackingRunLoopMode。这两个 Mode 都被标记为"Common"属性。DefaultMode 是 App 平时所处的状态, TrackingRunLoopMode 是追踪 ScrollView 滑动时的状态



可以看到，系统默认注册了5个Mode:

1. `kCFRunLoopDefaultMode`: App的默认 Mode，通常主线程是在这个 Mode 下运行的。
2. `UITrackingRunLoopMode`: 界面跟踪 Mode，用于 `ScrollView` 追踪触摸滑动，保证界面滑动时不受其他 Mode 影响。
3. `UIInitializationRunLoopMode`: 在刚启动 App 时第进入的第一个 Mode，启动完成后就不再使用。
4. `GSEventReceiveRunLoopMode`: 接受系统事件的内部 Mode，通常用不到。
5. `kCFRunLoopCommonModes`: 这是一个占位的 Mode，没有实际作用。