

**iphone程序生命周期**

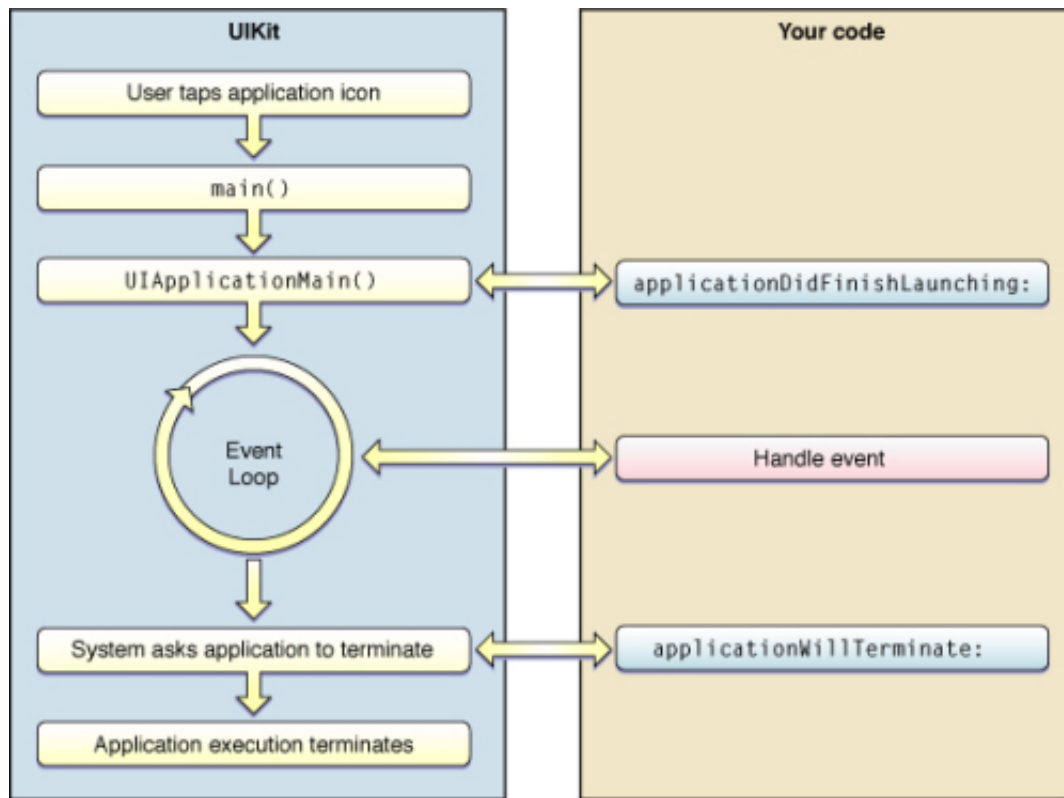
<http://husbandman.diandian.com/post/2012-08-14/40035165582>

## iphone程序生命周期

延参大师曰:绳命,是多么的回晃; 绳命, 是如此的并彩

1. 轻点图标
2. 显示过渡图形
3. 调用main函数
4. 调用UIApplicationMain
5. applicationDidFinishLaunching
6. Event loop
7. applicationWillTerminate
8. End

过程如图



## main

main函数是程序的入口.主要是用来完成两方面的工作:

- 对于释放池的处理
- 调用UIApplicationMain函数

对于释放池的处理

自动引用计数(ARC)

如果采用自动引用计数(ARC),那么代码形式如下

```
@autoreleasepool {  
    return UIApplicationMain(argc, argv, nil, NSStringFromClass([AppDelegate class]));  
}
```

@autoreleasepool是通过编译器自动实现的引用计数,在ARC环境下,retain,release和autorelease是无效的.引用计数由编译器自动实现.

## 手动引用计数(MRC)

如果采用手动引用计数(MRC),那么代码形式会像下面这样

```
NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];  
int result = UIApplicationMain(argc, argv, nil, @"AppDelegate");  
[pool release];  
return result;
```

首先创建一个NSAutoreleasePool的自动释放池,所有的自动释放对象将注册到这个pool中.当执行[pool release]时候,所有注册到pool中的对象先进行释放,然后在释放pool

## UIApplicationMain函数

UIApplicationMain的函数形式如下

```
int UIApplicationMain(int argc, char *argv[], NSString *principalClassName, NSString *delegateClassName);
```

### 参数说明

- principalClassName 是UIApplication类或其子类的名称.如果principalClassName传入nil,则默认采用UIApplication
- delegateClassName 是一个遵循UIApplicationDelegate协议的类的名称.如果传入nil,则通过info.plist确定的主nib文件中的对象来假定是应用程序的委托对象.

探讨

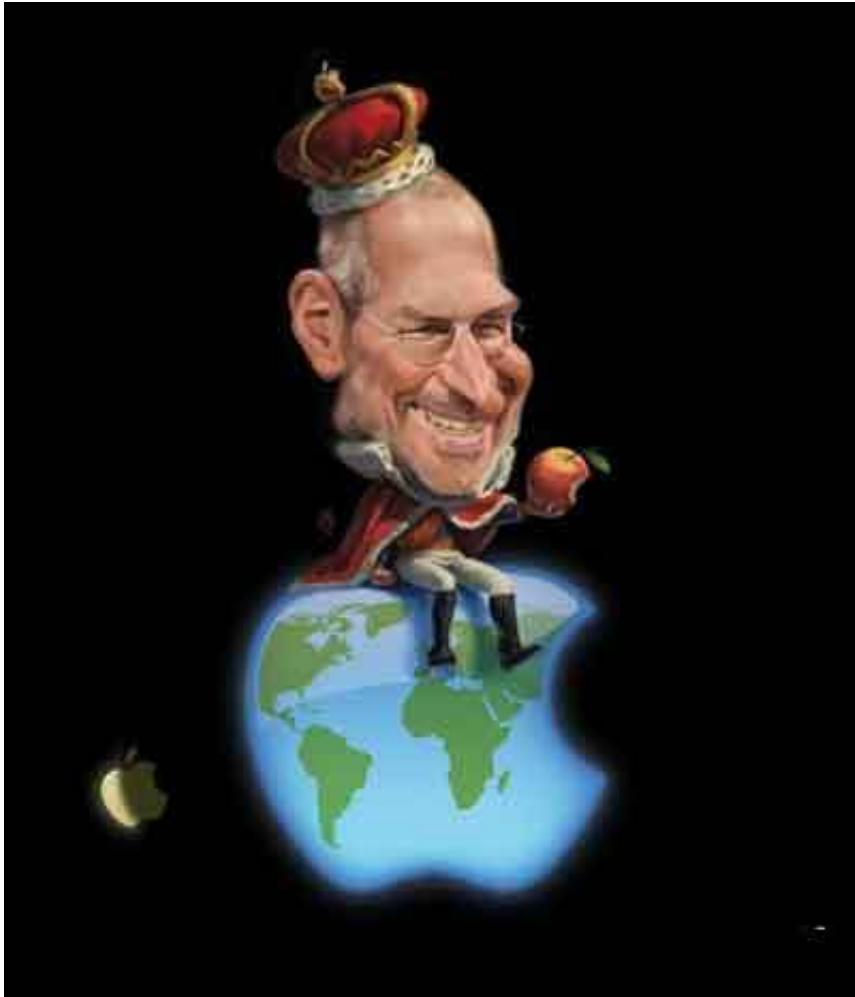
在UIApplicationMain内部,实现了类似的代码

```
UIApplication *app = [[UIApplication alloc] init];  
UIApplicationDelegate *delegate = [[UIApplicationDelegate alloc] init];  
app.delegate = delegate;
```

UIApplication的实例,作为用户和程序交互的应答者.而UIApplication的代理则用来处理相应的交互.委托的角色,类似于法人代表,CEO,领导者,国家元首,的角色.



奥巴马之于美国



## 乔布斯之于苹果

以美国为例,国家决策权本应属于全体美国公民,但由于太过分散,不利于集中表达,国家紧急情况由国家元首总统代为行使.

你在UIApplication的委托中写代码,类似于作为当前程序王国的总统(伟大的社会主义国家实质的一把手),决定整个程序的运行状况.

**applicationDidFinishLaunching**

这个消息类似于jQuery的ready事件,告诉你一切准备就绪,你可以开始了.也可以形象的比喻成,好比总统宣誓就任,你可以决定提议国务卿和内阁组成,以及美联储主席等要职的安排等.

## Event loop

常言道: "兵来将挡, 水来土掩",事到其间,道在人为.少不得你我打点礼物,早差人上东京,央及老爷那里去(<<金瓶梅词话>> 第四十八回)

好一个道在人为,一个应用程序的绳命,与一个人绳命的是一样样的.

Opportunities favor the prepared mind(机遇总偏袒有准备的人)

所以,怎样应对绳命中的大事,是整个程序能否回晃,能否井彩的关键.

作为当前程序王国的总统或国王,你可以对各个事件置之不理,亦可以事必躬亲,亦可以排兵布将,定夺人事.

## applicationWillTerminate

这个预示程序就要结束.

映射为当前程序王国的总统,就是你的任期到了,要做交接处理了.

## 示例

下面我们模拟一个声明周期.模拟的顺序如下:

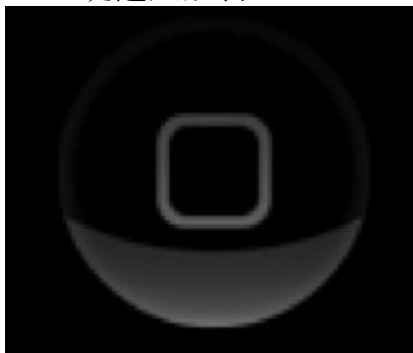
1. 启动点点



## 2. 发生事件

- application:didFinishLaunchingWithOptions:
- applicationDidBecomeActive:

## 3. home键进入后台



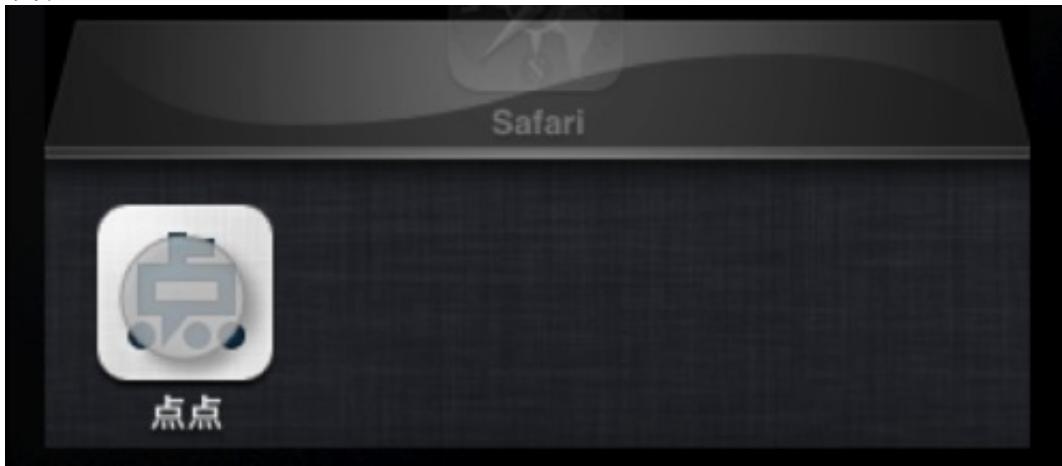
## 4. 发生事件

- applicationWillResignActive:
- applicationDidEnterBackground:

## 5. 进入前台



或者



## 6. 发生事件

- `applicationWillEnterForeground:`
- `applicationDidBecomeActive:`



## 提示:

在支持后台执行的程序中, `applicationDidEnterBackground:` 替代了 `applicationWillTerminate:` 作为程序快照保存的事件.

## 参考

[iPhone 应用程序编程指南](#)