面试题01答案:

@synchronized(参数)

1.简述OC中内存管理机制。与retain配对使用的方法是dealloc还是release，为什么？需要与alloc配对使用的方法是dealloc还是release，为什么？readwrite，readonly，assign，retain，copy，nonatomic 、atomic、strong、weak属性的作用？

(MRC或ARC)

OC采用引用计数的内存管理机制。当一个对象被alloc时，对象的引用计数变为1，每次向对象发送retain或copy消息时，对象的引用计数加1，每次向对象发送release消息时，对象的引用计数减1。当引用计数为0时，对象占用的内存被自动释放。

与retain对应的是release,因为retain之后，对象的引用计数加1，需要release使对象的引用计数减1。

与alloc对应的是dealloc,因为alloc后为对象分配内存，引用计数加1，dealloc回收对象占用的内存。

readwrite: 该属性声明的对象，可读可写,让系统生成setter 和getter

readonly: 只能读，而不能写入, 系统只生成getter

assign:引用计数不会加1，适用于基本类型。常用于指向储存于静态区的变量以及delegate, 直接赋值。

retain:引用计数加1，获得对象的所有权。赋值

copy:复制一个对象，并且储存在不同的内存区。原对象的引用计数不变。遵循NSCoding协议的可以深拷贝。

Unsafe\_unretained:弱引用

nonatomic:非原子性，不保证多线程下保护安全

atomic:原子性，保证多线程下安全。有一个属性需要多个线程同时访问一个变量，使用atomic能自动加锁解锁。

strong:用于ARC模式下，表明对象属于强引用，类似于retain,但是比retain更安全，只要对象存在拥有者，就可以引用，不会被释放，只要对象失去了拥有者，立即被抛弃。不必再将已经释放的对象的指针指向空，避免了野指针。

weak:类似与assign,弱引用，引用计数不会增加，释放指针自动置为nil;

2. 类变量的@protected ,@private,@public,@package，声明各有什么含义？

@protected:受保护类型的变量，只能是子类或者是该类可以引用变量

@private:私有类型的变量，只能是该类可以引用变量。

@public:共有类型的变量，所有类都可以引用变量。

@package:表示框架内访问，外部无法访问

3.线程是什么？进程是什么？二者有什么区别和联系

进程：进程是程序的一次执行活动，是操作系统中可以并行工作的基本单元，一个应用程序有一个进程，一个进程至少有一个线程。进程是有一定独立功能的程序关于某个数据集合上的一次运行活动，进程是系统进行资源分配和调度的一个独立单位。

线程：线程是进程的一个实体，是CPU调度和分配的基本单位，它是比进程更小的能独立运行的基本单位。线程不拥有系统资源，只拥有一点在运行中必不可少的资源。

联系：一个线程可以创建和撤销另一个线程，同一个进程中的多个线程之间可以并发执行。线程是进程的基本组成单位。

区别：它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其他进程产生影响，而线程只是一个进程中不同的执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。

4. 谈谈你对多线程开发的理解？ios中有几种实现多线程的方法？

在一个进程中有多个线程共同存在，每个线程执行各自的任务，相互之间不会干扰，一个线程可以创建或者撤销其他的线程。让一个进程可以同时执行很多任务，这就减少了等待时间，提高了进程的运行效率。但是多线程会耗费资源，并且由于一个线程死掉会造成整个进程死掉，多线程容易造成安全性的问题。多线程的目的是为了防止长时间执行的任务卡死主线程的情况。多线程可以创建多个子线程去完成某项数据处理或者网络数据请求的任务。当该任务完成后，使用主线程进行界面的刷新或者其他操作，以此解决系统卡死，用户交互差的问题。

NSThread:[NSThred detachNewThreadSelector:@selector() toTarget: withObject:]

NSObject分类中的方法：[self performSelectorInBackground:@selector() withObject:nil];

NSOprationQueue (操作队列)

GCD:Grand Central Dispatch

5.线程同步和异步的区别？IOS中如何实现多线程的同步？

线程同步是多个线程同时访问同一资源，等待资源访问结束，浪费时间，效率低，串行执行任务。线程异步：访问资源时在空闲等待时同时访问其他资源，实现多线程机制，并行执行任务。

原子操作，加锁。

使用GCD的主队列

使用NSOperationQueue,把最大线程数设为1.

6.假设有一个字符串aabcad，请写一段程序，去掉字符串中不相邻的重复字符串，即上述字符串处理之后的输出结果为：aabcd。

方法一：思想：将字符串的每一个字符放到集合set中，如果集合set的count加1了，不做任何操作；如果set的count不变，将所在字符和前面的进行比较，如果相同，不做任何操作；如果不同，记录下字符所在数组的下标。便利完成之后，将存储下标的数组中的元素从后面进行遍历，删除字符。（从后面进行删除的原因是，如果从前面删除，会破坏字符数组原有的顺序，导致后面按照数组的下标删除时造成误删）

NSMutableString \*str = [NSMutableString stringWithString:@"aabcbacd"];

NSMutableSet \*set = [NSMutableSet setWithCapacity:10];

NSMutableArray \*array = [NSMutableArray arrayWithCapacity:10];

for (int i = 0 ; i < str.length ; i++)

{

char a = [str characterAtIndex:i];

NSInteger num = set.count;

[set addObject:[NSString stringWithFormat:@"%c", a]];

if ((set.count - num) == 1)

{

}else

{

char c = [str characterAtIndex:i - 1];

if (a == c)

{

}else

{

[array addObject:[NSString stringWithFormat:@"%d", i]];

}

}

}

for (int i = array.count - 1; i >= 0; i--)

{

NSInteger num = [[array objectAtIndex:i] integerValue];

[str deleteCharactersInRange:NSMakeRange(num, 1)];

}

NSLog(@"%@", str);

方法二：

NSMutableString \*str = [NSMutableString stringWithString:@"aabcad"];

NSMutableArray \*indexArray = [[NSMutableArray alloc]initWithCapacity:10];

for (int i = 0; i < str.length - 1; i++)

{

char a = [str characterAtIndex:i];

for (int j = i + 1; j < str.length; j++)

{

char c = [str characterAtIndex:j];

if (a == c)

{

if ((i + 1) == j)

{

}else

{

[indexArray addObject:[NSString stringWithFormat:@"%d", j]];

}

}

}

}

NSSet \*set = [NSSet setWithArray:indexArray];

NSArray \*array = [set allObjects];

for (int i = array.count - 1; i >= 0; i--)

{

NSInteger num = [[array objectAtIndex:i] integerValue];

[str deleteCharactersInRange:NSMakeRange(num, 1)];

}

NSLog(@"%@", str);

7.获取一台设备唯一标识的方法有哪些

ios7之前获取设备唯一标识的方法：

1.UDID(Unique Device Identifier,),它是苹果ios设备的唯一识别码，它由40个字符的字母和数字组成。

2.UUID(Universally Unique Identifier),通用唯一识别码。

3.MAC Address

4.OPEN UDID

5.广告标示符(IDFA-identifierForIdentifier)

6.Vindor标示符(IDFV-identifierForVendor) Vendor是CFBundleIdentifier的前两部分。来自同一个运营商的应用运行在同一设备上，此属性的值是相同的；不同的运营商应用运行在同一设备上值不同。

Ios7 :keychain(钥匙串)通常情况下，ios系统用，NSUSerDefaults存储数据信息，但是对于一些私密信息，比如密码，证书等等，就需要使用更为安全的keychain了。

如何使用keychain 我们需要导入Security.framework, 我们使用已经封装好的工具类KeychainItemWrapper来对keychain进行操作。

KeychainItemWrapper是apple官方例子“GenericKeychain”里一个访问keychain常用操作的封装类，在官网上下载了GenericKeychain项目后，

只需要把“KeychainItemWrapper.h”和“KeychainItemWrapper.m”拷贝到我们项目，并导入Security.framework 。KeychainItemWrapper的用法：

初始化一个保存用户帐号的KeychainItemWrapper

KeychainItemWrapper \*wrapper = [[KeychainItemWrapper alloc] initWithIdentifier:@"Account Number" accessGroup:@"YOUR\_APP\_ID\_HERE.com.yourcompany.AppIdentifier"];

//保存数据

[wrapper setObject:@"<帐号>" forKey:(id)kSecAttrAccount];

[wrapper setObject:@"<帐号密码>" forKey:(id)kSecValueData];

//从keychain里取出帐号密码

NSString \*password = [wrapper objectForKey:(id)kSecValueData];

//清空设置

[wrapper resetKeychainItem];

其中方法“- (void)setObject:(id)inObject forKey:(id)key;”里参数“forKey”的值应该是Security.framework 里头文件“SecItem.h”里定义好的key，用其他字符串做key程序会出错

8.iOS类是否可以多继承？如果没有，那可以用其他方法实现吗？简述实现过程

ios不允许多继承，多继承可以通过协议实现。

9.堆和栈的区别？

栈区：由编译器自动释放，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈。

堆区：一般由程序员分配释放，若程序员不释放，程序结束时可能由os回收。与数据结构中的堆是两码事，分配方式类似于链表。

常量区：程序结束后由系统释放

代码区：存放函数体的二进制代码

全局区（静态区）：程序结束后由系统释放。初始化的全局变量和静态变量在一块区域，未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。

例如:NSString \*str = [[NSString alloc]init];

左边的str指针在栈中，是系统自动分配的，右侧是在堆中申请内存，存放信息。Str指向的位置是右侧空间的地址。

栈是系统调用，使用方便快捷，但是自由度较小。堆是程序员主动申请，灵活度高，自由度大。但是需要关注内存管理问题

10. iOS本地数据存储都有哪几种方式？

plist(属性列表)

数据库(SQLite)

归档和反归档

文件读写

CoreData

NSUserDefaults

11.iOS动态类型和动态绑定 动态载入， sel类型

动态类型：id是共有的对象类型，可以用来存放任意类型。Id类型可以在代码运行时判断对象的类型，使用id类型可以在运行的时候使用任务数据类型来替换。动态类型可以让程序更加灵活，但是会使数据的统一性降低。我们常用的静态类型，比如NSString等都有着自己的优势，使用静态类型编译器可以完全分析代码的性能，可以预知性更高。

动态绑定：可以在代码运行的时候判断需要调用什么方法。

面试题02答案:

1.写出方法获取ios内存使用情况。

2.深拷贝和浅拷贝的理解？

子类对象的地址是否改变。

浅拷贝：指针拷贝，不会产生新的对象。源对象计数器加1.

深拷贝：内容拷贝。会产生新的对象。新对象的引用计数器加1，源计数器的引用计数不变。

不可变字符串调用copy实现浅拷贝

不可变字符串调用mutableCopy实现深拷贝

可变字符串调用copy实现深拷贝

可变字符串调用mutableCopy实现深拷贝

拷贝自定义对象需要实现<NSCopying, NSMutableCopying>

3.怎样实现一个singleton的类

4.什么是安全释放？

先释放，再将对象的指针置为nil

5. RunLoop是什么？

线程中的运行循环。首先循环体的开始需要检测是否有处理的事件，如果有则去处理，如果没有则进入睡眠以节省CPU的时间。每个线程包括程序的主线程都有与之对应的runLoop对象，主线程的runloop是默认启动的,子线程如果没有打开runloop，在子线程中设置NSTimer,只会执行一次，因为子线程的runLoop的没有打开，在子线程的事件监听包括通知，按钮点击，也不执行。

6. 什么是序列化和反序列化，可以用来做什么？如何在OC中实现复杂对象的存储？

序列化：把对象写到二进制流中去。

反序列化：从二进制流读取对象

应用归档和反归档可以实现数据的持久化。

遵循NSCoding协议，来实现复杂对象的存储，实现该协议后就可以对其进行打包或者解包，然后转化为NSData，方便传输

7.写一个标准宏MIN，这个宏输入两个参数并返回较小的一个？

#define MIN(x, y) (((x) >= (y)) :( y )? (x))

8.iphone os有没有垃圾回收机制？简单阐述一下OC内存管理

没有垃圾回收机制。内存管理机制同面试题01.1

补充：OC的内存管理

1:对你自己拥有的对象负责.你只能释放你拥有的对象.

2:凡是你通过retain,alloc,copy等手段获得了所有权的对象,都必须在你不再使用它的时候,,由你来调用release,autorelease等手段来释放对它的所有权.

3:在一定的代码段内,对同一个对象所作的copy,alloc和retain的次数应该与autorelease及release的次数相等;

4:可以在dealloc的方法中释放你所占有的实例变量.

5:对于便利构造器和访问器来说,你没有通过上述的手段获得对象的所有权,因此在这些情况下你无须对获得的对象进行额外的释放操作.

6:autorelease只不过意味着预定延迟发送一条release信息.当前的引用计数并没有变.

9.简述应用程序按Home键进入后台时的生命周期，以及从后台回到前台时的生命周期？

启动程序  
lifeCycle[40428:11303] willFinishLaunchingWithOptions  
lifeCycle[40428:11303] didFinishLaunchingWithOptions  
lifeCycle[40428:11303] applicationDidBecomeActive  
按下home键  
lifeCycle[40428:11303] applicationWillResignActive  
lifeCycle[40428:11303] applicationDidEnterBackground  
双击home键，再打开程序  
lifeCycle[40428:11303] applicationWillEnterForeground  
lifeCycle[40428:11303] applicationDidBecomeActive

10. ViewController 的 alloc，loadView, viewDidLoad,viewWillAppear,viewDidUnload,dealloc、init分别是在什么时候调用的？在自定义ViewController的时候这几个函数里面应该做什么工作？

alloc申请内存空间的时候调用。创建的时候

init 视图初始化的时候，赋一些程序的初始值。

LoadView 加载视图的时候调用，self.view != nil的时候，执行loadview,加载一些自己的视图

viewDidLoad:视图已经加载的时候调用（添加通知，）

viewWillAppear:视图将要出现的时候

viewDidUnload:视图已经卸载（视图释放掉，移除通知）

dealloc:视图已经销毁的时候，自己拥有的实例变量释放掉

面试题03答案：

1.描述应用程序的启动顺序

先执行main函数，main函数会调用UIApplicationMain函数, 为应用程序创建一个window，设置一些window的属性，设置UIWindow的rootViewController属性，展示window;

2.为什么很多内置类如UITableViewControl的delegate属性都是assign而不是retain？请举例说明

会引起循环引用。不需要对delegate方法的对象有拥有权

我们只是想得到实现了它delegate方法的对象，然后拿到这个对象的指针就可以了，我们不期望去改变或者做别的什么操作，所以我们只要用assign拿到他的指针就可以了。而用retain的话，计数器加1。我们在别的地方期望释放掉delegate这个对象，然后通过一些判断他是否已经释放掉，做一些操作。但是实际上他的retainCount还是1，没有释放掉，两者相互持有，retaincount永远不会变成0，dealloc不会执行，两者都不会释放。

3. 使用UITableView时候必须要实现的几种方法？

-（NSInteger）tableView:(UITableView\*)tableView numberOfRowsInSection:(NSInteger)section

- (UITbaleViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexP

ath \*)indexPath;

4.写一个便利构造器

-(id)initWithName:(NSString \*)name Sex:(NSString \*)sex Age:(NSInteger)age

{

self = [super init];

if(self)

{

\_name = name;

\_sex = sex;

\_age = age;

}

return self;

}

+ (Person **)personWithName:(NSString**\*)name Sex:(NSString \*)sex Age:(NSInteger)age

{

Person \*person = [[Person alloc ] initWithName:name Sex:sex Age:age];

return [person autorelease] ; // mrc

}

5.UIImage初始化一张图片有几种方法？简述各自的优缺点

imageNamed:检查系统缓存中是否有该名字的image，如果有的话，直接返回；如果没有，则先加载图片到缓存，然后返回。

NSURL \*url = [NSURL urlWithString]

imageWithData：

int

initWithContentsOfFile：系统不会检查系统缓存，而直接从文件系统中加载image并返回

imageWithCGImage：scale：orientation：

当scale = 1 的时候图像为原始大小 orientation ，制定绘制图像的方向

6.回答person的retainCount值，并解释为什么

Person \* per = [[Person alloc] init];

self.person = per;

person的属性为assign时retainCount 为1；为retain时retainCount为2

7.这段代码有什么问题吗:

@implementation Person

- (void)setAge:(int)newAge {

self.age = newAge;

}

@end

8.

for这段代码有什么问题,如何修改

for (int i = 0; i < someLargeNumber; i++) {

NSString \*string = @”Abc”;

string = [string lowercaseString];

string = [string stringByAppendingString:@"xyz"];

NSLog(@“%@”, string);

}

修改后：

for(int i = 0:i < someLargeNumber:i++)

@autoreleasePool{

{

NSString \*string = @”Abc”;

string = [string lowercaseString];

string = [string stringByApprnendingString:@”xyz”];

NSLOog(@”%@”, string);

}

}

内存泄露，方法自带的自动释放池也来不及释放，

在for循环中，自己添加内存池，产生一个释放一个，添加一个自动释放池

9.

面试题04：

1.简述你对UIView、UIWindow和CALayer的理解

UIView的直接父类是UIResponder, UIResponder继承与NSObject. 可以响应用户事件，构建用户的界面，UIView侧重于对显示内容的管理，CALayer 侧重于对内容的绘制，对内容进行动画处理依赖与UIView来进行显示。UIView是CALyer的高层封装。UIView和CALayer是相互依赖的关系。UIView依赖Calayer提供的内容，CALayer依赖uivew提供的容器来显示绘制的内容。归根到底CALayer是这一切的基础，如果没有CALayer，UIView自身也不会存在，UIView是一个特殊的CALayer实现，添加了响应事件的能力

UIWindow:应用程序的窗口，通常跟屏幕一样大，呈现程序的全部内容。

2.写一个完整的代理，包括声明，实现

@protocol MarrayDelegate

@required

- (void)create;

@optional

-(void)care;

@end

@interface Girl :NSObject

@property (nonatomic, assign)id<MarrayDelegate> delegate;

@end

@interface Boy :NSObject<MarrayDelegate>

@end

3. 分析json、xml的区别？json、xml解析方式的底层是如何处理的

DOM:所有的数据读取完再解析

SAX:读入一句解析一句，读入的过程中就解析。

XML:扩展标记语言，是一种允许用户对自己的标记语言进行定义的源语言，格式统一，跨平台和语言，早已成为业界的公认，相对于json格式是一种重量级的数据交换方式。XML提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据。XML文件庞大，文件格式复杂，传输占带宽，客户端不同浏览器之间的解析XML的方式不一致，需要重复编写很多代码，服务器端和客户端需要花费大量代码来解析xml,导致服务器端和客户端代码变得异常复杂且不易维护。

json:一种轻量级的数据交换格式，具有良好的可读和便于快速编写的特性。可在不同平台之间进行数据交换。json采用兼容性很高的、完全独立于语言文本格式，同时具备类似于c语言的习惯。数据格式比较简单，易于读写，格式都是压缩的，占用带宽小；易于解析；json数据能直接为服务器端代码使用，大大简化了服务器端和客户端的代码开发量，且完成任务不变，并且易于维护。json格式没有XMl格式推广的深入人心和广泛，没有XML那么通用性。json格式目前在webservice中推广还属于初级阶段。

xml 相对于json的重量级体现在解析上。

4.ViewController 的 didReceiveMemoryWarning 是在什么时候被调用的默认的操作是什么?

当程序受到内存警告时viewController会调用didReceiveMemoryWarning 这个方法，调用这个方法之后，对view 进行释放并且调用viewDidUnload方法。从ios3.0开始，不需要重载这个函数，把释放内存的代码放到viewDidUnload中去。

5.面向对象的三大特征，并作简单的介绍

多态，不同对象对于同一消息不同响应

封装，隐藏内部实现，稳定外部接口

继承，子类可以直接重用父类的成员和方法

6.重写一个NSStrng类型的，retain方式声明name属性的setter和gette方法

- （void）setName:(NSString \*)name

if(\_name != name)

{

[\_name release];

\_name = [name retain];

}

* (NSString \*)name

{

return [[\_name retain] autorelease];

}

7.简述NotificationCenter、KVC、KVO、Delegate？并说明它们之间的区别

通知中心：发布通知，注册观察者，移除观察者。

键值编码：不使用属性，就可以对变量进行赋值。

- (void)setValuesForKeysWithDictionary:(NSDictionary \*)keyedValues;

KVO是基于KVC

键值观察者：监听属性的变化，一旦发现属性值发生变化，触发事件。

最后需要移除观察者；

代理：代理模式，由代理帮委托者完成一些事情。

8. What is lazy loading?

懒加载，又称延迟记载，通常方法是，你有一个UITextField类型的property,简单定义为userNameTextField，但是你不在初始化方法里为其alloc/init,它就是一个指针，不会占用内存。在访问器里判断此property的指针是否为空，若为空，就alloc/init,这时才真正生成这个对象除非这个对象被使用，否则他永远不会真正生成，也就不会占用内存。

UITableViewCell 的imageView属性。

9.什么是Protocol？什么是代理？写一个委托的interface？委托的property声明用什么属性？为什么？

10.分别描述类别（categories）和延展（extensions）是什么？以及两者的区别？继承和类别在实现中有何区别？为什么Category只能为对象添加方法，却不能添加成员变量？

类目是对现有的一个类的扩展，延展是定义私有变量，继承是子类对于父类已经拥有的成员的继承，创建了新的类。类目还是原有的类，只是对于原有的类进行了一定的功能补充（有待补充）

类目问什么不能添加实例变量：因为类的方法中使用的都是静态变量，不能使用实例变量。

面试题05答案

1. Objective-C有私有方法么？私有变量呢？如多没有的话，有没有什么代替的方法？

静态方法和实例方法

http://blog.sina.com.cn/s/blog\_bf9843bf0101f5m9.html

oc中，所有的实例变量默认都是私有的，所有的实例方法默认都是公有的

OC中不存在私有方法，但是可以通过延展的方式声明私有方法

2.#import、#include和@class有什么区别

#include<>:用于对系统文件的引用，编译器会在系统目录下去查找该文件。 #include””:用于对系统自定义的文件的引用，编译器首先会在用户目录下查找，然后去安装目录，最后去系统目录查找。

但是include 要注意重复引用的问题：

classA,classB 都引用了classC,classD若引用classA与classB，就会报重复引用的错误。

#import

功能与#include基本相同，但是它避免了重复引用的问题，所以在oc中我们基本上都是使用import

@class

@class 就是告诉编译器有这个类存在，但是如何实现的不用告诉编译器。若在.m文件用到了这个类，还要在.m文件中import这个类的

不在.h中import的原因

classA引用了clasB,classB又引用了classc,….,class,A,B,C…..的头文件又引用了很多的文件,呢么importA的话，编译器就需要编译大量的文件，编译时间就会增加

难道头文件中都是用@class吗？，当然不是，有时也是需要#import的，那么什么时候用呢？

（1）一般如果有继承关系的用#import,如果B是A的子类，那么在B中声明A时用#import

（2）另外就是如果有循环依赖的关系，

(3)还有自定义代理的时候，如果在头文件中想声明代理的话，协议是自己定义的时候。

3.谈谈你对MVC的理解？为什么要用MVC？在Cocoa中MVC是怎么实现的？你还熟悉其他的OC设计模式或别的设计模式吗？

高内聚，解耦合

MVC模式的目的就是实现web系统的职能分工。Model层实现系统中的业务逻辑。View层用于与用户的交互。Controller层是Model与View之间沟通的桥梁，他可以分派用户的请求并选择恰当的视图用于显示，同时他可以解释用户的输入并将他们映射为模型层可执行的操作。

采用MVC可以实现页面代码和逻辑的分开。

单例模式，KVO模式，KVC模式，delegate模式，消息通知模式，Target/Action模式，工厂模式

4.如监测系统键盘的弹出

[.[NSNotificationCenter defaultCenter] addObserver:self selector:@selector() name:UIKeyBoardWillShowNotification object:nil];

5.举出5个以上你所熟悉的ios sdk库有哪些和第三方库有哪些

Foundation.framework

UIKit.framework

CoreGraphics.framework

MediaPlayer.framework

Security.framework

MediaPlayer.framework

libsqlite3.dylib

QuartzCore.framework

MapKit

CoreLocation

常用第三方类库：

SDWebImage

FMDatabase

AFNetWorking

XMPPFramework

ASIHttprequest

EGO MJ

6.如何将产品进行多语言发布？

7.如何将敏感字变成\*\*\*\*

8. objc中的减号与加号代表什么？

- 号是实例方法 ， +号方法是类方法

9.单例目的是什么，并写出一个？

保证系统中只有一个实例对象。

static Singleton \*singleton = nil;

+(Singleton \*)defaultShared

{

static dispatch\_once\_t \*onceToken;

dispatch\_once(&onceToken,^{

singleton = [[Singleton alloc]init];

});

return singleton;

}

10.说说响应链

响应链的过程：当事件发生的时候，响应链首先发送给第一个响应者（往往是事件发生的视图，也就是用户触摸屏幕的地方）。事件将沿着响应链一直向下传递，直到被接受并作出处理。一般来说，第一响应就是个视图对象或者其子类，当其被触摸后事件就交由它处理，如果它不处理，事件就会被传递给其视图控制器UIViewController(如果存在)，然后是他的父视图对象，依次类推直到顶层视图。接下来会沿着顶层视图到窗口，再到程序的UIApplication对象，如果整个过程都没有响应这个事件，则事件被丢弃，一般情况下，在响应链中只要有对象处理事件，事件就会被传递。

典型的响应路线图：

处理：

响应：

First Responser 🡪 The Window 🡪 The Application 🡪App Delegate🡪根视图

面试题06答案：

1. 请写出代码,用block来取代上例中的protocol,并比较两种方法的优势。实际应用部分？请写出代码，用block取代协议或回调方法

2.你做iphone开发时候，有哪些传值方式，view和view之间是如何传值的

Block传值，代理传值，单例传值，属性传值

3.给定的一个字符串，判断字符串中是否含有png，有就删除它？

字符串替换

- (NSString \*)stringByReplacingOccurrencesOfString:(NSString \*)target withString:(NSString \*)replacement;

4. 对于语句NSString\* testObject = [[NSData alloc] init];testObject 在编译时和运行时分别是什么类型的对象？

编译时是NSString类型的，运行时是NSData类型的。

声明NSString \*testObject 是告诉编译器，testObject是一个指向某个OC对象的指针。因为不管指向的是什么类型的对象，一个指针所占的内存空间都是固定的，所以这里声明成任何类型的对象，最终生成的可执行代码都是没有区别的。这里限定了NSString只不过告诉编译器，请把testObject当做一个NSString 来检查，如果后面调用了非NSString的方法，会产生警告。接着，你创建了一个NSData对象，然后把这个对象所在地内存地址保存在testObject里，那么运行时，testObject指向的内存空间就是一个NSData对象。

5. OC中是所有对象间的交互是如何实现的

对象间的交互是通过消息发送实现的，两种角色，数据接受者，数据发送者。我们通过oc中给我们提供的手段来实现两者间的通讯。

例如：通知中心，NSNotificationcenter;代理协议的方式

6.目标-动作机制

目标动作机制：允许一个对象向另一个对象发送一条消息（即动作），以之作为对某个用户事件的响应，接收到消息的对象则可以对消息进行响应，并针对业务要求做出处理 ，UIButton的处理方法。

7. for(int index = 0; index < largenumber; index ++){  
NSString \*tempStr = @”tempStr”;  
NSLog(tempStr);  
NSNumber \*tempNumber = [NSNumber numberWithInt:2];  
NSLog(tempNumber);  
}  
这段代码有什么问题.？会不会造成内存泄露（多线程）？在内存紧张的设备上做大循环时自动释放池是写在循环内好还是循环外好？为什么？

8.描述上拉加载、下拉刷新的实现机制？

描述上拉加载、下拉刷新的实现机制？

视觉效果和之后的一些操作。<http://www.tuicool.com/articles/RFzy2a>

9.ios应用程序只能为该程序创建的文件系统中读取文件，不可以去其他地方访问，此区域被称为沙盒，所以所有的非代码文件都要保存在此，例如图像，图标，声音，映像，属性列表，文本文件等

默认情况下，每个沙盒有三个文件夹：Documents,Library和tmp,因为应用的沙盒机制，应用只能在几个目录下读写文件

Documents:苹果建议将程序中建立的或在程序中浏览到的文件数据保存在该目录下，iTunes备份和恢复的时候会包括此目录

Library:存储程序的默认设置或其它状态信息；Library/Caches :存放缓存文件，iTunes不会备份此目录，此目录下文件不会在应用退出时删除

tmp:提供一个即时创建临时文件的地方

10. 介绍一下XMPP？有什么优缺点吗

基于XML的点对点的及时通讯协议

XMPP是一种基于[标准通用标记语言](http://baike.baidu.com/view/5286041.htm" \t "_blank)的子集[XML](http://baike.baidu.com/view/63.htm" \t "_blank)的协议，它继承了在XML环境中灵活的发展性。因此，基于XMPP的应用具有超强的可扩展性。经过扩展以后的XMPP可以通过发送扩展的信息来处理用户的需求，以及在XMPP的顶端建立如内容发布系统和基于地址的服务等应用程序。而且，XMPP包含了针对服务器端的[软件](http://baike.baidu.com/view/37.htm" \t "_blank)协议，使之能与另一个进行通话，这使得开发者更容易建立客户[应用程序](http://baike.baidu.com/view/330120.htm" \t "_blank)或给一个配好系统添加功能、

优点：

开源、跨平台、良好的安全性、扩展性。

缺点：只能传输文本。

XMPP协议是自由、开放、公开的，并且易于了解。而且在客户端、服务器、组件、源码库等方面，都已经各自有多种实现。

互联网工程工作小组（IETF）已经将Jabber的核心XML流协议以XMPP之名，正式列为认可的实时通信及Presence技术。而XMPP的技术规格已被定义在RFC 3920及RFC 3921。任何IM供应商在遵循XMPP协议下，都可与Google Talk实现连接。

第一个Jabber（现在XMPP）技术是Jeremie Miller在1998年开发的，现在已经相当稳定；数以百计的开发者为XMPP技术而努力。今日的互联网上有数以万计的XMPP服务器运作著，并有数以百万计的人们使用XMPP实时传讯软件。

XMPP网络的架构和电子邮件十分相像；XMPP核心协议通信方式是先创建一个stream，XMPP以TCP传递XML数据流，没有中央主服务器。任何人都可以运行自己的XMPP服务器，使个人及组织能够掌控他们的实时传讯体验。

任何XMPP协议的服务器可以独立于公众XMPP网络（例如在企业内部网络中），而使用SASL及TLS等技术的可靠安全性，已内置于核心XMPP技术规格中。

XML命名空间的威力可使任何人在核心协议的基础上建造定制化的功能；为了维持通透性，常见的扩展有XMPP标准基金会。

XMPP除了可用在实时通信的应用程序，还能用在网络管理、内容供稿、协同工具、文件共享、游戏、远程系统监控等。

用XMPP协议来建造及布署实时应用程序及服务的公司及开放源代码计划分布在各种领域；用XMPP技术开发软件，资源及支持的来源是多样的，使得使你不会陷于被“绑架”的困境。

缺点

随着通常超过70%的XMPP协议的服务器的数据流量的存在和近60%的被重复转发，XMPP协议目前拥有一个大型架空中存在的数据提供给多个收件人。新的议定书正在研究，以减轻这一问题。

XMPP协议的方式被编码为一个单一的长的XML文件，因此无法提供修改二进制数据。因此， 文件传输协议一样使用外部的HTTP。如果不可避免，XMPP协议还提供了带编码的文件传输的所有数据使用的Base64。至于其他二进制数据加密会话（encrypted conversations）或图形图标（graphic icons）以嵌入式使用相同的方法。

11.谈谈对性能优化的看法，如何做？

封装：

内存的管理：

代码的重用：减少代码的冗余

网络判断的优化：

适合的数据存储方式：

良好的编程习惯和程序设计，算法的改进

用户的体验优化：

http://blog.csdn.net/yangxt/article/details/8173412

面试题07答案 ：

1.应用程序如何省电？

设备一定时间内没有检测到触摸事件，系统最初会使屏幕变暗，并最终关闭屏幕。大多数开发者都应该让这个定时器打开，但是，游戏和不使用触摸输入的程序开发者可以禁用这个定时器，使屏幕在应用程序运行时不会变暗。将共享的UIApplication对象的idleTimerDisabled属性设置为YES，就可以禁用自动休眠定时器。

比较费电的功能：定位，网络连接，信号的搜索。及时关闭定位；不要频繁的请求网络，做本地存储，让用户主动的更新数据；提升程序的算法，提高代码的质量；蓝牙，需要才连接，用完及时断开；界面的渲染，尽量提升效率，减少渲染次数。

2.写一个递归方法：计算N的阶乘，然后将计算结果进行存储。以便应用退出后下次启动时直接获取该值。

long func(int num)

{

if (num == 0 || num == 1)

{

return 1;

}else

{

return num \* func((num - 1));

}

}

3. NSArray和NSMutableArray的区别，多线程操作哪个更安全？

NSArray创建之后会在堆中建立一块区域,然后指针mo指向这块区域，但是这块区域的大小是已经确定不能更改的了。但是我们还是能够看到有setobject的方法，可以向里面添加数值。这是因为会在堆中新建一个内存区域B,把原来堆内存A中的数据复制到这个堆区域B里面中，然后在这个新的区域B里面添加你setobject进去的值，并把栈中的指针对象mo指向b,这样就是说原来A的堆内存值没有发生变化，但是你的指针mo在指向新的内存区域B之后指针对应的值已经发生变化。因为B中有A的复制内容，还有最后你新添加进去setobject的值。源数组是否发生了变化，NSArray 更安全，不涉及到数变化，只是读取。

4.当前有一个数组，里面有若干个重复的数据，如何去除重复的数据？

方法一: NSSet \*set = [NSSet setWithArray:array];

NSArray \*newArray = [set allObjects];

方法二：for循环

for (int i = 0; i < array.count; i++)

{

for (int j = i + 1; j < array.count; j++)

{

if ([[array objectAtIndex:i] isEqualToString:[array objectAtIndex:j]])

{

[array removeObjectAtIndex:j];

}

}

}

方法三:字典的覆盖性

NSDictionary\*dic=[NSDictionary dictionaryWithObjects:[[NSArray alloc]initWithObjects:@"dd", @"do", @"di", nil] forKeys:arrayNew];

NSLog(@"%@", [dic allKeys]);

5. isKindOfClass、isMemberOfClass作用分别是什么？

isKindOfClass:我们可以用isKindOfClass来确定一个对象是否是一个类的实例，或者是该类父类的实例 // Bool值 继承链上都会返回yes

isMemberOfClass只能用来判断前者，不能用来判断后者。

isMemberOfClass不能检测任何的类都是基于NSObject类这一事实

6.写出下面程序段的输出结果

NSDictionary \*dict = [NSDictionary dictionaryWithObject:@"a string value" forKey:@"akey"];

NSLog(@"%@", [dict objectForKey:@"akey"]);

[dict release];

7.请写出以下代码的执行结果

NSString \* name = [ [ NSString alloc] init ];

name = @”Habb”;

[ name release]；

8.请分别写出SEL、id的意思？

SEL 选择器 ，类方法的编号，函数地址 @selector()就是取类方法的编号

id 对象类型（有待补充）

9. iPhone上，不能被应用程序直接调用的系统程序是什么？

配置plist文件可以调用setting;

eventKit事件添加，不可以调用计算器，视频，指南针，股市，备忘录，天气，提醒事项等关于隐私的程序。

http://blog.sina.com.cn/s/blog\_7dc11a2e01016qve.html

http://blog.csdn.net/yhawaii/article/details/7587355

10.以.mm为拓展名的文件里，可以包含的代码有哪些？c和obj-c如何混用

obj-c 的编译器处理后缀为.m的文件时，可以实现objc-c和c的代码

处理.mm文件可以识别obj-c,c,c++代码

（有待补充）

11.说说如何进行后台运行程序？

ios4之后开始支持：音乐，GPS,VOIP(网络电话)，后台下载，后台网络请求，地图

（有待补充）

socket链接和http的区别：

12.sizeof和strlen的区别和联系

sizeof用来计算内存大小，srelen用来计算字符长度的

13. sprintf,strcpy,memcpy的功能？使用上要有哪些要注意的地方

sprintf：是格式化函数。将一段数据通过特定的格式，格式化到一个字符串缓冲区中去。sprintf 格式化的函数的长度不可控，有可能格式化后的字符串会超出缓冲区的大小，造成溢出。

sprint:将打印的内容输出到文件。

strcpy:是一个字符串拷贝函数，它的函数原型为strcpy(char \*dfst, const char \*src);

将src开始的一段字符串拷贝到dst开始的内存中去，结束的标志符号‘\0’,由于拷贝的长度不是我们控制的，所以这个字符串拷贝很容易出错.具备字符串拷贝功能的函数有memcpy,这是一个内存拷贝函数，它的函数原型为memcpy(char \*dst, const char \*src, unsigned int len),将长度为len的一段内存，从src 拷贝到dst中去，这个函数的长度可控，但是会有内存叠加的问题。

14.

自己写函数，实现strlen功能

unsigned long strlen(const char \*str)

{

unsigned long length = 0;

while (str[length] != '\0')

{

length++;

}

return length;

}

自己写函数实现strcpy的功能

void strcpyPersonal(char \* destination, const char \* src)

{

int i = 0;

while (\*(src + i) != '\0')

{

\*(destination + i) = \*(src + i);

i++;

}

\*(destination + i) = '\0';

}

15.写一个代码片段输入一个字符串“20130322152832”,输出一个NSDate类型的对象，打印该对象输出2013-03-22 15：28：32

NSString \*dateString = @"20130322152832";

NSDateFormatter \*formatter = [[NSDateFormatter alloc]init];

[formatter setTimeZone:[NSTimeZone timeZoneForSecondsFromGMT:8]];

[formatter setDateFormat:@"yyyyMMddHHmmss"];

NSDate \*date = [formatter dateFromString:dateString];

NSLog(@"%@", date);

// 相反的操作

NSDate \*date = [NSDate date];

NSLog(@"%@", date);

NSTimeZone \*zone = [NSTimeZone systemTimeZone];

NSLog(@"%@", zone);

NSDateFormatter \*formatter = [[NSDateFormatter alloc]init];

[formatter setTimeZone:zone];

[formatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];

NSString \*dateString = [formatter stringFromDate:date];

NSLog(@"%@", dateString);

16.找错误

a：void test1()

{

　char string[10];

　char\* str1 = "0123456789";

　strcpy( string, str1 );

}

b:void GetMemory( char \*\*p, int num )

{

　\*p = (char \*) malloc( num );

free(p);// C语言函数中的释放

}

void Test( void )

{

　char \*str = NULL;

　GetMemory( &str, 100 );

　strcpy( str, "hello" );

　printf( str );

}

17.用变量a写出以下定义

a、一个整型数 int a;

b、一个指向整型数的指针 int \*a;

c、一个指向指针的指针，它指向的指针是指向一个整型数 int \*\*a;

d、一个有10个整型数的数组 int a[10]

e、一个有10个指针的数组，该指针是指向一个整型数的 int \*a[10];

f、一个指向有10个整型数数组的指针 int (\*a)[10]

g、一个指向函数的指针，该函数有一个整型参数，并返回一个整型数

int (\*a)(int x);

- (NSArray \*)componentsSeparatedByString:(NSString \*)separator;

-(NSString \*)stringByReplacingOccurrencesOfString:(NSString \*)target withString:(NSString \*)replacement;

12、我们说的obc是动态运行时语言是什么意思? **多态。主要是将数据类型的确定由编译时,推迟到了运行时。这个问题其实浅涉及到两个概念,运行时和多态。简单来说, 运行时机制使我们直到运行时才去决定一个对象的类别,以及调用该类别对象指定方法。多态:不同对象以自己的方式响应相同的消息的能力叫做多态。意思就是假设生物类(life)都用有一个相同的方法-eat;那人类属于生物,猪也属于生物,都继承了life后,实现各自的eat,但是调用是我们只需调用各自的eat方法。也就是不同的对象以自己的方式响应了相同的消 息(响应了eat这个选择器)。因此也可以说,运行时机制是多态的基础**

19、**drawRect和layoutSubviews的区别 两个方法都是异步执行的,layoutSubviews方便数据计算,drawRect方便视图重绘。 layoutSubviews对subviews重新布局**

**layoutSubviews方法调用先于drawRect**

23、http和scoket通信的区别?socket连接相关库,TCP,UDP的连接方法,HTTP的几种常用方式? **http和scoket通信的区别:**

**http是客户端用http协议进行请求,发送请求时候需要封装http请求头,并绑定请求的数据,服务器一般有web服务器配 合(当然也非绝对)。 http请求方式为客户端主动发起请求,服务器才能给响应,一次请求完毕后则断开连接,以节省资 源。服务器不能主动给客户端响应(除非采取http长连接技术)。iphone主要使用类是NSUrlConnection。 scoket是客户端跟服务器直接使用socket“套接字”进行连接,并没有规定连接后断开,所以客户端和服务器可以保持连 接通道,双方都可以主动发送数据。一般在游戏开发或股票开发这种要求即时性很强并且保持发送数据量比较大的场合使 用。主要使用类是CFSocketRef。 UDP:是用户数据报协议:主要用在实时性要求高以及对质量相对较弱的地方,但面对现在高质量的线路不是容易丢包除非 是一些拥塞条件下 ,如流媒体 TCP:是传输控制协议:是面连接的,那么运行环境必然要求其可靠性不可丢包有良好的拥塞控制机制如http ftp telnet 等**

**http的常用方式:get,post**

25、你连接服务器用的是什么方法,如果请求过程中,网络出了问题这么办?

**NSUrlConnection 连接后,有一系列委托方法来接受来自服务器的响应和数据,其中接受相应的方法回得到服务器要传 回的数据有多大,接受数据的方法会反复调用来不断接受服务器数据,如果网络出了问题了,会调用一个方法让你来做相关 处理。**

34、objective-c中的词典对象、可变词典对象是哪个,初始化一个含有两个键值对的可变词典对象,并动态的添 加和删除一条记录,输出第一条记录**词典NSDictionary,可变词典NSMutableDictionary, //初始化一个可变词典,带有2个键值对**

**NSMutableDictionary \*dic = [NSMutableDictionary dictionaryWithObjectsAndKeys:@"value1",@"key1",@"value2",@"key2",nil]; //添加**

**[dic setObject:@"value3" forKey:@"key3"]; //删除**

**[dic removeObjectForKey:@"key3"]; //获取(按key获取)**

**[dic objectForKey:@"key1"];**

35、c和obj-c如何混用? **1)obj-c的编译器处理后缀为m的文件时,可以识别obj-c和c的代码,处理mm文件可以识别obj-c,c,c++代码,但cpp文件必须只**

**能用c/c++代码,而且cpp文件include的头文件中,也不能出现obj- c的代码,因为cpp只是cpp。**

**2) 在mm文件中混用cpp直接使用即可,所以obj-c混cpp不是问题**

**3)在cpp中混用obj- c其实就是使用obj-c编写的模块是我们想要的。如果模块以类实现,那么要按照cpp class的标准写类的定 义,头文件中不能出现obj-c的东西,包括#import cocoa的。 实现文件中,即类的实现代码中可以使用obj-c的东西,可以import,只是后缀是mm。如果模块以函数实现,那么头文件要按 c的格 式声明函数,实现文件中,c++函数内部可以用obj-c,但后缀还是mm或m。总结:只要cpp文件和cpp include的文件中不包含 obj-c的东西就可以用了,cpp混用obj-c的关键是使用接口,而不能直接使用实现代码,实际上cpp混用的是obj-c编译后的o文 件,这个东西其实是无差别的,所以可以用。obj-c的编译器支持cpp。**

38、UITableView的重用机制?如何在一个view上显示多个tableView,tableView要求不同的数据源以及不同的样

式 (要求自定义cell), 如何组织各个tableView的delegate和dataSource?请说说实现思路?在一个tableView 中需要自定义多种样式的cell(两种或三种),通常你如何实现,说说思路即可?UITableView的那些元素是可以自 定义的?

**UITableView的重用机制:**

**查看UITableView头文件,会找到NSMutableArray\*visiableCells,和NSMutableArray\*reusableTableCells两 个结构。visiableCells内保存当前显示的cells,reusableTableCells保存可重用的cells。 TableView显示之初,reusableTableCells为空,那么**

**tableView dequeueReusableCellWithIdentifier:CellIdentifier返回nil。开始的cell都 是通过 [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier] 来创 建,而且cellForRowAtIndexPath只是调用最大显示cell数的次数。 比如:有100条数据,iPhone一屏最多显示10个cell。程序最开始显示TableView的情况是:**

**1. 用[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier] 创建10次cell,并给cell指定同样的重用标识(当然,可以为不同显示类型的cell指定不同的标识)。并且10个cell全部都加 入到 visiableCells数组,reusableTableCells为空。**

**2. 向下拖动tableView,当cell1完全移出屏幕,并且 cell11(它也是alloc出来的,原因同上)完全显示出来的时候。 cell11加入到visiableCells,cell1移出 visiableCells,cell1加入到reusableTableCells。**

**3. 接着向下拖动tableView,因为reusableTableCells中已 经有值,所以,当需要显示新的cell, cellForRowAtIndexPath再次被调用的时 候,**

**tableView dequeueReusableCellWithIdentifier:CellIdentifier,返回cell1。 cell1加入到visiableCells,cell1 移出reusableTableCells;cell2移出 visiableCells,cell2加入到reusableTableCells。之后再需要显示的Cell就可 以正常重用了**􏰄

41、ios平台怎么做数据的持久化?coredata和sqlite有无必然联系?coredata是一个关系型数据库吗? **iOS中可以有四种持久化数据的方式: 属性列表、对象归档、SQLite3和Core Data coredata可以使你以图形界面的方式快速的定义app的数据模型,同时在你的代码中容易获取到它。coredata提供了基础结构去处 理常用的功能,例如保存,恢复,撤销和重做,允许你在app中继续创建新的任务。在使用coredata的时候,你不用安装额外的数据 库系统,因为coredata使用内置的sqlite数据库。coredata将你app的模型层放入到一组定义在内存中的数据对象。coredata会 追踪这些对象的改变,同时可以根据需要做相应的改变,例如用户执行撤销命令。当coredata在对你app数据的改变进行保存的时 候,core data会把这些数据归档,并永久性保存。**

**mac os x中sqlite库,它是一个轻量级功能强大的关系数据引擎,也很容易嵌入到应用程序。可以在多个平台使用,sqlite是一个轻 量级的嵌入式sql数据库编程。与coredata框架不同的是,sqlite是使用程序式的,sql的主要的API来直接操作数据表。**

**Core Data不是一个关系型数据库,也不是关系型数据库管理系统(RDBMS)。虽然Core Dta支持SQLite作为一种存储类型, 但它不能使用任意的SQLite数据库。Core Data在使用的过程种自己创建这个数据库。Core Data支持对一、对多的关系。**

45、如何将产品进行多语言发布?

**程序国际化;**

**比如:本地化应用程序名称**

**(1、选中工程,Info—Localizations点击“+”添加要国际化的语言。 (2、在InfoPlist.strings右边会多出一个三角形,点击展开可看到InfoPlish.strings(english)和**

**InfoPlish.strings(chinese)两个版本的文件; (3、在InfoPlish.strings(english)文件中加入:CFBundleDisplayName ="Program"; 其中“Program”为英文应用程序名称,同理在InfoPlish.strings(chinese)文件中加入: CFBundleDisplayName ="应用程序"; 其中“应用程序”为中文名称,注**

46、什么是动态链接库和静态链接库?调用一个类的静态方法需不需要release?􏰂**程序的编译过程--链接--- 静态连接库就是把(lib)文件中用到的函数代码直接链接进目标程序,程序运行的时候不再需要其它的库文件;动态链接就是 把调用的函数所在文件模块 (DLL)和调用函数在文件中的位置等信息链接进目标程序,程序运行的时候再从DLL中寻找 相应函数代码,因此需要相应DLL文件的支持。 静态链接库和动态链接库的另外一个区别在于静态链接库中不能再包含其他的动态链接库或者静态库,而在动态链接库中还 可以再包含其他的动态或静态链接库。**

**动态:运行时才加载到内存中,静态:编译时就加载到内存中 静态方法也就是类方法,不需要release意:CFBundleDisplayName加不加双引号都行;**

48、什么是push?远程推送?

**第一步:UIApplication向APNS注册push notification服务**

**1、应用程序 要支持 推送服务(在网页里配置) (1)https://developer.apple.com/devcenter/ios/index.action**

**(2)登录 苹果开发者账号(注意是收费账号,$99或$299) 3)下载push证书(主要是给程序签名,push服务只有收费开发者才具备。所以需要签名验证),如果没有 push证书,创建一个push证书(App ID->钥匙串程序生成request->push证书)注意事项:App ID的 Bundle ID必须和程序plist文件里的Bundle identifier一致。App ID一旦生成,将不可修改。 (4)把证书安装到钥匙串里(双击证书文件)**

**(5)生成 编译程序 用的描述文件(网页里进行) 2、向APNS注册push服务(UIApplication的registerForRemoteNotificationTypes:方法)**

**第二步 获取APNS分配的DeviceToken(64位16进制串)**

**- (void)application:(UIApplication \*)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData \*)deviceToken**

**第三步 把DeviceToken发送给自己的后台服务器,服务器记录每台设备的DeviceToken以便日后推送信息给客 户端。(需要有一个网络接口,让客户端发送DeviceToken)**

**第四步 服务器推送信息给客户端 1、服务器除了需要有客户端的DeviceToken之外,还要有push证书,对push的内容进行签名。(苹果为了防 止 恶意向客户端(比如DeviceToken泄露了)发送消息,每次推送消息,都需要证书进行签名,从而避免黑客恶 意攻击用户手机。) 2、如果你的服务器是java写的,可以直接使用钥匙串导出的p12文件(证书和密钥一起导出)。如果你的服务器 是php写的,因为php语言不支持p12文件类型,需要转换为pem文件。**

**3、将p12转换为pem文件:终端 先找到你p12所在的目录 openssl pkcs12 -in CertificateName.p12 - outCertificateName.pem -nodes**

**4、服务器发送信息给APNS,APNS自动将信息推送给客户端**

**第五步 客户端处理收到的信息**

**- (void)application:(UIApplication \*)application didReceiveRemoteNotification:(NSDictionary \*)userInfo**

**注意事项**􏰅

**1、测试版的push证书仅仅用于开发期间测试,发布版的程序需要生成一个发布版的push证书。 2、测试版APNS的ssl地址和发布版的ssl地址不同 3、测试版的DeviceToken和发布版的DeviceToken也不同**

**4、php文件要喝pem文件放在同一目录。**

**5、除了alert sound和badge之外,json串里还可以包含自定义信息。**

**6、推送的信息最大255字节 7、推送的信息受网络影响较大,有可能造成延迟甚至丢失,重要信息的传递不应该使用push通知,应该有专门的 后台接口。8、借助push推送,两个客户端可以实现即时通信,工程里面存放我们的p12文件,客户端自己组织json串,发 送请求到APNS。**

49、什么是沙箱模型?哪些操作是属于私有api范畴? **某个iphone工程进行文件操作有此工程对应的指定的位置,不能逾越。 iphone常见私有api的应用(比如直接发送短信,访问沙箱之外的磁盘文件).**

50、如何将敏感字变成\*\* **延展---几个字变为几个\*如何实现? 调用stringByReplacingOccurrencesOfString:withString:给定字符串,指定替换字,替换\*\*** 51、iphone阅读器,如果要读取一个文本文件,请问你是如何处理编码问题的?另外像pdf格式的文件,你如何读

取?

**首先检测文本编码格式(只需读取小部分用来判断), iphone手机阅读器中对于PDF格式的阅读,可以直接用UIWebView控件显示,也可以从网上下到很多直接读取pdf格式 的代码**

**直接从pdf中得到数据。复杂表格动画- (void)insertRowsAtIndexPaths:(NSArray \*)indexPaths withRowAnimation:(UITableViewRowAnimation)animation;**

**-(void)deleteRowsAtIndexPaths:(NSArray \*)indexPaths withRowAnimation: (UITableViewRowAnimation)animation;- (void)reloadRowsAtIndexPaths:(NSArray \*)indexPaths withRowAnimation:(UITableViewRowAnimation)animation;** 52、在开发大型项目的时候,如何进行内存泄露检测的? 内存泄露怎么处理?

**如何检测内存泄露:**

**可以通过xcode的自带工具run---start with performance tool里有instruments下有个leaks工具,启动此工具 后,运行项目,工具里可以显示内存泄露的情况,双击可找到源码位置,可以帮助进行内存泄露的处理。 如何处理:先定位到具体位置, 再解决之。**

53、iphone app为什么会被打回来,如何制止? 􏰂**app的设置界面、按钮使用了类似iphone的操作方式以及icon的圆角设计 -> 重新设计…**

**app的年龄设置太低 -> 改了年龄…**

**app里有实物奖励 -> 免责声明,和苹果无关…**

**app描述里提了后续版本的功能的字样 -> 删除…**

**app有打分的功能 -> 有reject的,也有通过的…**

**app需要使用location,没有提示用户 -> 加了提示,允许用户拒绝…**

**app没提供测试账号 -> 提供…**

**app里有私有api -> 修改… 应用内含有有某公司LOGO的图片,没有该公司授权文件,被拒-> 修改…**

**第三方静态库包含私有api的调用(联系第三方技术支持,更新静态库); 包含潜在的色情,暴力等内容(调整应用年龄限制等级,并加入举报功能) 做浏览器的,分级必须选17+**

54、iphone应用程序的项目基本结构?

**· Classes -> 源程序文件(.h、.m)**

**· Other Sources-> main.m 等,不需要程序员修改 -Prefix.pch**

**· Resources -> 界面文件(.xib)、配置文件-info.plist**

**· Frameworks -> 链接的库· Targets -> 项目的不同Target(资源、编译配置不同)**

**· Executables -> 项目中所有的可执行文件 -Prefix.pch:\_Prefix为所有的项目程序文件预先配置运行环境的前缀标头,在程序运行之前,引入所需框架中的(.h)头文 件。这样可以减少每个头文件对程序编译做出相同的定义,在巨型的应用程序项目开发中节省大量的时间,例如,程序有 100个根文件需要定义abc.h,只需要在\_Preix.pch文件下建立一个对象,所有的根文件便可以重复地对程序编译做出定 义。** 55、**请写出代码,用blocks来取代上例中的protocol,并比较两种方法的优势。实际应用部分?请写出代码,用**

**blocks取代协议或回调方法**

56、你做iphone开发时候,有哪些传值方式,view和view之间是如何传值的? **block, target-action ,代理,属性,**

58、编译语言和解释语言的区别 **区别:C语言,OC语言属于编译语言;解释语言:也可以理解为脚本文件,不需要编译, 编译型语言写的程序执行之前,需要一个专门的编译过程,把程序编译成为机器语言的文件,比如exe文件,以后要运行的 话就不用重新翻译了,直接使用编译的结果就行了(exe文件),因为翻译只做了一次,运行时不需要翻译,所以编译型语 言的程序执行效率高,但也不能一概而论,部分解释型语言的解释器通过在运行时动态优化代码,甚至能够使解释型语言的 性能超过编译型语言。 解释则不同,解释性语言的程序不需要编译,省了道工序,解释性语言在运行程序的时候才翻译, 比如解释性basic语言,专门有一个解释器能够直接执行basic程序,每个语句都是执行的时候才翻译。这样解释性语言每 执行一次就要翻译一次,效率比较低。解释是一句一句的翻译。**

62、描述程序启动的顺序

􏰂 **1、main.m是程序的入口**

**2、UIApplicationMain()创建应用程序对象,并且为此对象指定委托,检测程序的执行,同时开启事件循环,处理程序接收到的事 件**

**3、UIApplicationDelegate方法的执行 4、加载window**

**5、指定根视图控制器 6、在指定的试图控制器中添加控件,实现应用程序界面**

81、简述应用程序按Home键进入后台时的生命周期,和从后台回到前台时的生命周期? **应用程序:**

**-[AppDelegate application:willFinishLaunchingWithOptions:]**

**-[AppDelegate application:didFinishLaunchingWithOptions:]**

**-[AppDelegate applicationDidBecomeActive:] 退到后台:**

**-[AppDelegate applicationWillResignActive:] -[AppDelegate applicationDidEnterBackground:] 回到前台:**

**-[AppDelegate applicationWillEnterForeground:] -[AppDelegate applicationDidBecomeActive:] ViewController之间**

**加载页面:**

**-[mainViewController viewDidLoad] -[mainViewController viewWillAppear:] -[mainViewController viewWillLayoutSubviews] -[mainViewController viewDidLayoutSubviews] -[mainViewController viewDidAppear:] 退出当前页面:**

**-[mainViewController viewWillDisappear:]**

**-[mainViewController viewDidDisappear:]**

**返回之前页面:**

**-[mainViewController viewWillAppear:]**

**-[mainViewController viewWillLayoutSubviews]**

**-[mainViewController viewDidLayoutSubviews]**

**-[mainViewController viewDidAppear:]**

91、你了解svn,cvs等版本控制工具么?

**版本控制 svn、cvs 是两种版控制器,需要配套相关的svn,cvs服务器。scm是xcode里配置版本控制的地方。版本控制的原理就是a和b同时开发一个项目,a写完当天的代码之后把代码提交给服务器,b要做的时候先从服务器得到最新版本, 就可以接着做。 如果a和b都要提交给服务器,并且同时修改了同一个方法,就会产生代码冲突,如果a先提交,那么b提交 时,服务器可以提示冲突的代码,b可以清晰的看到,并做出相应的修改或融合后再提交到服务器。**

90、说说如何进行后台运行程序?

**答:判断是否支持多线程**

**UIDevice\* device = [UIDevice currentDevice];**

**BOOL backgroundSupported = NO;**

**if ([device respondsToSelector:@selector(isMultitaskingSupported)])**

**backgroundSupported = device.multitaskingSupported; 声明你需要的后台任务Info.plist中添加UIBackgroundModes键值,它包含一个或多个string的值,包括audio:在后台提供声音 播放功能,包括音频流和播放视频时的声音 location:在后台可以保持用户的位置信息 voip:在后台使用VOIP功能 前面的每个value让系统知道你的应用程序应该在适当的时候被唤醒。例如,一个应用程序,开始播放音乐,然后移动到后台仍然需要 执行时间,以填补音频输出缓冲区。添加audio键用来告诉系统框架,需要继续播放音频,并且可以在合适的时间间隔下回调应用程 序;如果应用程序不包括此项,任何音频播放在移到后台后将停止运行。除了添加键值的方法,IOS还提供了两种途径使应用程序在后 台工作:**

**Task completion—应用程序可以向系统申请额外的时间去完成给定的任务**

**Local notifications—应用程序可以预先安排时间执行local notifications 传递实现长时间的后台任务:应用程序可以请求在后台 运行以实现特殊的服务。这些应用程序并不连续的运行,但是会被系统框架在合适的时间唤醒,以实现这些服务**