**多线程网络**

1.多线程的底层实现？

1> 首先搞清楚什么是线程、什么是多线程

2> Mach是第一个以多线程方式处理任务的系统，因此多线程的底层实现机制是基于Mach的线程

3> 开发中很少用Mach级的线程，因为Mach级的线程没有提供多线程的基本特征，线程之间是独立的

4> 开发中实现多线程的方案

* C语言的POSIX接口：#include <pthread.h>
* OC的NSThread
* C语言的GCD接口（性能最好，代码更精简）
* OC的NSOperation和NSOperationQueue（基于GCD）

2．线程间怎么通信？ 主线程和子线程切换的时候传递数据

1> performSelector:onThread:withObject: waitUntilDone:

2> NSMachPort

3．网络图片处理问题中怎么解决一个相同的网络地址重复请求的问题？

利用字典（图片地址为key，下载操作为value）

4．用NSOpertion和NSOpertionQueue处理A,B,C三个线程,要求执行完A,B后才能执行C,怎么做？

// 创建队列

NSOperationQueue \*queue = [[NSOperationQueue alloc] init];

// 创建3个操作

NSOperation \*a = [NSBlockOperation blockOperationWithBlock:^{

NSLog(@”operation1---“);

}];

NSOperation \*b = [NSBlockOperation blockOperationWithBlock:^{

NSLog(@”operation1---“);

}];

NSOperation \*c = [NSBlockOperation blockOperationWithBlock:^{

NSLog(@”operation1---“);

}];

// 添加依赖

[c addDependency:a];

[c addDependency:b];

// 执行操作

[queue addOperation:a];

[queue addOperation:b];

[queue addOperation:c];

5．列举cocoa中常见对几种多线程的实现，并谈谈多线程安全的几种解决办法及多线程安全怎么控制？

1> 只在主线程刷新访问UI

2> 如果要防止资源抢夺，得用synchronized进行加锁保护

3> 如果异步操作要保证线程安全等问题, 尽量使用GCD(有些函数默认就是安全的)

6．GCD内部怎么实现的

1> iOS和OS X的核心是XNU内核，GCD是基于XNU内核实现的

2> GCD的API全部在libdispatch库中

3> GCD的底层实现主要有Dispatch Queue和Dispatch Source

* Dispatch Queue ：管理block(操作)
* Dispatch Source ：处理事件

7. 你用过NSOperationQueue么？如果用过或者了解的话，你为什么要使用NSOperationQueue，实现了什么？请描述它和GCD的区别和类似的地方（提示：可以从两者的实现机制和适用范围来描述）。

1> GCD是纯C语言的API，NSOperationQueue是基于GCD的OC版本封装

2> GCD只支持FIFO的队列，NSOperationQueue可以很方便地调整执行顺序、设置最大并发数量

3> NSOperationQueue可以在轻松在Operation间设置依赖关系，而GCD需要写很多的代码才能实现

4> NSOperationQueue支持KVO，可以监测operation是否正在执行（isExecuted）、是否结束（isFinished），是否取消（isCanceld）

5> GCD的执行速度比NSOperationQueue快

任务之间不太互相依赖：GCD

任务之间有依赖\或者要监听任务的执行情况：NSOperationQueue

8. 既然提到GCD，那么问一下在使用GCD以及block时要注意些什么？它们两是一回事儿么？block在ARC中和传统的MRC中的行为和用法有没有什么区别，需要注意些什么？

Block的使用注意：

1. block的内存管理
2. 防止循环retian

* 非ARC（MRC）：\_\_block
* ARC：\_\_weak\\_\_unsafe\_unretained

9．在异步线程中下载很多图片,如果失败了,该如何处理?请结合RunLoop来谈谈解决方案.(提示:在异步线程中启动一个RunLoop重新发送网络请求,下载图片)

1> 重新下载图片

2> 下载完毕, 利用RunLoop的输入源回到主线程刷新UIImageVIUew

**内存管理**

1. 怎么保证多人开发进行内存泄露的检查.
2. 使用Analyze进行代码的静态分析
3. 为避免不必要的麻烦, 多人开发时尽量使用ARC

2．非自动内存管理情况下怎么做单例模式.

创建单例设计模式的基本步骤 ·

>声明一个单件对象的静态实例,并初始化为nil。

>创建一个类的类工厂方法,当且仅当这个类的实例为nil时生成一个该类的实例

>实现NScopying协议, 覆盖allocWithZone:方法,确保用户在直接分配和初始化对象时,不会产 生另一个对象。

>覆盖release、autorelease、retain、retainCount方法, 以此确保单例的状态。

>在多线程的环境中,注意使用@synchronized关键字或GCD,确保静态实例被正确的创建和初始化。

3．对于类方法（静态方法）默认是autorelease的。所有类方法都会这样吗？

1> 系统自带的绝大数类方法返回的对象,都是经过autorelease的

4．block在ARC中和MRC中的用法有什么区别,需要注意什么

1.对于没有引用外部变量的Block，无论在ARC还是非ARC下，类型都是\_\_NSGlobalBlock\_\_，这种类型的block可以理解成一种全局的block，不需要考虑作用域问题。同时，对他进行Copy或者Retain操作也是无效的

2.应注意避免循环引用

5．什么情况下会发生内存泄漏和内存溢出？

当程序在申请内存后，无法释放已申请的内存空间(例如一个对象或者变量使用完成后没有释放,这个对象一直占用着内存)，一次内存泄露危害可以忽略，但内存泄露堆积后果很严重，无论多少内存,迟早会被占光。内存泄露会最终会导致内存溢出！

当程序在申请内存时，没有足够的内存空间供其使用，出现out of memory；比如申请了一个int,但给它存了long才能存下的数，那就是内存溢出。

6．[NSArray arrayWithobject:<id>] 这个方法添加对象后，需要对这个数组做释放操作吗？

不需要 这个对象被放到自动释放池中

7．Json数据的解析，和解析数据的时候有内存泄露吗？有的话 如何解

1. JSON解析的方案

* SBJson
* JSONkit
* NSJSONSerialization

8. 自动释放池底层怎么实现

自动释放池以栈的形式实现:当你创建一个新的自动释放池时，它将被添加到栈顶。当一个对象收到发送autorelease消息时,它被添加到当前线程的处于栈顶的自动释放池中,当自动释放池被回收时,它们从栈中被删除, 并且会给池子里面所有的对象都会做一次release操作.

**Foundation**

1. KVO内部实现原理
2. KVO是基于runtime机制实现的
3. 当某个类的对象第一次被观察时， 系统就会在运行期动态地创建该类的一个派生类，在这个派生类中重写基类中任何被观察属性的 setter 方法。

派生类在被重写的 setter 方法实现真正的通知机制（Person🡪 NSKVONotifying\_Person）

1. 是否可以把比较耗时的操作放在NSNotificationCenter中

如果在异步线程发的通知，那么可以执行比较耗时的操作；

如果在主线程发的通知，那么就不可以执行比较耗时的操作

3．Foundation对象与Core Foundation对象有什么区别

1> Foundation对象是OC的，Core Foundation对象是C对象

2> 数据类型之间的转换

* ARC:\_\_bridge\_retained、\_\_bridge\_transfer
* 非ARC: \_\_bridge

4．不用中间变量,用两种方法交换A和B的值

A = A + B

B = A - B

A = A - B

或者

A = A^B;

B = A^B;

A = A^B;

5．简单描述下对单利模式设计的理解？

节省内存资源，一个应用就一个对象。

6．什么是动态，举例说明

1> 在程序运行过程才执行的操作

7．runtime实现的机制是什么,怎么用，一般用于干嘛. 你还能记得你所使用的相关的头文件或者某些方法的名称吗？

运行时机制，runtime库里面包含了跟类、成员变量、方法相关的API，比如获取类里面的所有成员变量，为类动态添加成员变量，动态改变类的方法实现，为类动态添加新的方法等 需要导入<objc/message.h><objc/runtime.h>

1. runtime，运行时机制，它是一套C语言库
2. 实际上我们编写的所有OC代码，最终都是转成了runtime库的东西，比如类转成了runtime库里面的结构体等数据类型，方法转成了runtime库里面的C语言函数，平时调方法都是转成了objc\_msgSend函数（所以说OC有个消息发送机制）
3. 因此，可以说runtime是OC的底层实现，是OC的幕后执行者
4. 有了runtime库，能做什么事情呢？runtime库里面包含了跟类、成员变量、方法相关的API，比如获取类里面的所有成员变量，为类动态添加成员变量，动态改变类的方法实现，为类动态添加新的方法等
5. 因此，有了runtime，想怎么改就怎么改

8．是否使用Core Text或者Core Image等？如果使用过，请谈谈你使用Core Text或者Core Image的体验。

CoreText

* 随意修改文本的样式
* 图文混排(纯C语言)
* 国外:Niumb

Core Image(滤镜处理)

\* 能调节图片的各种属性(对比度, 色温, 色差等)

9．NSNotification和KVO的区别和用法是什么？什么时候应该使用通知，什么时候应该使用KVO，它们的实现上有什么区别吗？如果用protocol和delegate（或者delegate的Array）来实现类似的功能可能吗？如果可能，会有什么潜在的问题？如果不能，为什么？（虽然protocol和delegate这种东西面试已经面烂了…）

通知比较灵活(1个通知能被多个对象接收, 1个对象能接收多个通知),

代理比较规范，但是代码多(默认是1对1)

KVO性能不好(底层会动态产生新的类)，只能监听某个对象属性的改变, 不推荐使用(1个对象的属性能被多个对象监听, 1个对象能监听多个对象的其他属性)

更详细参考：

http://blog.csdn.net/dqjyong/article/details/7685933

10．Block内部的实现原理

Objective-C是对C语言的扩展，block的实现是基于指针和函数指针

**数据存储**

1. 如果后期需要增加数据库中的字段怎么实现，如果不使用CoreData呢？

编写SQL语句来操作原来表中的字段

1. 增加表字段

ALTER TABLE 表名 ADD COLUMN 字段名 字段类型;

1. 删除表字段

ALTER TABLE 表名 DROP COLUMN 字段名;

1. 修改表字段

ALTER TABLE 表名 RENAME COLUMN 旧字段名 TO 新字段名;

1. SQLite数据存储是怎么用？
2. 添加SQLite动态库：



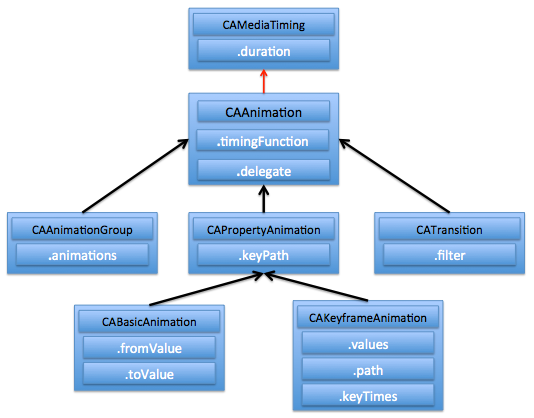
1. 导入主头文件：#import <sqlite3.h>
2. 利用C语言函数创建\打开数据库，编写SQL语句
3. 简单描述下客户端的缓存机制？
4. 缓存可以分为：内存数据缓存、数据库缓存、文件缓存
5. 每次想获取数据的时候
6. 先检测内存中有无缓存
7. 再检测本地有无缓存(数据库\文件)
8. 最终发送网络请求
9. 将服务器返回的网络数据进行缓存（内存、数据库、文件）， 以便下次读取
10. 你实现过多线程的Core Data么？NSPersistentStoreCoordinator，NSManagedObjectContext和NSManagedObject中的哪些需要在线程中创建或者传递？你是用什么样的策略来实现的？
11. CoreData是对SQLite数据库的封装
12. CoreData中的NSManagedObjectContext在多线程中不安全
13. 如果想要多线程访问CoreData的话，最好的方法是一个线程一个NSManagedObjectContext
14. 每个NSManagedObjectContext对象实例都可以使用同一个NSPersistentStoreCoordinator实例，这是因为NSManagedObjectContext会在便用NSPersistentStoreCoordinator前上锁

**UI控件**

1. 怎么解决缓存池满的问题(cell)

ios中不存在缓存池满的情况，因为通常我们ios中开发，对象都是在需要的时候才会创建，有种常用的说话叫做懒加载，还有在UITableView中一般只会创建刚开始出现在屏幕中的cell，之后都是从缓存池里取，不会在创建新对象。缓存池里最多也就一两个对象，缓存池满的这种情况一般在开发java中比较常见，java中一般把最近最少使用的对象先释放。

1. CAAnimation的层级结构



1. UIButton与UITableView的层级结构
2. 继承结构
3. 内部的子控件结构
4. 如何渲染自定义格式字符串的UILabel

1> 通过NSAttributedString类

1. 设置scroll view的contensize能在Viewdidload里设置么,为什么

1> 能

1. 按钮或者其它UIView控件的事件传递的具体过程

1> 回顾事件处理课程即可

1. 控制器View的生命周期及相关函数是什么？你在开发中是如何用的？

\* 1.首先判断控制器是否有视图，如果没有就调用loadView方法创建：通过storyboard或者代码；

\* 2.随后调用viewDidLoad，可以进行下一步的初始化操作；只会被调用一次；

\* 3.在视图显示之前调用viewWillAppear；该函数可以多次调用；

\* 4.视图viewDidAppear

\* 3.在视图显示之前调用viewWillDisappear；该函数可以多次调用；

如需要）；

\* 5.在布局变化前后，调用viewWill/DidLayoutSubviews处理相关信息；

**应用程序**

1.NSRunLoop的实现机制,及在多线程中如何使用

NSRunLoop是IOS消息机制的处理模式

>1.NSRunLoop的主要作用：控制NSRunLoop里面线程的执行和休眠，在有事情做的时候使当前NSRunLoop控制的线程工作，没有事情做让当前NSRunLoop的控制的线程休眠。

>2.NSRunLoop 就是一直在循环检测，从线程start到线程end，检测inputsource(如点击，双击等操作)异步事件，检测timesource同步事件，检测到输入源会执行处理函数，首先会产生通知，corefunction向线程添加runloop observers来监听事件，意在监听事件发生时来做处理。

>3.runloopmode是一个集合，包括监听：事件源，定时器，以及需通知的runloop observers

>1. 只有在为你的程序创建次线程的时候，才需要运行run loop。对于程序的主线程而言，run loop是关键部分。Cocoa提供了运行主线程run loop的代码同时也会自动运行run loop。IOS程序UIApplication中的run方法在程序正常启动的时候就会启动run loop。如果你使用xcode提供的模板创建的程序，那你永远不需要自己去启动run loop

>2. 在多线程中，你需要判断是否需要run loop。如果需要run loop，那么你要负责配置run loop并启动。你不需要在任何情况下都去启动run loop。比如，你使用线程去处理一个预先定义好的耗时极长的任务时，你就可以毋需启动run loop。Run loop只在你要和线程有交互时才需要

2．IOS7之前,后台执行内容有几种形式,都是什么

一般的应用在进入后台的时候可以获取一定时间来运行相关任务，也就是说可以在后台运行一小段时间(10S左右)。

1. 后台播放音乐

2. 后台GPS跟踪

3. 后台voip支持

3简单说一下APP的启动过程,从main文件开始说起

程序启动分为两类:1.有storyboard 2.没有storyboard

有storyboard情况下:

1.main函数

2.UIApplicationMain

\* 创建UIApplication对象

\* 创建UIApplication的delegate对象

3.根据Info.plist获得最主要storyboard的文件名,加载最主要的storyboard(有storyboard)

\* 创建UIWindow

\* 创建和设置UIWindow的rootViewController

\* 显示窗口

没有storyboard情况下:

1.main函数

2.UIApplicationMain

\* 创建UIApplication对象

\* 创建UIApplication的delegate对象

3.delegate对象开始处理(监听)系统事件(没有storyboard)

\* 程序启动完毕的时候, 就会调用代理的application:didFinishLaunchingWithOptions:方法

\* 在application:didFinishLaunchingWithOptions:中创建UIWindow

\* 创建和设置UIWindow的rootViewController

\* 显示窗口

4把程序自己关掉和程序进入后台,远程推送的区别

1. 关掉后不执行任何代码, 不能处理事件

2. 应用程序进入后台状态不久后转入挂起状态。在这种状态下，应用程序不执行任何代码，并有可能在任意时候从内存中删除。只有当用户再次运行此应用，应用才会从挂起状态唤醒，代码得以继续执行

3.或者进入后台时开启多任务状态，保留在内存中，这样就可以执行系统允许的动作

4.远程推送是由远程服务器上的程序发送到APNS,再由APNS把消息推送至设备上的程序,当应用程序收到推送的消息会自动调用特定的方法执行事先写好的代码

5本地通知和远程推送通知对基本概念和用法？

本地通知和远程推送通知都可以向不在前台运行的应用发送消息,这种消息既可能是即将发生的事件,也可能是服务器的新数据.不管是本地通知还是远程通知,他们在程序界面的显示效果相同,都可能显示为一段警告信息或应用程序图标上的微章.

本地通知和远程推送通知的基本目的都是让应用程序能够通知用户某些事情, 而且不需要应用程序在前台运行.二者的区别在于本地通知由本应用负责调用,只能从当前设备上的iOS发出, 而远程通知由远程服务器上的程序发送到APNS,再由APNS把消息推送至设备上的程序

**第三方框架**

1. 第三方API是怎么用的？
2. 大公司的开放API
3. github上面别人发布的框架

用

* 第三方API在官方文档都有说明，按照官方文档的一步一步做
* 参考官方提供的示例程序
* 先自己创建一个工程试试，等熟悉了，在使用到项目中

1. 列举现在熟悉iOS开发库和第三方开发库？

友盟（包括第三方登录和分享），高德地图，百度地图,AFN,SDWebimage,FMDB, MBProgressHUD等等

**其他**

1. 客户端安全性处理方式？
2. 网络数据传输(敏感数据[账号\密码\消费数据\银行卡账号], 不能明文发送)
3. 协议的问题(自定义协议, 游戏代练)
4. 本地文件存储(游戏的存档)
5. 源代码
6. sip是什么？
7. SIP（Session Initiation Protocol），会话发起协议
8. SIP是建立VOIP连接的 IETF 标准，IETF是全球互联网最具权威的技术标准化组织
9. 所谓VOIP，就是网络电话，直接用互联网打电话，不用耗手机话费
10. 有些图片加载的比较慢怎么处理?你是怎么优化程序的性能的?
11. 图片下载放在异步线程
12. 图片下载过程中使用占位图片
13. 如果图片较大，可以考虑多线程断点下载
14. 你实现过一个框架或者库以供别人使用么？如果有，请谈一谈构建框架或者库时候的经验；如果没有，请设想和设计框架的public的API，并指出大概需要如何做、需要注意一些什么方面，来使别人容易地使用你的框架。
15. 提供给外界的接口功能是否实用、够用
16. 别人使用我的框架时，能不能根据类名、方法名就猜出接口的具体作用
17. 别人调用接口时，提供的参数是否够用、调用起来是否简单
18. 别人使用我的框架时，要不要再导入依赖其他的框架
19. App需要加载超大量的数据，给服务器发送请求，但是服务器卡住了如何解决？
20. 设置请求超时
21. 给用户提示请求超时
22. 根据用户操作再次请求数据
23. 利用NSNotificationCenter实现跨多控制器传输数据和消息中用同步还是异步？
24. 如何编写xml
25. 地图导航不能用了怎么办

提示用户打开导航定位功能

1. SDWebImage具体如何实现
2. 利用NSOperationQueue和NSOperation下载图片, 还使用了GCD的一些函数(解码GIF图片)
3. 利用URL作为key，NSOperation作为value
4. 利用URL作为key，UIImage作为value
5. AFN 与 ASI 有什么区别
6. AFN基于NSURL，ASI基于底层的CFNetwork框架，因此ASI的性能优于AFN
7. AFN采取block的方式处理请求，ASI最初采取delegate的方式处理请求，后面也增加了block的方式
8. AFN只封装了一些常用功能，满足基本需求，直接忽略了很多扩展功能，比如没有封装同步请求；ASI提供的功能较多，预留了各种接口和工具供开发者自行扩展
9. AFN直接解析服务器返回的JSON、XML等数据，而ASI比较原始，返回的是NSData二进制数据
10. 你在实际开发中，有哪些手机架构与性能调试经验
11. 刚接手公司的旧项目时，模块特别多，而且几乎所有的代码都写在控制器里面，比如UI控件代码、网络请求代码、数据存储代码
12. 接下来采取MVC模式进行封装、重构

* 自定义UI控件封装内部的业务逻辑
* 封装网络请求工具类
* 封装数据存储工具类

1. runloop定时源和输入源
2. 你创建的程序不需要显示的创建run loop；每个线程，包括程序的主线程（main thread）都有与之相应的run loop对象, 主线程会自行创建并运行run loop
3. Run loop处理的输入事件有两种不同的来源：输入源（input source）和定时源（timer source）
4. 输入源传递异步消息，通常来自于其他线程或者程序。定时源则传递同步消息，在特定时间或者一定的时间间隔发生
5. 你们项目中都用了哪些框架, 及何种开发工具,具体到是哪个版本,这个版本的特性有哪些???(比如xcode的版本)
6. AFN、SDWebImage、MBProgressHUD等 (搜索: 网易新闻 第三方库)
7. Xcode 5.0
8. 自己怎么实现原子锁

iOS中一般用nonaotomic

1. 运行时的内部实现
2. 即时通讯中的大数据处理

用put上传到文件服务器，然后发带url的自定义格式的给对方，对方接收到之后下载

1. json解析的具体实现
2. SBJson
3. JSONKit
4. NSJSONSerialization
5. 如果在网络数据处理过程中,发现一处比较卡,一般怎么解决
6. 检查网络请求操作是否被放在主线程了
7. 看看异步请求的数量是否太多了（子线程数量）
8. 数据量是否太大？如果太大，先清除一些不必要的对象（看不见的数据、图片）
9. 手机CPU使用率和内存问题
10. 怎么介绍一个项目
11. 项目的价值（可以加些“老板”关键字）
12. 项目的模块
13. 我做的是哪个模块
14. 怎么解决sqlite锁定的问题
15. 设置数据库锁定的处理函数

int sqlite3\_busy\_handler(sqlite3\*, int(\*)(void\*,int), void\*);

1. 设定锁定时的等待时间

int sqlite3\_busy\_timeout(sqlite3\*, 60); ：