

Objective-C 部分

1. 你如何理解 OC 的内存管理

OC 内存管理是基于引用计数。谁想使用某个对象 B，就要把对象 B 的计数器+1，如果不使用这个对象了，那么就把对象 B 计数器-1，如果 B 对象计数器减到 0，那么 B 对象会自动调用自己的 dealloc 函数，也就是这个对象被销毁。

一句话就是 谁拥有对象，谁负责释放该对象，谁 alloc 谁应该 release 或者 autorelease

系统有一个自动释放池，对于一些对象我们可以放在自动释放池中。如果自动释放池本身销毁，那么里面的内容会被全部调用 release。

2. assign, retain, copy, autorelease, release, dealloc 的不同

assign 就是基本的赋值，就是 C 的基本的赋值

retain 就是让对象引用计数+1，表示拥有了这个对象

copy 把对象复制一份。copy 是深拷贝，浅拷贝

autorelease 把对象放在自动释放池里面

release 表示对象计数器-1 如果减到 0 那么对象自动 dealloc

3. self.name= “yang”和 name=“yang”的关系

self.name 表示调用 setter 调用一个函数

name=”year”表示给字段 field 赋值

4. 什么是深拷贝和浅拷贝 copy

浅拷贝只是拷贝对象本身，不递归的拷贝里面的子对象

深拷贝是把对象以及对象的对象递归拷贝。注意强调 mutableCopy 不是深拷贝

如果要对自己对象 BookModel 实现拷贝 那么必须实现 NSCopying 协议

实现里面的 - (id) copyWithZone:(NSZone *)zone;

5. copy 和 mutableCopy 是什么

mutableCopy 可变拷贝 NSString --> mutableCopy->NSMutableString

NSArray --> mutableCopy->NSMutableArray;

NSMutableString --> copy---NSString

6. #import 和#include 区别

#import 对于包含多个相同文件只是包含一次 防止了重复包含

#include 需要加上

```
#ifndef __QF_Header__
```

```
#define __QF_Header__
```

```
#endif
```

7. OC 有没有 GC/Gabbage Collector 垃圾回收

iPhone/OC 没有 GC 当时 Mac 本地的 OC 是有 GC 的

8. KVC 是什么，有什么好处

KVC 就是 key value coding. 主要用来大量的对象赋值上。比如网络下载 json 数据进行解析，然后要给数据模型赋值。

```
bookModel.name = [dict objectForKey:@"name"];
```

```
bookModel.author = [dict objectForKey:@"author"];
```

如果要用 kvc 那么就可以简写成

```
NSArray *arr = [NSArray arrayWithObjects:@"name", @"author", nil];
for(NSString *field in arr) {
    NSString *v = [dict objectForKey:field];
    bookModel setValue:v forKey:field;
}
```

kvc 有 2 个方法 setValue:forKey:和 valueForKey:方法分别对应 setter 和 getter 方法

KVO 主要用在属性的监听上。KVO 比如用在播放视频的时候我想知道当前的播放进度

协议和代理的关系和区别

协议就是规范接口(让每个对象都有相同的函数), 主要是用来在多个对方中数据传输的统一接口。类似于 java 中得接口。OC 协议的所有方法不一定要全部实现。@optional, 当时 java 中得接口全部需要实现。

代理是一种设计模式。在 iOS 中主要是用来做反向传值。

OC 里面的 NSArray, NSDictionary 如何实现的

NSArray 使用链表实现, NSDictionary 使用 hash 表

什么是单例?

单例就是一个全局对象, 就是只有一个对象, 目的是为了数据共享。

单例在多线程重要注意什么? 需要加锁 OC, 避免多个线程同时创建单例

```
+(id) sharedInstance {
    static id _s;
    @synchronized(self) {
        if (_s == nil) _s = [[[self class] alloc] init];
    }
    return _s;
}
```

界面之间传值有哪些方法?

正向传就是属性, 方法

反向就是代理。用单例, 用 blocks, 用数据库, 通知中心

NSNotification 原理是啥?

原理就是接收端把接受函数注册到一个数组中, 发送端发送数据就直接调用之前存得函数

NSNotification 是同步的。不是异步的。

1. NSNotification 是同步还是异步的

是同步的. 如果需要异步 必须使用 NSNotificationQueue 实现

A notification center delivers notifications to observers synchronously. In other words, the postNotification: methods do not return until all observers have received and processed the notification. To send notifications asynchronously use **NSNotificationQueue**.

In a multithreaded application, notifications are always delivered in the thread in which the

notification was posted, which may not be the same thread in which an observer registered itself.

Hope it helps you.

类别 Category 和继承区别

类别是用于给一个类扩展方法，不用保存变量，因为类别不能保存变量

集成扩展方法和变量。比较重一些。类别 Category 相对轻量一些。当时不能完全替代集成。

你如何理解 ScrollView 复用, TableView 才有

下拉刷新原理是?

EGO

iOS5, iOS6, iOS7 适配

```
if (isIOS7) {  
    // 让iOS7 导航控制器不透明  
    self.navigationController.navigationBar.translucent = NO;  
    self.edgesForExtendedLayout = UIRectEdgeNone;  
    // 可以认为上面2 让它恢复成iOS6的经典模式  
}
```

对于图片需要比如实现一个宏来取得正确的版本的图片

iphone5高度适配，要动态的获取高度

UI 部分

网络部分

2. 你如何理解线程

iOS 线程是为了界面流畅，防止界面假死。

比如解析一个大文件，比如一个地区详情文件，解析时间比如是 10s，那么不用线程的话，界面就会假死/卡 10s，用户体验差，那么解决方法就是很快把界面暂时，启动线程去后台解析。

再比如，做图片滤镜。图像操作会耗费很多时间，比如 5s，那么界面会假死 5s，那么用户体验差，解决方法是启动线程在后台使用滤镜算法处理图像，一旦处理完就通知主线程刷新界面。

在比如：网络下载数据，由于网络状态好坏时间会不确定，那么我们需要使用线程来处理这种不确定的关系，一定数据接收完成，那么就可以通知主线程处理。

再比如读取数据全国电话号码数据库，时间很长，那么就需要通过线程来做

特别强调的是，线程不是快。是为了给用户一个快得假象

3. 你如何理解线程池

线程池就是一堆项目的集合，就是一个线程队列来管理这些线程对象。如果需要处理一些任务，可以从线程池中直接取线程，而不是临时创建线程。比如 `NSOperationQueue` 就是一个线程池，管理 `NSOperation` 线程对象

线程锁 `NSLock`

线程锁是多个线程要访问同一个资源引入的一种机制。比如 2 个线程同时写数据文件，那么需要用锁 `NSLock` 来保证同一个时刻只能有一个线程写。写结束后另外一个线程才能接着写。比如创建单例的时候，为了保证哪怕多个线程同时创建单例，也只能创建一个单例对象。

4. 你如何理解图片下载

图片下载一般使用 `Lazy` 懒加载和图片缓存的模式。懒加载就是需要的时候才去临时记载。主要是节省内存空间和用户流量。缓存就是把 `url` 地址做成 `md5` 存在沙盒中以便下次直接使用。

3. ASI 的原理是啥??

ASI 使用 Apple 底层的 `CFNetwork` 框架实现的。而不是用 `BSD Socket` 实现的。他是一个基于 `NSOperation` 的线程处理网络框架。`CFNetwork` 是基于 `InputStream/OutputStream` 流的方式管理数据。它内部使用了多线程异步模式进行数据的通讯。比如数据的上传进度，下载进度，缓存的管理机制。大文件下载，大文件上传，安全机制等。

4. `SDWebImage` 原理是什么?

`SDWebImage` 是一个基于 `NSOperation/NSInvocationOperation` 的一个多线程图片缓存框架 `framework`,

原理是每次通过 `[UIImageView setImageWithURL:]` 就启动一个异步线程,负责下载网络图片,一旦下载完成就把网址使用 `md5` 做一个校验 `checksum`, 然后存在一个沙盒 `sandbox` 管理目录中,以便下次可以直接从 `sandbox` 中读取,而不需要从网络上下载,目的是为了节省用户流量,加快图片访问速度。

5. HTTP 原理是?

HTTP 是基于 `TCP/IP` 的一个网络连接协议,它是客户端和服务端通讯的常用方式。他是使用请求,响应模型,客户端请求,服务端响应的方式。HTTP 协议有请求头,请求体,响应头,响应体。`XML/Json` 是一般作为响应的数据格式传输。`GET/POST` 是和请求的 2 中常见方式。还有 2 中 `PUT, DELETE`

TCP/IP 的原理是?

`TCP/IP` 是网络底层的二进制协议。它是网络传输的标准协议。`TCP` 是稳定,可靠的点对点传输,`UDP` 是不可靠的,高速的网络传输协议。在手机客户单中,一般用在聊天类 `IM(Instance Message)`,或者视频流,语音等大数据中。`TCP` 可以和服务器保持长连接。一般使用 `Socket` 来进行 `TCP/IP` 开发程序,常见的套接字是 `AsyncSocket, BSD Socket`。或者 `CFNetwork` 都可以

HTTP 断点下载原理

在 HTTP 协议头上加上 Range 字段就可以使用断点下载，每次取得文件大小，然后设置 Range 字段，然后下载的数据追加到文件的最后。这种方式就是断点下载。

Socket 原理是什么？

AsyncSocket 是 iOS 中一个 TCP/IP 通讯的开源的套接字框架。比 BSD Socket 好用一些。AsyncSocket 使用了 CFNetwork 中异步回调模型。所有的收发数据都是使用代理来通知。内部基于 CFNetwork 里面的 Stream 流的模式。他支持 TCP/UDP,支持 UDP 广播，组播，比如 XMPP 里面就用了 AsyncSocket 来进行底层通讯。

Cookie 原理

Cookie 类似于 Token，主要是为了让服务端能够识别当前用户。在手机上一般使用 token 模式。在浏览器上比较喜欢 cookie, iOS NSURLConnection 和 ASI 都支持 Cookie，Cookie 就是一块小的数据存储空间。

MVC

MVC 是 Model, View, Controller 的简称，主要是为了解决 Model 和 View 界面之间耦合度的问题（保证低耦合）。保证数据模型和刷新界面相对独立。Controller 就是在中间把 Model 和 View 耦合在一起。因为大型软件要求 Model 尽量和界面进行独立一些。比如很多界面有列表模式，地图模式。界面有 2 分，当时数据模型就一份。

比如：iOS 常见的开发大多基于 MVC 模型，比如 Controller 去启动网络，数据处理，然后解析数据，然后建立数据模型，然后把数据模型关联到 TableViewCell 上。这种就是典型的 MVC

6. weibo/OAuth 分享原理是什么

Weibo 使用了 OAuth 网络协议。目前使用 OAuth2.0 协议。他是基于 HTTP 的一个网络认证协议，通过 AppKey 等调用 weibo 的页面，然后用户输入账号，weibo 密码服务器返回成功后的 token。

7. 微信，支付宝，银联原理是什么

微信，支付宝，银联都是在真机上使用。我们在程序中调用微信,支付宝，银联等程序，然后给他们传入相应地参数接口，这个就是 2 个程序之间项目的传输数据。是通过 URL 进行识别的。具体他们协议都没有公开，所以我们需要他们自己的 SDK 进行开发。

8. XMPP 原理？

XMPP 是基于 Socket 套接字的一个专门聊天 IM 的协议。他支持注册，登陆，以及单聊，群组聊天功能。对图片，语音，文件支持不好，一般使用 HTTP 来进行转发一下。或者做一个 base64 编码。XMPP 是一个即时聊天框架，支持和 googletalk msn 早期的 qq 进行通讯。他内部支持重连机制(断线后自动重连)。心跳机制。内部使用 CoreData 进行数据的管理。

CoreData 是什么？

CoreData 是 Apple 自己的一个数据库管理框架，它好处是可以避免使用 sql 语句，是一个图形化的操作框架，当时从开发者角度来说，不然 FMDB/sql 直接使用 sql 语句更方便。因

为 FMDB 可以支持支持标准的 sql 语句。支持事务 Transition，回滚 roolback 等。

CoreText 是什么？

CoreText 是 Apple 官方一个标准专业排版框架。它可以精确的处理图片，问题的位置。一般用 CoreText 用在图文混排上。

FMDB 的原理是？

FMDB 是一个开源的封装了 Apple 底层 C 的 sqlite3 数据库，他使用方便。支持标准 sql 语句，支持多线程处理，支持是事务，回滚等。