**面试题**

1、判断IP地址合法性以及优化方法

2、单链表逆置

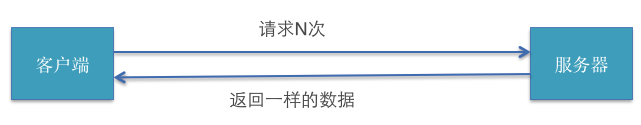
iOS相关

1. cache缓存机制

**iOS开发网络篇—数据缓存**

**一、关于同一个URL的多次请求**

　　有时候，对同一个URL请求多次，返回的数据可能都是一样的，比如服务器上的某张图片，无论下载多少次，返回的数据都是一样的。



上面的情况会造成以下问题

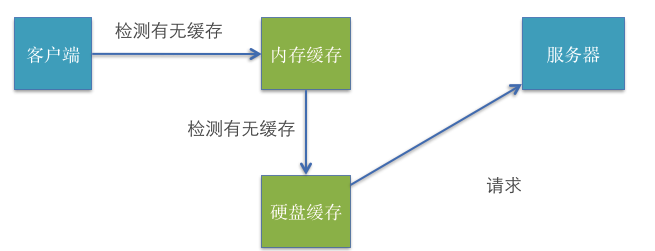
（1）用户流量的浪费

（2）程序响应速度不够快

解决上面的问题，一般考虑对数据进行缓存。

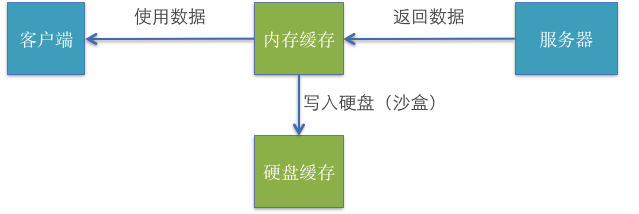
**二、缓存**

　　为了提高程序的响应速度，可以考虑使用缓存（内存缓存\硬盘缓存）



　　第一次请求数据时，内存缓存中没有数据，硬盘缓存中没有数据。

**缓存数据的过程**

****

当服务器返回数据时，需要做以下步骤

（1）使用服务器的数据（比如解析、显示）

（2）将服务器的数据缓存到硬盘（沙盒）

此时缓存的情况是：内存缓存中有数据，硬盘缓存中有数据。

再次请求数据分为两种情况：

（1）如果程序并没有被关闭，一直在运行

　　那么此时内存缓存中有数据，硬盘缓存中有数据。如果此时再次请求数据，直接使用内存缓存中的数据即可

（2）如果程序重新启动

　　那么此时内存缓存已经消失，没有数据，硬盘缓存依旧存在，还有数据。如果此时再次请求数据，需要读取内存中缓存的数据。

提示：从硬盘缓存中读取数据后，内存缓存中又有数据了

**三、缓存的实现**

1.说明：

由于GET请求一般用来查询数据，POST请求一般是发大量数据给服务器处理（变动性比较大）

因此一般只对GET请求进行缓存，而不对POST请求进行缓存

　　在iOS中，可以使用NSURLCache类缓存数据

　　iOS 5之前：只支持内存缓存。从iOS 5开始：同时支持内存缓存和硬盘缓存

2.NSURLCache

iOS中得缓存技术用到了NSURLCache类。

缓存原理：一个NSURLRequest对应一个NSCachedURLResponse

缓存技术：把缓存的数据都保存到数据库中。

3.NSURLCache的常见用法

（1）获得全局缓存对象（没必要手动创建）NSURLCache \*cache = [NSURLCache sharedURLCache];

（2）设置内存缓存的最大容量（字节为单位，默认为512KB）- (void)setMemoryCapacity:(NSUInteger)memoryCapacity;

（3）设置硬盘缓存的最大容量（字节为单位，默认为10M）- (void)setDiskCapacity:(NSUInteger)diskCapacity;

（4）硬盘缓存的位置：沙盒/Library/Caches

（5）取得某个请求的缓存- (NSCachedURLResponse \*)cachedResponseForRequest:(NSURLRequest \*)request;

（6）清除某个请求的缓存- (void)removeCachedResponseForRequest:(NSURLRequest \*)request;

（7）清除所有的缓存- (void)removeAllCachedResponses;

4.缓存GET请求

　　要想对某个GET请求进行数据缓存，非常简单

　　NSMutableURLRequest \*request = [NSMutableURLRequest requestWithURL:url];

　　// 设置缓存策略

　　request.cachePolicy = NSURLRequestReturnCacheDataElseLoad;

　　只要设置了缓存策略，系统会自动利用NSURLCache进行数据缓存

5.iOS对NSURLRequest提供了7种缓存策略：（实际上能用的只有4种）

NSURLRequestUseProtocolCachePolicy // 默认的缓存策略（取决于协议）

NSURLRequestReloadIgnoringLocalCacheData // 忽略缓存，重新请求

NSURLRequestReloadIgnoringLocalAndRemoteCacheData // 未实现

NSURLRequestReloadIgnoringCacheData = NSURLRequestReloadIgnoringLocalCacheData // 忽略缓存，重新请求

NSURLRequestReturnCacheDataElseLoad// 有缓存就用缓存，没有缓存就重新请求

NSURLRequestReturnCacheDataDontLoad// 有缓存就用缓存，没有缓存就不发请求，当做请求出错处理（用于离线模式）

NSURLRequestReloadRevalidatingCacheData // 未实现

6.缓存的注意事项

缓存的设置需要根据具体的情况考虑，如果请求某个URL的返回数据：

　　（1）经常更新：不能用缓存！比如股票、彩票数据

　　（2）一成不变：果断用缓存

　　（3）偶尔更新：可以定期更改缓存策略 或者 清除缓存

提示：如果大量使用缓存，会越积越大，建议定期清除缓存

四、简单的代码示例

- (void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event  
{  
// 1.创建请求  
NSURL \*url = [NSURL URLWithString:@"http://127.0.0.1:8080/YYServer/video"];  
NSMutableURLRequest \*request = [NSMutableURLRequest requestWithURL:url];  
  
// 2.设置缓存策略(有缓存就用缓存，没有缓存就重新请求)  
request.cachePolicy = NSURLRequestReturnCacheDataElseLoad;  
  
// 3.发送请求  
[NSURLConnection sendAsynchronousRequest:request queue:[NSOperationQueue mainQueue] completionHandler:^(NSURLResponse \*response, NSData \*data, NSError \*connectionError) {  
if (data) {  
NSDictionary \*dict = [NSJSONSerialization JSONObjectWithData:data options:NSJSONReadingMutableLeaves error:nil];  
  
NSLog(@"%@", dict);  
}  
}];  
}

/\*\*  
// 定期处理缓存  
// if (缓存没有达到7天) {  
// request.cachePolicy = NSURLRequestReturnCacheDataElseLoad;  
// }  
// 获得全局的缓存对象  
NSURLCache \*cache = [NSURLCache sharedURLCache];  
// if (缓存达到7天) {  
// [cache removeCachedResponseForRequest:request];  
// }  
  
// lastCacheDate = 2014-06-30 11:04:30  
  
NSCachedURLResponse \*response = [cache cachedResponseForRequest:request];  
if (response) {  
NSLog(@"---这个请求已经存在缓存");  
} else {  
NSLog(@"---这个请求没有缓存");  
}  
\*/

1. 隐式动画与显示动画区别

1、隐式动画一直存在 如需关闭需设置；显式动画是不存在，如需显式 要开启(创建)。

2、显式动画是指用户自己通过beginAnimations:context:和commitAnimations创建的动画。  
隐式动画是指通过UIView的animateWithDuration:animations:方法创建的动画。

3、隐式动画是系统框架自动完成的。Core Animation在每个runloop周期中自动开始一次新的事务，即使你不显式的用[CATransaction begin]开始一次事务，任何在一次runloop循环中属性的改变都会被集中起来，然后做一次0.25秒的动画。在iOS4中，苹果对UIView添加了一种基于block的动画方法：+animateWithDuration:animations:。这样写对做一堆的属性动画在语法上会更加简单，但实质上它们都是在做同样的事情。CATransaction的+begin和+commit方法在+animateWithDuration:animations:内部自动调用，这样block中所有属性的改变都会被事务所包含。

4、显式动画，Core Animation提供的显式动画类型，既可以直接对退曾属性做动画，也可以覆盖默认的图层行为。我们经常使用的CABasicAnimation，CAKeyframeAnimation，CATransitionAnimation，CAAnimationGroup等都是显式动画类型，这些CAAnimation类型可以直接提交到CALayer上。  
无论是隐式动画还是显式动画，提交到layer后，经过一系列处理，最后都经过上文描述的绘制过程最终被渲染出来

1. block内部结构和原理

4、tableView的高度预估机制

5、autorealease的原理和应用场景

6、KVC和KVO

7、tableView优化方法

8、property属性

9、多线程GCD和NSOperation比较

10、NSCopy协议

11、pushviewController后view的释放时机

12、weak应用场景 和assign区别：weak实现原理

13、block内部修改外部变量，\_bock修饰符原理，闭包

14、有没有看过哪些优秀第三方库的源码。例如:SDWebImage的缓存机制是如何实现的

1、开始自我介绍

2、根据简历了解个人研究方向的问题

3、说一个自己最近印象深刻的项目，然后根据做的项目说一下里面的关键技术点，

     最后引导出其中的安全问题，如何加密解密，数字签名、服务器配置

4、Objective-C的属性和实例变量的区别

5、UIView和CALayer的区别 还用过哪些Layer类(CATextLayer等)

6、Category如何扩展，有什么好处？为什么不能扩展属性？

7、tableView的优化原理，cell如何重用

1、HTTP协议原理，get和post的区别

2、Cookie是什么

3、对称加密和非对称加密的区别

4、关于 ReactNative

5、算法题：一个整数数组，判断是否有两个整数的和等于一个整数m，要求一次循环实现

6、对公司技术和业务的认识

1、不用临时变量怎么实现swap(a,b)，用加法或异或都可以

2、二维有序数组查找数字

3、亿级日志中，查找登录次数最多的10个用户

4、简述排序算法“快排”partion函数的原理，堆排序(不稳定)，归并排序、基数排序

iOS相关问题理解：

1、Objective-C中load方法和initialize方法有什么区别？各自执行时机和用途是什么？

没实现子类的方法会不会调用父类的？

2、对ARC和MRC的理解

3、UITableView的调优方案

4、MVC和MVVM的区别，各自的优缺点是什么？为啥要使用MVVM？

5、对Runtime的理解，方法的查找，消息的转发，对象的内存布局？

6、说说对Block的理解，如何解决循环引用的问题？

自我介绍后问了几个iOS相关的问题

1、说说对属性的理解，如何用于内存管理？

2、介绍一下iOS中的Block，有哪些类型的Block？

3、如何优化UITablView防止卡顿？(高频问题)

4、野指针是什么？iOS开发中什么情况下会有野指针，如何避免？

开始先笔试写算法

1、给一个字符串，如何判断它是否是合法的IP地址，例如:”192.168.1.1”就是合法的，方法思路不限，例如使用正则表达式

2、说说大数相加的思路，动手写代码实现

3、简述TCP建立和关闭连接握手的过程，前者为什么是三次，后者为啥是四次？

4、问题:假设10万条电话号码，如何通过输入电话号码的某一段内容，快速搜索出来。例如输入234，以下两个号码都会显示在

搜索框

    123456789000

    1888888823400

5、向Objective-C的数组中添加nil对象会有什么问题？

1、ARC的原理，在MRC和ARC下如何获取对象引用计数器

2、#include 与 #import区别

3、内存的大小弊端，int的高权位在哪一端？

4、TableView中Cell高度缓存方案，定高Cell和变高Cell如何处理？

5、int、chat、double类型的内存大小，sizeof使用

6、二维数据、字典及缓存数据模型问题

7、Objective-C的编译过程中有哪些流程要处理？

8、iOS中的数据持久化方式及使用？

9、const、static以及全局变量的区别？

const

1、const用来修饰右边的基本变量和指针变量

2、被修饰的变量只读，不能被修改

int const \*p //\*p 只读 p变量

Int \*const p // \*p变量 p只读

const int \* const p // p和\*p都只读

Int const \*const p // p和\*p都只读

//定义一个全局的只读变量

NSString \* const Kname = @’appKey’;

//static修饰后此全局变量只能在本文件访问

Static NSString \*const Key = “abc”;

Static

1、修饰局部变量

保证局部变量永远只初始化一次,在程序的运行过程中永远都只有一份内存，生命周期直到程序销毁，作用域不变

1. 修饰全局变量

使全局变量的作用域仅限于当前文件内部，即当前文件内部才能访问到该全局变量

extern

作用就是声明外部全局变量，这里特别注意的是extern只能声明不能用于实现

我们通常抽一类来管理一些全局变量

在.h文件中声明

extern NSString \*const name;

extern NSInteger const count;

在.m文件中实现: NSString \*const name = @’xxx’; NSInteger const count = 123;

1、获取对象的引用计数CFGetRetainCount 和 retainCount的用法，在哪些情况下返回的计数是不正确的

2、@property 中 weak、strong、copy、assign、retain、nonatomic等字段的含义和用法

3、关联对象内部原理，主对象如何指向它关联对象

4、简述Objective-C的Runtime对象模型和消息机制？

5、Objective-C 和 Java 的 RunTime有什么区别？

6、weak 的内部实现原理及在MRC下如何实现weak？

1、MVC具有什么样的优势？各个模块之前怎么通信？例如单击button后怎么通知Model

2、如何判断两个无限长度链表有没有交点？

3、SDWebImage的缓存机制是怎样的？如何设计图片缓存？

4、用GCD如何实现这个需求？A、B、C三个任务并发，完成后执行D

5、KVO、Notification、delegate各自的优缺点、效率和使用场景？

6、Objective-C中的copy方法

7、Runtime中，SEL 和 IMP的区别

8、autoreleasepool的使用场景和原理

9、Runloop的实现原理

10、Block为啥会有循环引用，如何解决？

11、如何手动通知KVO

12、UITableView的相关优化

13、NSOperation 和 GCD 的区别

14、CoreData的使用，如何处理多线程的问题

1、进程和线程的区别

2、TCP和UDP的区别

3、TCP流量控制

4、数组和链表的区别

5、Autoreleasepool什么时候释放？在什么场景下使用？

6、怎么判断某个Cell是否显示在屏幕上？

7、UIVIew的生命周期

8、如果页面A跳转到页面B，A的viewDidDisappear的方法和B的viewDidAppear方法哪个先调用

9、ARC的本质

10、Runloop的基本概念，它是怎么休眠的

11、如何找到字符串第一个不重复的

1、知道哪些设计模式？单例是为了处理什么问题使用的？单例和全局变量的区别

2、什么是元类？实际中会用元类做什么事情？

3、TCP和UDP的区别

4、HTTPS的工作原理，和HTTP的区别 RSA加密原理

5、Block底层原理，结构、内存以及需要注意的地方

6、一张png或jpg格式的图片渲染到页面上显示有哪些流程，png和jpg的有什么区别 png一定是无损的吗

7、Runtime相关

8、RunLoop的结构和循环流程，用RunLoop做过什么事情

9、多线程，NSOperationQueue和GCD的区别，用多线程做过什么事情？线程安全的问题。加锁有几种方式

10、NSURLSession 和 NSURLConnection 的区别

11、做过什么动画效果？用什么实现的？隐式动画和显示动画的区别

12、SDWebImage的框架结构，缓存机制

13、对组件化的了解，组件化是为了解决什么问题？

14、JSpatch原理

15、如何分层打印二叉树

1、MVC和MVVM的区别

2、如何编写单元测试？例如写了一个网络库，如何测试这个网络库，用例怎么写的更全面一些？

3、代码Git上上拉下来到生成.ipa都有哪些过程，期间都生成了什么文件

4、看过哪些框架、开源代码，有什么收获？

5、JSPatch是如何处理Block的？

6、如果没有Instruments，该如何检测内存泄漏，僵尸对象之类的问题？

1、单例和全局变量的区别

2、Block 和 Protocol的区别，Block是为了解决什么问题使用的

3、iOS的设计模式

4、冒泡、插入、快速排序的平均时间复杂度和最坏时间复杂度

5、堆排序有时间复杂度为0(n)的排序吗

6、读过的开源框架，读过的书（高频问题）

1、解释属性修饰关键词的作用(weak、strong、copy、readOnly、assign、nonatomic等)

2、线程和进程的区别，如何保证线程安全？

3、写算法输出 2～100的素数，如何进一步优化？

4、了解哪些设计模式？

5、MVC的设计模式的思想是什么？对比MVVM设计模式

6、堆和栈的区别，工程中哪些数据存储在堆中哪些在栈中？

7、iOS 中的NSCopying协议，copy和mutableCopy的区别？

8、有什么问题问面试官

1、iOS中的内存管理

2、iOS中开发中用过哪些测试性能工具  
3、二叉树的先序遍历，递归和非递归思路

4、写快速排序算法，并分析原理

5、HTTP为什么底层是TCP不是UDP

6、TCP是基于流式传输的，怎么设计协议

7、TCP为啥要进行三次握手

8、抓包的原理是什么

9、socket异常断开时，设计一个合理的重连机制

10、在10亿个数中如何快速找出最大的前100个数

1、解释属性修饰关键词的作用

2、项目中用过哪些设计模式

3、结合响应者链条和设计模式，解释事件怎样传递和处理

4、block、NSNotification、delegate、Observer比较

5、block为什么会导致循环引用，如何解决？

1、个人项目提问

2、tableView的性能优化

3、Cocoapods的实现思路

4、Runtime在特殊需求当中的运用

5、SDK的接口设计过程(一般设计为进入业务线提供一个注册方法，在需要的时候使用代理回调，向SDK里传递数据一般用单例模式，暴露相应的接口)

6、AFN实现思路，对源码的理解程度

7、项目总如何保证数据的安全性

8、快速排序的原理

9、C 语言中的strlen 和 sizeof的区别

1、数组中(无序的正整数数)如何找到第n个的数？

2、数组中 1 - 100 的有序数字如何找到第n大的数

3、对Objective-C的运行时和动态性特性的理解

4、Objective-C是如何找到一个方法的？

5、Objective-C是有没有重载函数？

6、MRC 和 ARC 下 内存管理的区别

7、autoreleasepool被释放了，里面的对象都会被释放吗？过程是怎样的？

1、++和Objective-C的内存管理比较，以及实现方式

2、++和Objective-C有哪些区别？

3、Objective-C的MRC和ARC的内存管理原理

4、两个数组，一个有100万的数据，另一个有1亿的数据，如何找出相同的数据，时间复杂度是什么？

5、Swift 和 Objective-C有哪些区别？

6、冒泡排序和快速排序的时间复杂度