一、自我介绍：

我叫\*\*，毕业于\*\*\*大学，此前一直就职于\*\*\*公司，担任iOS开发工程师一职，在职期间做过的项目种类繁多，有商场类，医疗类，旅游类，金融类等等一共十多个，其中有四个是自己独立完成的，其余的是团队合作完成的。

二、面试专业技能

1、熟悉开发环境，熟悉语言，大部分面试官会对debug能力比较感兴趣，都会问这个问题，我的回答是：

对于每个程序员来说，遇到bug是最头痛的事情，想着自己写的代码是如何漂亮，是如何思路清晰，程序怎么就给crash掉了的，那么我对于这样的问题，有我自己解决的一套，不会像普通程序员那样，遇到问题就急躁，不知所措。首先，我会先看下问题出错的原因，仔细查看reason，然后可以打一个全局断点，看断点指向的位置上下文代码的逻辑，一般情况下都可以很快找出问题所在，那么有的时候，我们会发现全局断点也找不到错误所在，那么我会根据reason找到出错的事件，一般情况下我们都是系统响应了某个事件才会crash的，我会根据事件逻辑相关的代码，仔细查看问题所在，若是碰到了bad\_access这样的问题，还可以打开僵尸调试来做处理。现在碰到了问题，想到的不再是代码明明是对的啊，而是想着我解决了一个问题，就是一次进步，如果不是碰到问题，进步的速度会变得很慢，有时候，很多新东西都是从bug中学到的。

注：有些面试官会问到自己是否会处理真机调试的crash日志，可以这样回答：

1、Xcode自己本身带有查看crash日志的功能，在device那里，我们可以根据日志的具体内容或是通过百度来找出crash的reason；

2、通过第三方的，虽然我没有用过，但是我知道可以实现这个功能，而且不会很麻烦，一个是听云，一个是我们熟悉的友盟，都可以获取这些crash log。

2、MRC/ARC，可以直接说自己没有实际MRC的产品，但是懂得大致的原理（可以说下黄金法则和自动释放池），但是目前主流的都是ARC，他本身就是比MRC方便，高效，在编译期间会自动判断在需要做release或者retain操作的地方加上内存管理的方法，这个也使得我们做开发的时候更加方便，快捷。

具体的内存管理语义呢（retain，copy，strong，weak，assign等等可以网上查资料，有很多）；

3、多线程：多线程分为四种：pthread, nsthread,GCD, nsopreation，由于前两者的复杂和不安全性，目前主流的是用的GCD（C的API）和nsopreation，那么GCD分任务和队列，它是才用block的方式，将任务加到队列中实现多线程，具体可以实现的种类有同步串行，同步并行，异步串行，异步并行，其中异步并行用的最多，当然GCD还有很多其他的用法，比如设置屏障，group，source等等，实现的功能繁多（要查的人可以查看这篇blog：<http://blog.devtang.com/blog/2012/02/22/use-gcd/> 和这篇：https://github.com/nixzhu/dev-blog/blob/master/2014-04-19-grand-central-dispatch-in-depth-part-1.md）

那么nsopreation是在GCD的基础上做了封装后OC的API，它采用对象的形式来创建多线程，其中nsopreation是一个抽象类，我们不可以直接使用，只能用到它的子类，系统已经帮我们设计了它的两个子类，一个是invocation（采用selector的形式），一个是block（采用block的形式），那么我们在通过子类创建人物后，再加载到queue上，系统自动帮我们分配多线程完成任务，同时我们也可以对不同线程设置依赖关系，功能同样很强大。

有的面试官会问你用到哪一种方式多一点，我个人回答的是nsopreation，回答如下：

他们两种都是主流的方式，我个人偏向nsopreation，主要还是看不同的情况，GCD是一种很直观的形式去创建多线程，在使用过程种如果创建线程不会多次使用，并且操作较为简单的话，我会选择GCD，大多数情况下需要重复创建子线程，那么我会选择nsopreation，因为他的重用性比较高，而且使用起来，我觉得更加适合面向对象编程的思想，在一些大的项目我更加偏爱nsopreation。

4、TCP/IP协议，socket相关：

我回答的要点：TCP/IP协议是传输层的协议族，他提供了数据传输的可靠性和可行性，让我们的数据可以在网络上传输；Http是应用层的协议，保证了数据传输的安全和数据可靠性，是对TCP/IP协议的再次封装，通过加头部和校验，让我们可以更方便的传输数据；

socket：是封装了TCP/IP协议的一组API，他让我们程序员更加方便的使用网络去达到我们想要的效果，传输数据，我们不需要针对复杂的网络协议做太多的了解，就可以实现C/S模式的一个通讯方式，它是一个长连接的过程，我们具体要做的事情是：create一个socket对象，监听IP地址和端口，readData&writeData，我们只需简单的几步就可以完成强大的功能。

有的面试官问我是怎么实现IM的，我说的是用公司底层的接口，自行更具客户需要写客户端，这样面试官就不会问底层的东西了。

如果面试官问道TCP/IP 等协议的原理，那只能说面试官不懂或者是问的东西跟他表达的不一样。我们可以这样回答，TCP/IP协议族本来就是一个非常复杂的协议，可以写三本书，说到原理还真说不出来，如果说对于一个前端的工程师还需要了解这么多的网络知识，可能对于一个程序员来说，要求太苛刻，如果真的是这样，那么我们是不是把后台的事情一起做了算了呢？这显然是不合理的，不过公司若真是有这样的需求，当然，我觉得我对网络的了解，做这样的项目也是可以的，去学习并很快用到项目上也是可以的，但是对于项目进度，肯定会在最开始立项的时候就说明，时间可能会稍微久，这个都是可以商量的。

5、设计模式：

所有的设计模式都是从面向对象编程的六个原则来的（<http://m.blog.csdn.net/blog/dragonseek24/13293947>，这个需要花时间去理解，不过现在面试可以强记下，吹逼用下），那么可以先说原则后，在谈设计模式，切记装逼的话：我们编程不是为了用某种设计原则而用某种设计原则，而是说我心中有了这样的思想，为了让代码变得好看，逻辑更加清楚，便于扩展和重构，我才这样编程，在我们编写的过程中是不知不觉就用到了一些设计模式，所以，以思想为主，编程为辅。然后设计的模式大家都懂，随便说下就行了。

注：工厂模式是将功能相同或者类似的方法或者类封装起来，通过抽象类的方式，给出最少的接口，实现最多的功能。

6、AF的使用以及原理：

见blog（2.0版本：<http://blog.cnbang.net/tech/2320/>；3.0版本：http://www.jianshu.com/p/0ab3ffa9c5f5）

6、FMDB和coreData的区别：

FMDB：第三方lib,是对sqlite的封装，存储结构为表结构，使用sql语句来做存储管理；

coreData为苹果原生的API，同样是封装了SQLite，但是存储的基本单元是对象；

个人认为FMDB使用更加方便和灵活，在项目构架阶段，设计好了表结构后，后期的增删改查都非常容易；

7、使用过的第三方lib:SD，AF，JSONModel，MJ，MBProgressHUD，SVProgressHUD，Masonry，YTK，YYKit（这个是最新的，估计很多人都听过，用的人就更少了，但是这个工具库在GitHub上已经引起了爆炸性轰动）；

8、runtime

（http://southpeak.github.io/blog/2014/11/03/objective-c-runtime-yun-xing-shi-zhi-san-:fang-fa-yu-xiao-xi-zhuan-fa/）

9、runloop

http://blog.ibireme.com/2015/05/18/runloop/

10、YTK介绍：

<https://github.com/yuantiku/YTKNetwork>

找到基础教程和高级教程；

11、YYKit介绍：

<https://github.com/ibireme/YYKit> 查看中文介绍；

其他的像内存优化啊，蓝牙啊，支付啊，都是按照老师那套来的，我也是根据老师的demo说的。

搭建框架不是指的UI而是指的你怎么去设计一个类，整体规划整个产品所要用到的所有类，从面相对象编程的六个原则来说就行了。

1. 版本控制：

目前国内主流的是SVN，那么主流的SVN客户端有莲花和CornerStone ，我们在客户端做的事情就是下载前一个版本，然后对code进行添加和修改，然后提交版本到服务器上即可，一半来说，一天至少提交两次，中午吃饭前一次，下午下班前一次，若加班比较晚，代码更新量较多，也要提交一次。提交的过程中如果碰到了代码冲突，需要我们自行解决冲突后再上传，这个事情是自己做的，不是组长或者其他人，自己写的代码，自行管理。

那么，目前国际主流的版本控制软件的趋势是git，这个我们一般没有接触过，不过国内也有免费的git服务器可以供我们大家做合作开发的时候使用，那就是oschina，我们可以在上面做一些开源的团队合作项目，或者自己的项目管理，那么客户端的话，可以说sourceTree，这个是比较流行的，我没有怎么用过，大家可以网上找些教程（比如：http://jingyan.baidu.com/article/c45ad29cdf274e051753e22c.html）这个就差不多可以应付面试官了，用git的时候主要是做分支管理，管理员管理冲突。

1. 内存优化：

很多面试官会问我么如何做项目中的内存优化的，其实内存优化的方法有很多，大多数情况下，可以从以下几个方面回答问题a、为什么会出现内存爆炸的情况：大多数是图片过多，第二个是在有些可以做重用处理的地方没有做重用处理；

b、出现了怎么办：我们可以在 receving memory这个方法里面去做一些对象置空处理，或者结合SDWebImage的思路去回答面试官，做好缓存和清空缓存处理，做好本地存储，尽量减少网络请求，或者在利用重用机制；

c、在有些时候，可能是多个重量级控件结合使用导致内存瞬间占用爆炸，比如UItableView上添加UIWebView，这样会导致内存过高，界面很卡，我们需要及时对UIWebView进行置空处理，以及尽量避免这样的事情发生，另外我们也可以使用MK代替UIWebView，这个也是一个方法；

3、JSON和XML区别：

JSON，轻量级，方便解析，可以随便解析任何一个我们想要的数据，节约流量；可扩展性没有XML高，没有XML直观

XML：没有什么事XML没法扩展的，解析方式有通过path和父节点，直观，但是多了很多标签，在数据较多时，会在一定程度上耗费更多的流量。

1. 动画：

每次面试官问我的时候，我都会回答我对动画没有很多的研究，大多数情况下，客户的动画效果我都是在code4APP上找到，因为那个上面的代码还是比较健壮的，而且便于使用，我自己知道的动画有UIView的简单动画，转场动画，弹簧动画，关键帧动画等，对于核心动画只有简单的了解，可能达不到公司的需求，但是如果公司需要自己设计出一定的动画效果，我认为通过对API的学习和我自己本身就有较好的数学基础，应该可以很快使用在项目上，如果公司在动画效果需求很强烈，或者是时间紧迫，那么我会在工作之余，晚上去学习，不会影响公司项目进度，而且我有一个同学做这个还比较在行，碰到问题找他处理，应该可以很快解决。

1. 装逼提示：

大家可以看下3D touch，这个功能还蛮有趣，而且很简单，看了一个demo就可以知道怎么做了，大家可以把这个跟一些面试官提下，以表现自己爱学习，在专研前言知识的心；

面试官有一些一般的，会问很多UI的问题，如果回答不出来，就说我觉得UI除了动画效果，都是差不多的，我现在更把心思放在业务逻辑上，这个方面处理好了，软件的结构会更加清楚，客户更深层次的体验才会更好；

1. Swift

对于swift，我是有很浓厚的兴趣，但是这个并不是说是某个知识点，简单的学习就可以差不多掌握了，需要我们自己系统的规划时间，安排学习计划才能学好的，目前我对它也是了解阶段，那么之前在一家外包公司，实在是挤不出更多的时间系统学习，而且swift目前也是出于一个大神为我们踩坑的阶段，这门语言还在不停的更新，所以之前没有敢说下定决心去学习，现在，swift开源了，我们觉得真是很好，在下个阶段我就会在工作之余时间开始学习swift，并且在前段时间的WWDC上，我也有看了一点视频，里面出的新的思想，面向协议编程，我很有兴趣，而且也用在封装YTK中了，对于前沿的咨询还是在不断的了解的。

1. 项目阶段的问题：

有些面试官会问你在做项目的过程中遇到过最大的问题是什么，其实很多人会不记得自己有碰到什么大问题，因为编程少而且都解决了，我是这样回答的：

其实每个程序员都会花大量时间在解决bug上，我觉得我碰到问题就是会看到很多奇奇怪怪的bug，但是我同样要感谢这些bug，因为是他们让我学到很多东西，锻炼我解决问题的能力，以及通过查找问题的过程中学到更多新的知识；那么对于具体的某一个问题我可能一时间也举不出例子出来，那么面试官你可以举个例子，我们可以讨论下，看有没有什么思路或者解决办法的，如果我不会的话，我也向您学习下。

1. 异常情况

我在给面试官看我的APP时，虽然是从APPstore上下载的，但是也出现了闪退的情况，我会这样说，哎，这手机被我测试出问题了，什么程序都会出现闪退的情况，奇怪。

1. 学校问题

有个公司的老板跟我说道我们学校不是很好时，我是这样回答他的：

个人不应该由于他所处的集体来判断，虽然环境会影响一个人，但是并不仅仅是大学这四年的环境，我认为我们学校很好，是她培养了我四年，让我有了今天的成就，面试官看了我的简历今天面试我，不也是对我能力的一个认可吗？况且我们学校也出过一些名人，比如许家印，所以，我觉得公司招人还是看能力，并不是他从前是谁，哪个学校，而是他现在是谁，有和能力。