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？

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

10. Socket的实现原理及Socket之间是如何通信的

11. http协议的实现