<http://www.jianshu.com/p/1b2f63a45476>

待学习的知识点

1.双轴快排

2.秒杀的实现

3.线程同步的实现方式

4.网络各个协议

5.索引

MySQL数据库支持多种索引类型，如BTree索引，哈希索引，全文索引等等

# Java部分

## Java基础

### 线程池（与六部分一起看）

1. 线程池的实现原理

线程是稀缺资源，如果被无限制的创建，不仅会消耗系统资源，还会降低系统的稳定性，合理的使用线程池对线程进行统一分配、调优和监控，有以下好处：

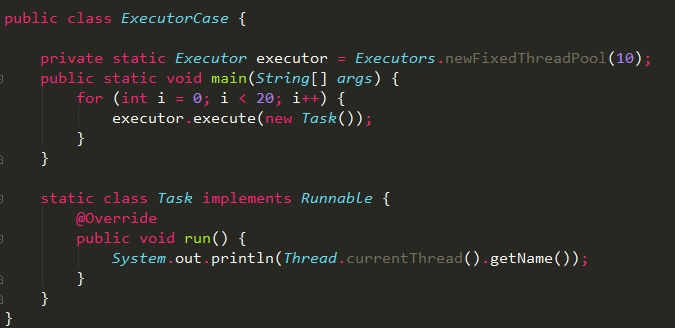
1、降低资源消耗；

2、提高响应速度；

3、提高线程的可管理性。

Java1.5中引入的Executor框架把任务的提交和执行进行解耦，只需要定义好任务，然后提交给线程池，而不用关心该任务是如何执行、被哪个线程执行，以及什么时候执行。

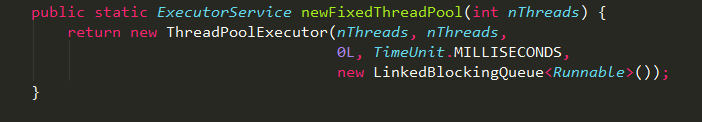
**示例**



1、Executors.newFixedThreadPool(10)初始化一个包含10个线程的线程池executor；  
2、通过executor.execute方法提交20个任务，每个任务打印当前的线程名；  
3、负责执行任务的线程的生命周期都由Executor框架进行管理；

**ThreadPoolExecutor**

Executors是java线程池的工厂类，通过它可以快速初始化一个符合业务需求的线程池，如Executors.newFixedThreadPool方法可以生成一个拥有固定线程数的线程池。



其本质是通过不同的参数初始化一个ThreadPoolExecutor对象，具体参数描述如下：

public ThreadPoolExecutor(

               int maximumPoolSize,

               long keepAliveTime,

               TimeUnit unit,

               BlockingQueue<Runnable> workQueue,

               ThreadFactory threadFactory,

               RejectedExecutionHandler handler

){...}

**corePoolSize**

线程池中的核心线程数，当提交一个任务时，线程池创建一个新线程执行任务，直到当前线程数等于corePoolSize；如果当前线程数为corePoolSize，继续提交的任务被保存到阻塞队列中，等待被执行；如果执行了线程池的prestartAllCoreThreads()方法，线程池会提前创建并启动所有核心线程。

**maximumPoolSize**

线程池中允许的最大线程数。如果当前阻塞队列满了，且继续提交任务，则创建新的线程执行任务，前提是当前线程数小于maximumPoolSize；

**keepAliveTime**

线程空闲时的存活时间，即当线程没有任务执行时，继续存活的时间；默认情况下，该参数只在线程数大于corePoolSize时才有用；

**unit**

keepAliveTime的单位；

**workQueue**

用来保存等待被执行的任务的阻塞队列，且任务必须实现Runable接口，在JDK中提供了如下阻塞队列：

1、ArrayBlockingQueue：基于数组结构的有界阻塞队列，按FIFO排序任务；

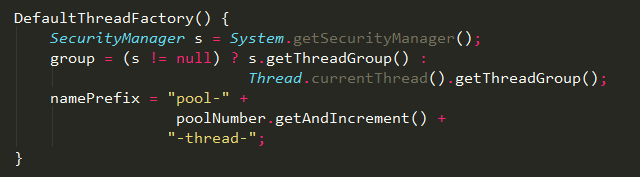
2、LinkedBlockingQuene：基于链表结构的阻塞队列，按FIFO排序任务，吞吐量通常要高于ArrayBlockingQuene；

3、SynchronousQuene：一个不存储元素的阻塞队列，每个插入操作必须等到另一个线程调用移除操作，否则插入操作一直处于阻塞状态，吞吐量通常要高于LinkedBlockingQuene；

4、priorityBlockingQuene：具有优先级的无界阻塞队列；

**threadFactory**

创建线程的工厂，通过自定义的线程工厂可以给每个新建的线程设置一个具有识别度的线程名。



**handler**

线程池的饱和策略，当阻塞队列满了，且没有空闲的工作线程，如果继续提交任务，必须采取一种策略处理该任务，线程池提供了4种策略：

1、AbortPolicy：直接抛出异常，默认策略；

2、CallerRunsPolicy：用调用者所在的线程来执行任务；

3、DiscardOldestPolicy：丢弃阻塞队列中靠最前的任务，并执行当前任务；

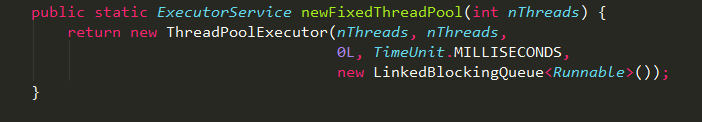
4、DiscardPolicy：直接丢弃任务；

当然也可以根据应用场景实现RejectedExecutionHandler接口，自定义饱和策略，如记录日志或持久化存储不能处理的任务。

**Exectors**

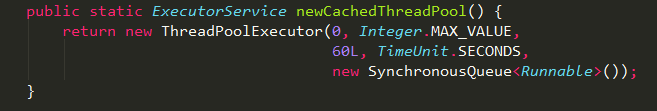
Exectors工厂类提供了线程池的初始化接口，主要有如下几种：

**newFixedThreadPool**



初始化一个指定线程数的线程池，其中corePoolSize == maximumPoolSize，使用LinkedBlockingQuene作为阻塞队列，不过当线程池没有可执行任务时，也不会释放线程。

**newCachedThreadPool**

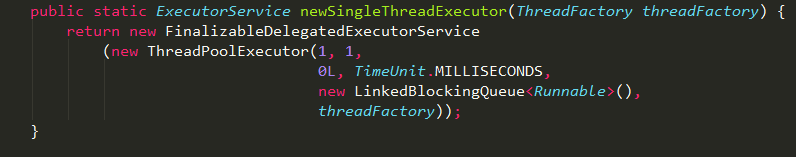


1、初始化一个可以缓存线程的线程池，默认缓存60s，线程池的线程数可达到Integer.MAX\_VALUE，即2147483647，内部使用SynchronousQueue作为阻塞队列；

2、和newFixedThreadPool创建的线程池不同，newCachedThreadPool在没有任务执行时，当线程的空闲时间超过keepAliveTime，会自动释放线程资源，当提交新任务时，如果没有空闲线程，则创建新线程执行任务，会导致一定的系统开销；

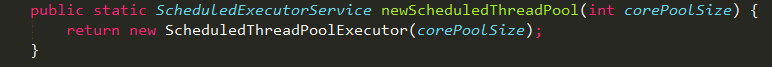
所以，使用该线程池时，一定要注意控制并发的任务数，否则创建大量的线程可能导致严重的性能问题。

**newSingleThreadExecutor**



初始化的线程池中只有一个线程，如果该线程异常结束，会重新创建一个新的线程继续执行任务，唯一的线程可以保证所提交任务的顺序执行，内部使用LinkedBlockingQueue作为阻塞队列。

**newScheduledThreadPool**

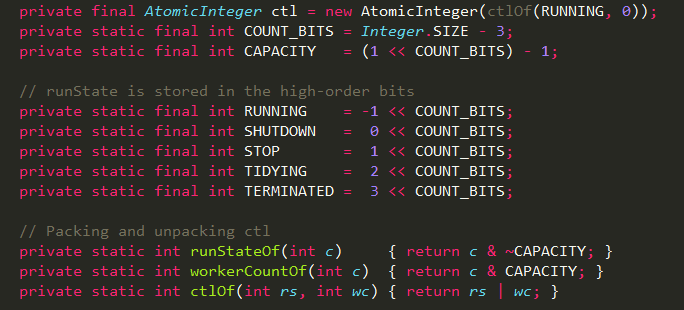


初始化的线程池可以在指定的时间内周期性的执行所提交的任务，在实际的业务场景中可以使用该线程池定期的同步数据。

实现原理

除了newScheduledThreadPool的内部实现特殊一点之外，其它几个线程池都是基于ThreadPoolExecutor类实现的。

**线程池内部状态**



其中AtomicInteger变量ctl的功能非常强大：利用低29位表示线程池中线程数，通过高3位表示线程池的运行状态：

1、RUNNING：-1 << COUNT\_BITS，即高3位为111，该状态的线程池会接收新任务，并处理阻塞队列中的任务；

2、SHUTDOWN： 0 << COUNT\_BITS，即高3位为000，该状态的线程池不会接收新任务，但会处理阻塞队列中的任务；

3、STOP ： 1 << COUNT\_BITS，即高3位为001，该状态的线程不会接收新任务，也不会处理阻塞队列中的任务，而且会中断正在运行的任务；

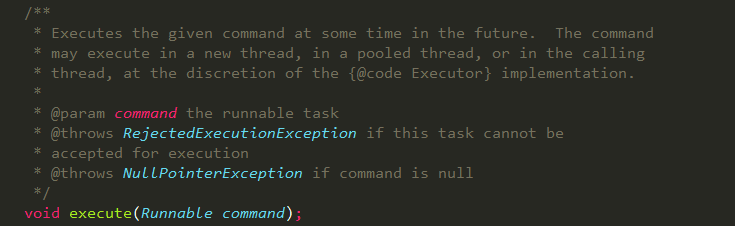
4、TIDYING ： 2 << COUNT\_BITS，即高3位为010；

5、TERMINATED： 3 << COUNT\_BITS，即高3位为011；

任务提交

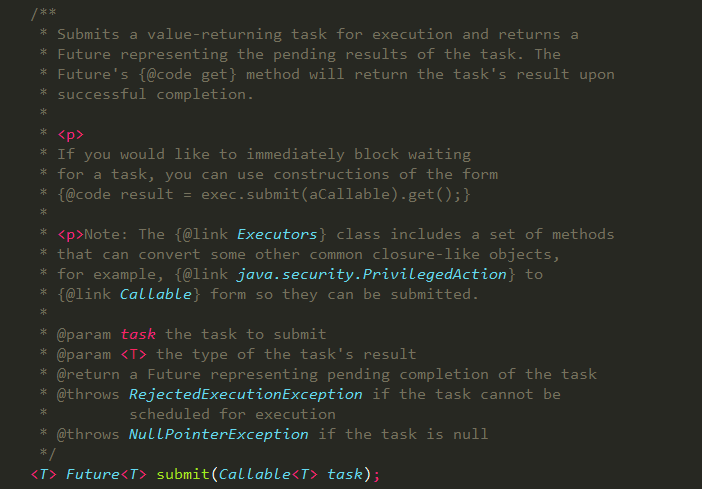
线程池框架提供了两种方式提交任务，根据不同的业务需求选择不同的方式。

**Executor.execute()**



通过Executor.execute()方法提交的任务，必须实现Runnable接口，该方式提交的任务不能获取返回值，因此无法判断任务是否执行成功。

**ExecutorService.submit()**

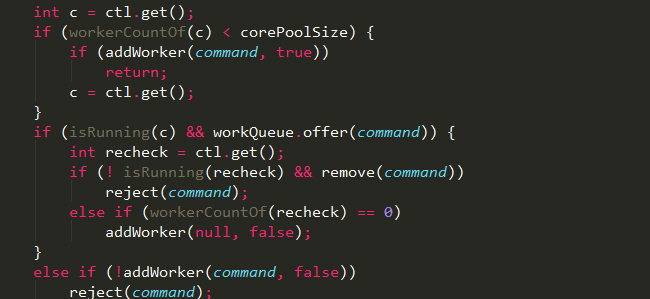


通过ExecutorService.submit()方法提交的任务，可以获取任务执行完的返回值。

**任务执行**

当向线程池中提交一个任务，线程池会如何处理该任务？

**execute实现**



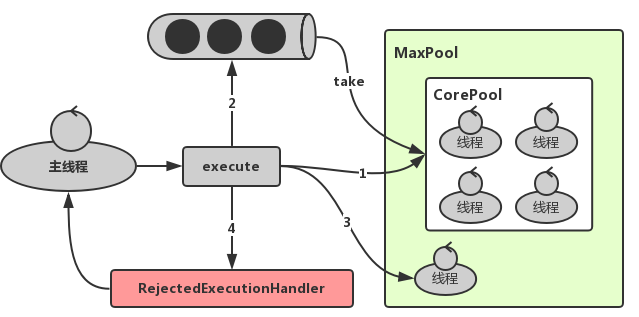
具体的执行流程如下：

1、workerCountOf方法根据ctl的低29位，得到线程池的当前线程数，如果线程数小于corePoolSize，则执行addWorker方法创建新的线程执行任务；否则执行步骤（2）；

2、如果线程池处于RUNNING状态，且把提交的任务成功放入阻塞队列中，则执行步骤（3），否则执行步骤（4）；

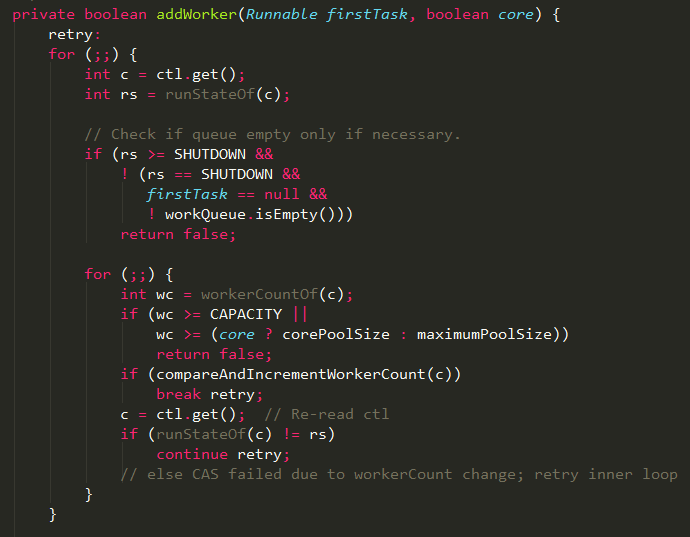
3、再次检查线程池的状态，如果线程池没有RUNNING，且成功从阻塞队列中删除任务，则执行reject方法处理任务；

4、执行addWorker方法创建新的线程执行任务，如果addWoker执行失败，则执行reject方法处理任务；



**addWorker实现**

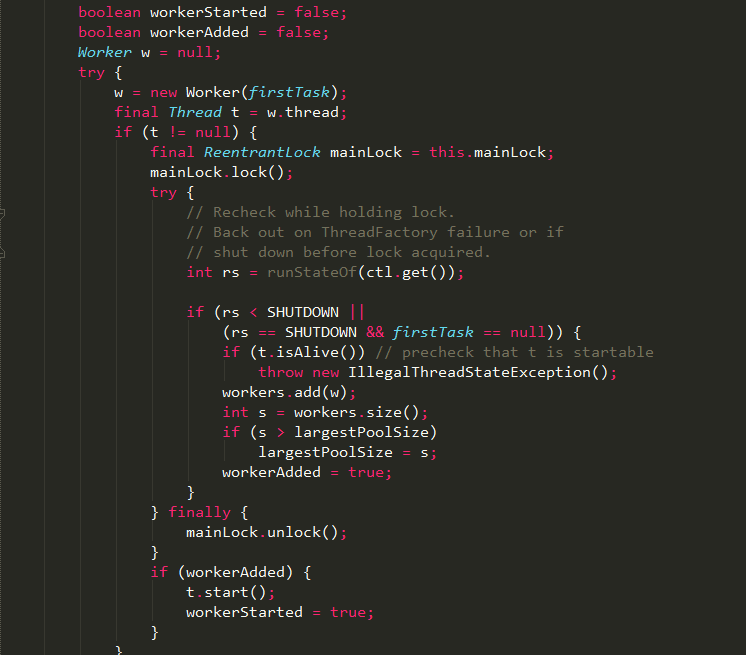
从方法execute的实现可以看出：addWorker主要负责创建新的线程并执行任务，代码实现如下：



这只是addWoker方法实现的前半部分：

1、判断线程池的状态，如果线程池的状态值大于或等SHUTDOWN，则不处理提交的任务，直接返回；

2、通过参数core判断当前需要创建的线程是否为核心线程，如果core为true，且当前线程数小于corePoolSize，则跳出循环，开始创建新的线程，具体实现如下：

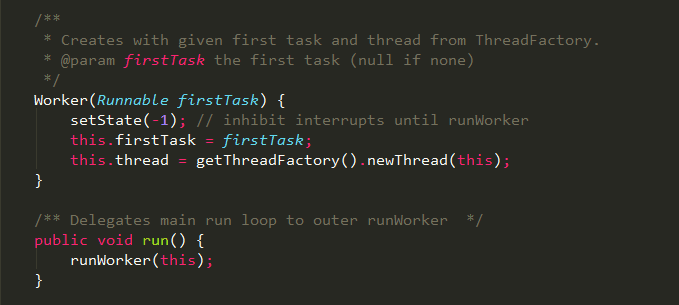


线程池的工作线程通过Woker类实现，在ReentrantLock锁的保证下，把Woker实例插入到HashSet后，并启动Woker中的线程，其中Worker类设计如下：

1、继承了AQS类，可以方便的实现工作线程的中止操作；

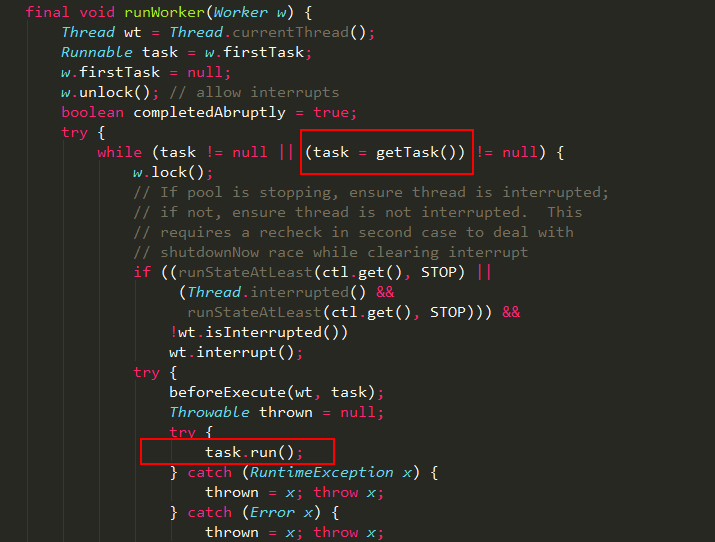
2、实现了Runnable接口，可以将自身作为一个任务在工作线程中执行；

3、当前提交的任务firstTask作为参数传入Worker的构造方法；



从Woker类的构造方法实现可以发现：线程工厂在创建线程thread时，将Woker实例本身this作为参数传入，当执行start方法启动线程thread时，本质是执行了Worker的runWorker方法。

**runWorker实现**



runWorker方法是线程池的核心：

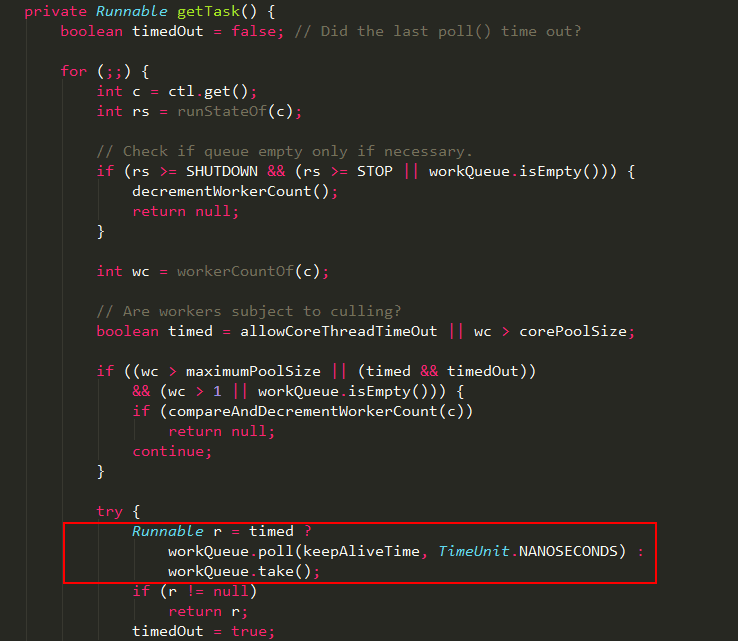
1、线程启动之后，通过unlock方法释放锁，设置AQS的state为0，表示运行中断；

2、获取第一个任务firstTask，执行任务的run方法，不过在执行任务之前，会进行加锁操作，任务执行完会释放锁；

3、在执行任务的前后，可以根据业务场景自定义beforeExecute和afterExecute方法；

4、firstTask执行完成之后，通过getTask方法从阻塞队列中获取等待的任务，如果队列中没有任务，getTask方法会被阻塞并挂起，不会占用cpu资源；

**getTask实现**



整个getTask操作在自旋下完成：

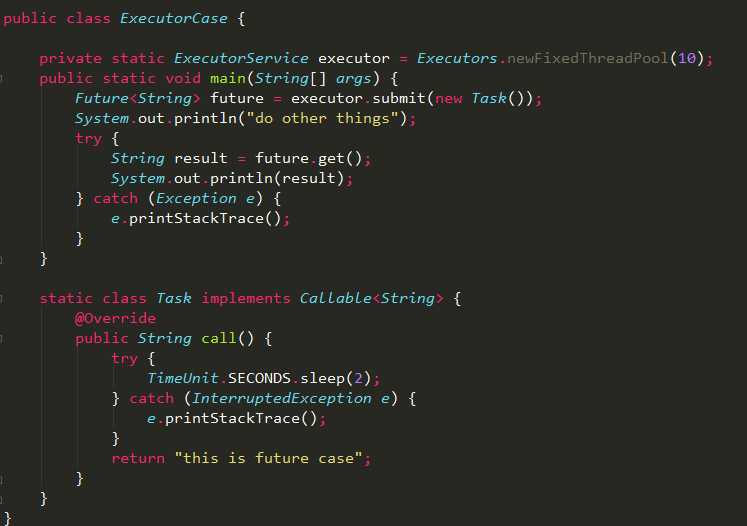
1、workQueue.take：如果阻塞队列为空，当前线程会被挂起等待；当队列中有任务加入时，线程被唤醒，take方法返回任务，并执行；

2、workQueue.poll：如果在keepAliveTime时间内，阻塞队列还是没有任务，则返回null；

所以，线程池中实现的线程可以一直执行由用户提交的任务。

**Future和Callable实现**

通过ExecutorService.submit()方法提交的任务，可以获取任务执行完的返回值。



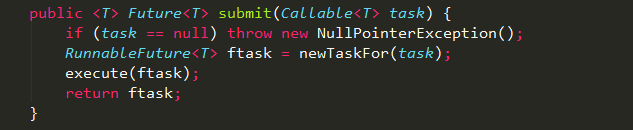
在实际业务场景中，Future和Callable基本是成对出现的，Callable负责产生结果，Future负责获取结果。

1、Callable接口类似于Runnable，只是Runnable没有返回值。

2、Callable任务除了返回正常结果之外，如果发生异常，该异常也会被返回，即Future可以拿到异步执行任务各种结果；

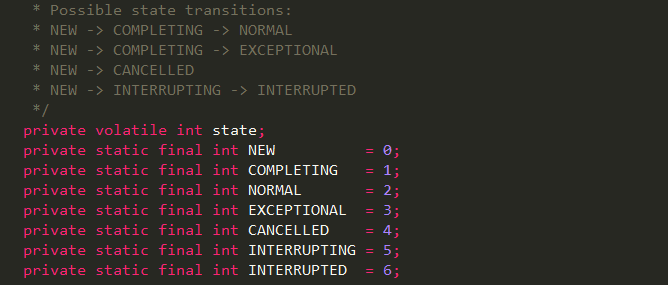
3、Future.get方法会导致主线程阻塞，直到Callable任务执行完成；

**submit实现**



通过submit方法提交的Callable任务会被封装成了一个FutureTask对象。

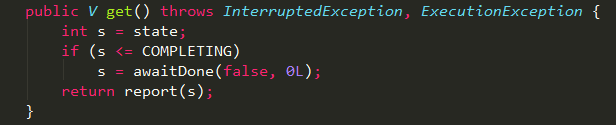
**FutureTask**



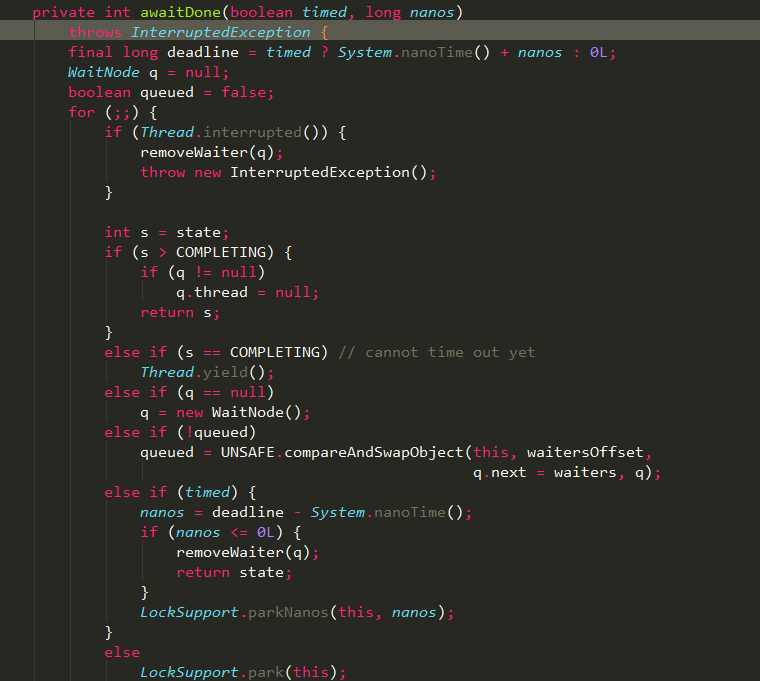
1、FutureTask在不同阶段拥有不同的状态state，初始化为NEW；

2、FutureTask类实现了Runnable接口，这样就可以通过Executor.execute方法提交FutureTask到线程池中等待被执行，最终执行的是FutureTask的run方法；

**FutureTask.get实现**



内部通过awaitDone方法对主线程进行阻塞，具体实现如下：



1、如果主线程被中断，则抛出中断异常；

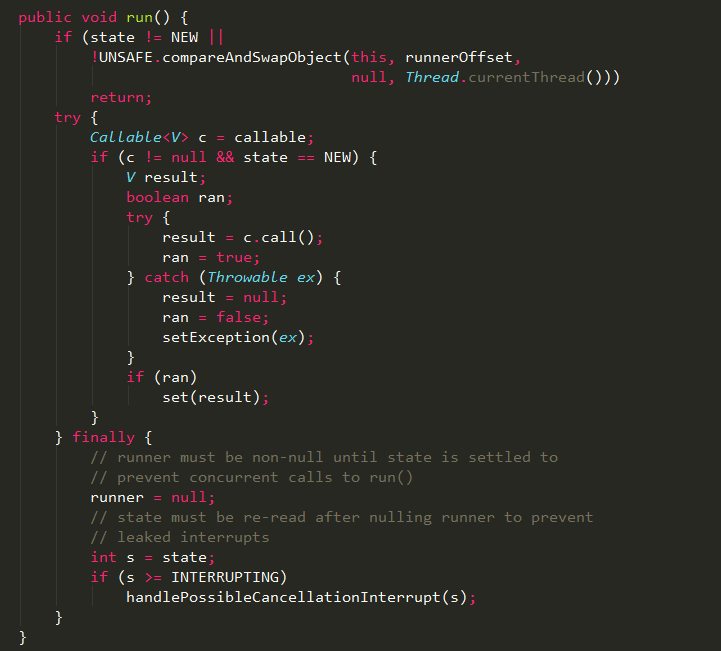
2、判断FutureTask当前的state，如果大于COMPLETING，说明任务已经执行完成，则直接返回；

3、如果当前state等于COMPLETING，说明任务已经执行完，这时主线程只需通过yield方法让出cpu资源，等待state变成NORMAL；

4、通过WaitNode类封装当前线程，并通过UNSAFE添加到waiters链表；

5、最终通过LockSupport的park或parkNanos挂起线程；

**FutureTask.run实现**



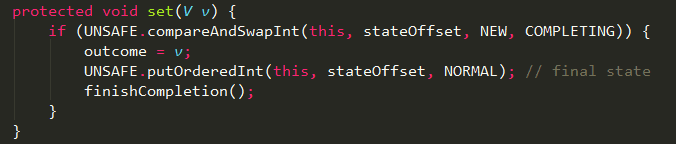
FutureTask.run方法是在线程池中被执行的，而非主线程

1、通过执行Callable任务的call方法；

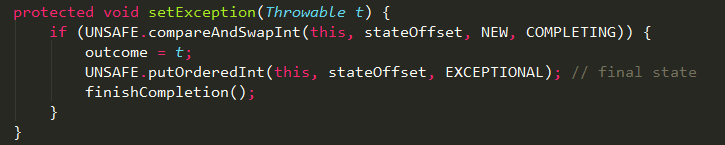
2、如果call执行成功，则通过set方法保存结果；

3、如果call执行有异常，则通过setException保存异常；

**set**

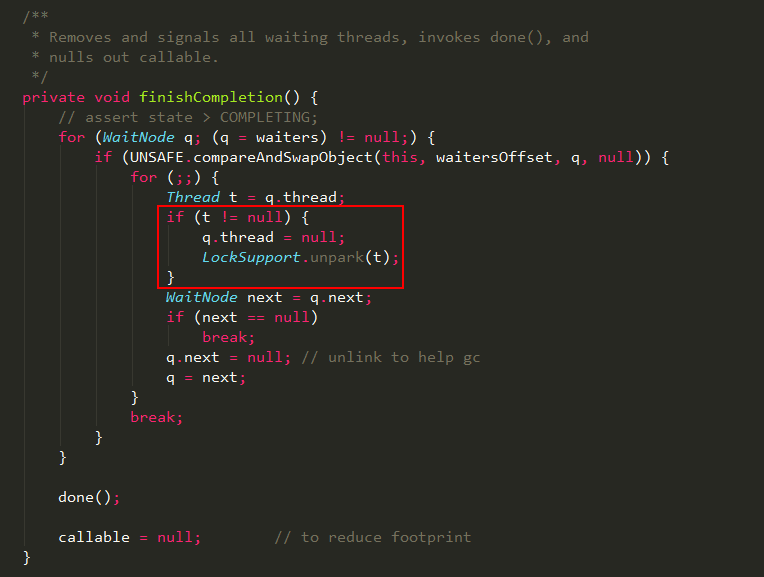


**setException**



set和setException方法中，都会通过UnSAFE修改FutureTask的状态，并执行finishCompletion方法通知主线程任务已经执行完成；

**finishCompletion**



1、执行FutureTask类的get方法时，会把主线程封装成WaitNode节点并保存在waiters链表中；

2、FutureTask任务执行完成后，通过UNSAFE设置waiters的值，并通过LockSupport类unpark方法唤醒主线程；

为了更好的控制多线程，JDK提供了一套线程框架Executor，帮助开发人员有效进行线程控制，他们都在java.util.concurrent包中。是JDK并发包的核心。非常重要的类：Executors。它是一个线程工厂，可以创建特定功能的线程。

Executors 创建线程池的方法：

**newFixedThreadPool()**方法，返回一个固定数量的线程池，该方法的线程数始终不变，当有一个任务提交时，若线程中空闲，则立即执行，若没有，则会被暂缓在一个任务队列中等待有空闲的线程去执行。

**newSingleThreadExecutor()**方法，创建一个线程池。若空闲，则执行，若没有空闲线程则暂缓在任务队列中。

**newCachedThreadPool()**方法，返回可根据实际情况调整线程个数的线程池。不限制最大线程数量，若用空闲的线程，则执行任务，若无任务则不创建线程。并且每一个空闲线程会在60秒后自动回收。

**newScheduledThreadPool()**方法，返回**SchededExecutorService**对象。 但该方法可以指定线程数量。 这里面的每一个线程都可以实现定时器的功能。

以上四个方法底层都是调用ThreadPoolExecutor的构造器来创建线程池。

public ThreadPoolExecutor(

               int maximumPoolSize,

               long keepAliveTime,

               TimeUnit unit,

               BlockingQueue<Runnable> workQueue,

               ThreadFactory threadFactory,

               RejectedExecutionHandler handler

){...}

（2） 线程池中的coreNum、maxNum和largestNum

（3） 在不同的业务场景中，线程池参数如何设置

### 锁的实现

线程安全的概念：当多个线程访问同一个类（或属性或方法）的时候，所得到的执行效果都是正确的，就说这个类（或属性或方法）是线程安全的。

锁竞争的基本概念：当线程要执行临界区（也称互斥区）的时候，要首先拿到临界区的锁对象，然后才可以执行；多个线程同时访问一个临界区的时候，就会发生锁竞争的情况。发生锁竞争是非常耗资源的（并发量很大的时候，甚至直接当机），应该尽量避免。

synchronized概念：同步，只有共享资源才需要同步。

异步概念：相互不影响。

脏读：写的时候有读的操作，就会产生脏读。（读操作和写操作都需要上锁同步）

1. synchronized和ReentrantLock实现原理，与异同点。

java在编写多线程程序时，为了保证线程安全，需要对数据同步，经常用到两种同步方式就是Synchronized和重入锁ReentrantLock。

**相似点：**

这两种同步方式有很多相似之处，它们都是加锁方式同步，而且都是阻塞式的同步，也就是说当如果一个线程获得了对象锁，进入了同步块，其他访问该同步块的线程都必须阻塞在同步块外面等待，而进行线程阻塞和唤醒的代价是比较高的（操作系统需要在用户态与内核态之间来回切换，代价很高，不过可以通过对锁优化进行改善）。

**区别：**

这两种方式最大区别就是对于Synchronized来说，它是java语言的关键字，是原生语法层面的互斥，需要jvm实现。而ReentrantLock它是JDK 1.5之后提供的API层面的互斥锁，需要lock()和unlock()方法配合try/finally语句块来完成。

**Synchronized**

Synchronized进过编译，会在同步块的前后分别形成monitorenter和monitorexit这个两个字节码指令。在执行monitorenter指令时，首先要尝试获取对象锁。如果这个对象没被锁定，或者当前线程已经拥有了那个对象锁，把锁的计数器加1，相应的，在执行monitorexit指令时会将锁计数器就减1，当计数器为0时，锁就被释放了。如果获取对象锁失败，那当前线程就要阻塞，直到对象锁被另一个线程释放为止。

**ReentrantLock**

由于ReentrantLock是java.util.concurrent包下提供的一套**互斥锁**，相比synchronized，ReentrantLock类提供了一些高级功能，主要有以下3项：

1.等待可中断。持有锁的线程长期不释放的时候，正在等待的线程可以选择放弃等待，这相当于synchronized来说可以避免出现死锁的情况。

2.公平锁。多个线程等待同一个锁时，必须按照申请锁的时间顺序获得锁，synchronized锁非公平锁，ReentrantLock默认的构造函数是创建的非公平锁，可以通过参数true设为公平锁，但公平锁表现的性能不是很好。

3.锁绑定多个条件，一个ReentrantLock对象可以同时绑定对个对象。

1. 读写锁
2. volatile关键字作用、实现原理
3. 锁的可重入性（详见：http://www.jianshu.com/p/007bd7029faf）
4. 对象监视器

并发编程中有三个概念：原子性、可见性、有序性。在这三个特性中，volatile关键字均可以发挥一些作用。

原子性：如转账。

可见性：

有序性：

1. Object.wait/notify实现机制
2. Unsafe类
3. 死锁和活锁
4. 并发相关面试题（http://www.cnblogs.com/binyue/p/4369985.html）

### ConcurrentHashMap

1. HashMap实现原理
2. ConcurrentHashMap作用、实现原理、扩容实现、其size方法的实现，1.7与1.8的不同实现

### LinkedHashMap

1. java LRU的缓存实现原理

最近最少使用缓存的回收（LRU）

怎么实现一个最近最少使用（LRU）的缓存。缓存可以通过哈希表来实现，然而为这个缓存增加大小限制会变成另一个有意思的问题。

为了实现缓存回收，我们需要很容易做到：

1.查询出最近最晚使用的项 2.给最近使用的项做一个标记

链表可以实现这两个操作，检测最近最少使用的项只需要返回链表的尾部。标记一项为最近使用的项只需要从当前位置移除，然后将该项放置到头部。 比较困难的事情是怎么快速的在链表中找到该项。

哈希表的帮助

看一下我们工具箱中的数据结构，哈希表可以在(消耗)常量的时间内索引到某个对象。如果我们创建一个形如Key-链表结点的哈希表，我们就能够在常量时间内找到最近使用的结点。 更甚的是，我们也能够在常量时间内判断结点是否存在（或不存在）；找到这个结点后，我们就能将这个结点移动到链表的最前端，标记为最近使用的项了。

Java的捷径

据我所知，很少有一种编程语言的标准库中有通用的数据结构能提供上述功能。这是一种混合的数据结构，我们需要在哈希表的基础上建立一个链表。 但是Java已经为我们提供了这种形式的数据结构LinkedHashMap！它甚至提供可覆盖回收策略的方法（见removeEldestEntry的文档）。 唯一需要我们注意的事情是，该链表的顺序是插入的顺序，而不是访问的顺序。但是，有一个构造函数提供了一个选项，可以使用访问的顺序。

该方法是最简便的实现方法。

代码如下：

|  |
| --- |
| **public** **class** LRUCache<K, V> **extends** LinkedHashMap<K, V> {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **private** **int** cacheSize;  **public** LRUCache(**int** cacheSize) {  **super**(16, 0.75f, **true**);  **this**.cacheSize = cacheSize;  }  **protected** **boolean** removeEldestEntry(Map.Entry<K, V> eldest) {  **return** size() >= cacheSize;  }  } |

另一种实现方法（不适用LinkedHashMap）如下：

|  |
| --- |
| **public** **class** LRUCache2 {  /\*双向链表\*/  **class** Node {  Node pre;  Node next;  Integer key;  Integer val;  Node(Integer k, Integer v) {  key = k;  val = v;  }  }  Map<Integer, Node> map = **new** HashMap<>();  Node head; // head是链表的最年老的节点  Node tail; // tail是链表的最年轻的节点  **int** cap;  **public** LRUCache2(**int** capacity) {  cap = capacity;  head = **new** Node(**null**, **null**);  tail = **new** Node(**null**, **null**);  head.next = tail;  tail.pre = head;  }  **public** **int** get(**int** key) {  Node n = map.get(key);  **if** (n != **null**) {  // 把n从链表中摘下来  n.pre.next = n.next;  n.next.pre = n.pre;  // 把n放到链表的尾部  appendTail(n);  **return** n.val;  }  **return** -1;  }  **public** **void** set(**int** key, **int** value) {  Node n = map.get(key);  **if** (n != **null**) {  n.val = value;  map.put(key, n);  // 把n从链表中摘下来  n.pre.next = n.next;  n.next.pre = n.pre;  // 把n放到链表的尾部  appendTail(n);  **return**;  }  **if** (map.size() == cap) {  Node tmp = head.next;  // 把tmp从链表上摘下来  head.next = head.next.next;  head.next.pre = head;  // 把tmp删掉  map.remove(tmp.key);  }  // n为空的时候  n = **new** Node(key, value);  appendTail(n);  map.put(key, n);  }  **private** **void** appendTail(Node n) {  n.next = tail;  n.pre = tail.pre;  tail.pre.next = n;  tail.pre = n;  }  } |

### 集合框架

HashMap和TreeMap区别

**定义：**

Java中的集合类：是一种工具类，就像是容器，存储任意数量的具有共同属性的对象。

**作用：**

集合可以：

1、在类的内部，对数据进行组织；

2、简单而快速的搜索大数量的条目；

3、有的集合接口，提供了一系列排列有序的元素，并且可以在序列中间快速的插入或者删除元素；

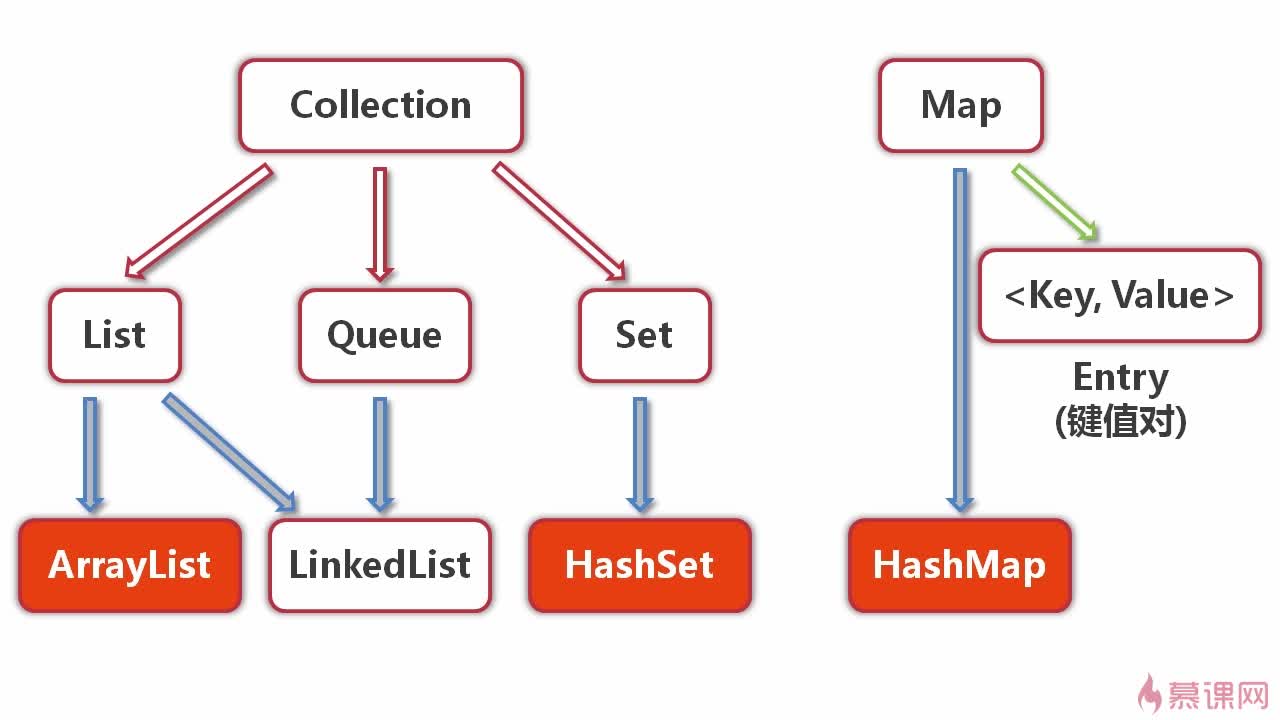
4、有的集合接口，提供了映射关系，可以通过关键字（key）快速的查找到对应的唯一对象，而这个关键字可以是任意类型。

**与数组的对比：**

1、数组的长度固定，集合长度可变；

2、数组只能通过下标访问元素，下标类型只能是数字型，而有的集合可以通过任意类型查找所映射的具体对象。

**Java集合框架的体系结构：**



**Collection接口**

1、是List、Set和Queue接口的父接口；

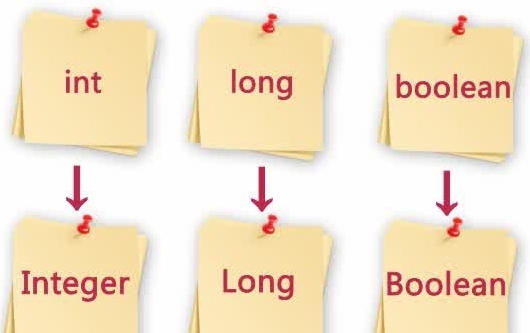
2、定义了可用于操作List、Set和Queue的方法——增删改查；

**List接口及其实现类——ArrayList**

* List（序列），元素有序，并且可重复；
* List可以精确控制元素的插入位置，或删除指定位置的元素；
* ArrayList——数组序列，是List的一个重要实现类；
* ArrayList底层是由数组实现的

泛型集合中，不能添加泛型规定的类型及其子类型以外的对象，否则会报错。

泛型集合中的限定类型，不能使用基本类型，必须用基本类型的包装类。



**Set接口及其实现类——HashSet**

* Set（集），元素无序，并且不可以重复；
* HashSet——哈希集，是Set的一个重要实现类；

**Map接口**

* Map提供了一种映射关系，元素是以键值对（key-value）的形式存储的，能根据key快速查找value；
* Map中的键值对以Entry类型的对象实例形式存在；
* key值不能重复，value值可以重复；
* key对value是多（一）对一的关系；
* Map接口提供了返回key值集合、value值集合、Entry值集合，的方法；
* Map支持泛型，形式如：Map<K,V>

**HashMap类**

* HashMap是Map的一个重要实现类，也是最常用的，基于哈希表实现；
* HashMap中的Entry对象是无序排列的；
* Key值和value值都可以为null，但是一个HashMap只能有一个Key值为null的映射（Key值不可重复）；

### 枚举

### 正则表达式

### 某些JDK版本的新特性

1. jdk8的新特性

（详见：https://www.ibm.com/developerworks/cn/java/j-lo-jdk8newfeature/index.html）

1. **函数式接口**

Java 8 引入的一个核心概念是函数式接口（Functional Interfaces）。通过在接口里面添加一个抽象方法，这些方法可以直接从接口中运行。如果一个接口定义个唯一一个抽象方法，那么这个接口就成为函数式接口。同时，引入了一个新的注解：@FunctionalInterface。可以把他它放在一个接口前，表示这个接口是一个函数式接口。这个注解是非必须的，只要接口只包含一个方法的接口，虚拟机会自动判断，不过最好在接口上使用注解 @FunctionalInterface 进行声明。在接口中添加了 @FunctionalInterface 的接口，只允许有一个抽象方法，否则编译器也会报错。

java.lang.Runnable 就是一个函数式接口。

|  |
| --- |
| @FunctionalInterface  **public** **interface** Runnable {  **public** **abstract** **void** run();  } |

1. **Lambda 表达式**

函数式接口的重要属性是：我们能够使用 Lambda 实例化它们，Lambda 表达式让你能够将函数作为方法参数，或者将代码作为数据对待。Lambda 表达式的引入给开发者带来了不少优点：在 Java 8 之前，匿名内部类，监听器和事件处理器的使用都显得很冗长，代码可读性很差，Lambda 表达式的应用则使代码变得更加紧凑，可读性增强；Lambda 表达式使并行操作大集合变得很方便，可以充分发挥多核 CPU 的优势，更易于为多核处理器编写代码；

Lambda 表达式由三个部分组成：第一部分为一个括号内用逗号分隔的形式参数，参数是函数式接口里面方法的参数；第二部分为一个箭头符号：->；第三部分为方法体，可以是表达式和代码块。语法如下：

1. 方法体为表达式，该表达式的值作为返回值返回。

|  |
| --- |
| (parameters) -> expression |

1. 方法体为代码块，必须用 {} 来包裹起来，且需要一个 return 返回值，但若函数式接口里面方法返回值是 void，则无需返回值。

|  |
| --- |
| (parameters) -> { statements; } |

下面是用匿名内部类的代码：

|  |
| --- |
| button.addActionListener(**new** ActionListener() {  @Override  **public** **void** actionPerformed(ActionEvent e) {  System.***out***.print("Helllo Lambda in actionPerformed");  }  });  // 下面是使用 Lambda 表达式后：  button.addActionListener(  // actionPerformed 有一个参数 e 传入，所以用 (ActionEvent e)  (ActionEvent e) -> System.***out***.print("Helllo Lambda in actionPerformed")); |

上面是方法体包含了参数传入 (ActionEvent e)，如果没有参数则只需 ( )，例如 Thread 中的 run 方法就没有参数传入，当它使用 Lambda 表达式后：

|  |
| --- |
| Thread t = **new** Thread(  // run 没有参数传入，所以用 (), 后面用 {} 包起方法体  () -> {  System.***out***.println("Hello from a thread in run");  });  // 通过上面两个代码的比较可以发现使用 Lambda 表达式可以简化代码，并提高代码的可读性。  // 为了进一步简化 Lambda 表达式，可以使用方法引用。例如，下面三种分别是使用内部类，使用 Lambda  // 表示式和使用方法引用方式的比较：  // 1. 使用内部类  Function<Integer, String> f = **new** Function<Integer, String>() {  @Override  **public** String apply(Integer t) {  **return** **null**;  }  };  // 2. 使用 Lambda 表达式  Function<Integer, String> f2 = (k) -> String.*valueOf*(k);  // 3. 使用方法引用的方式  Function<Integer, String> f1 = String::*valueOf*; |

要使用 Lambda 表达式，需要定义一个函数式接口，这样往往会让程序充斥着过量的仅为 Lambda 表达式服务的函数式接口。为了减少这样过量的函数式接口，Java 8 在 java.util.function 中增加了不少新的函数式通用接口。例如：

Function<T, R>：将 T 作为输入，返回 R 作为输出，他还包含了和其他函数组合的默认方法。

Predicate<T> ：将 T 作为输入，返回一个布尔值作为输出，该接口包含多种默认方法来将 Predicate 组合成其他复杂的逻辑（与、或、非）。

Consumer<T> ：将 T 作为输入，不返回任何内容，表示在单个参数上的操作。

|  |
| --- |
| //例如，People 类中有一个方法 getMaleList 需要获取男性的列表，这里需要定义一个函数式接口 PersonInterface：  **interface** PersonInterface {  **public** **boolean** test(Person person);  }  **public** **class** People {  **private** List<Person> persons = **new** ArrayList<Person>();  **public** List<Person> getMaleList(PersonInterface filter) {  List<Person> res = **new** ArrayList<Person>();  persons.forEach((Person person) -> {  **if** (filter.test(person)) {// 调用 PersonInterface 的方法  res.add(person);  }  });  **return** res;  }  }  // 为了去除 PersonInterface 这个函数式接口，  // 可以用通用函数式接口 Predicate 替代如下：  **class** People {  **private** List<Person> persons = **new** ArrayList<Person>();  **public** List<Person> getMaleList(Predicate<Person> predicate) {  List<Person> res = **new** ArrayList<Person>();  persons.forEach(person -> {  **if** (predicate.test(person)) {// 调用 Predicate 的抽象方法 test  res.add(person);  }  });  **return** res;  }  } |

## JVM

### 内存模型JMM

（1）内存模型，各个部分的名称，作用，注意事项，参数等

（2）对象在内存中的初始化过程

### 垃圾回收

（1）实现算法

（2）强软弱虚引用

### 虚拟机调优

### JVM启动流程

### Object.finalize

（1） Object类的finalize方法的实现原理

### JVM的一些命令

（详见：http://blog.csdn.net/fenglibing/article/details/6411953）  
（1）jmap命令

（2）jps命令

（3）jstat命令

### 类加载机制

（详见：http://wiki.jikexueyuan.com/project/java-vm/class-loading-mechanism.html）

## 设计模式

### 普通设计模式（23种）

创建型模式

单例模式

构建模式

原型模式

抽象工厂模式

工厂方法模式

行为型模式

结构型模式

### 六大原则

### 多线程设计模式

### 其他

(1) 对象适配器模式满足了哪些原则？

## 反射

### 反射（性能，冲破private等）

### 序列化

### 克隆

### 内省机制

## 并发编程

### 线程（概念、状态、创建方式、临界区）

（1）概念，线程和进程的区别（参见操作系统部分）

**线程的概念：线程是程序中的一个执行流，一个执行流是由CPU运行程序代码并操纵程序的数据形成的。JAVA中的现成模型就是一个CPU、 程序代码和数据的封装体。**（参考自 张杨著Java并行程序设计，清华大学出版社，13页）

（2）线程的状态

**创建、就绪、运行、阻塞、终止 五种状态**

**创建：处于创建状态的线程有自己的内存空间，但是处于该状态的线程还没有运行，没有被激活。**

**就绪：处于创建状态的的线程调用start方法启动后，就会进入就绪状态，此时，该线程已经拥有了运行所需的所有条件，将进入线程队列排队等待CPU调度。**

**运行：当就绪的线程获得CPU的时间片后，该线程就转换为运行状态。运行状态的线程执行run方法中定义的操作。直到该方法调用结束进入终止状态，或者运行时由于缺乏运行必备的资源而进入阻塞状态。处于运行状态的线程也不是一直占用CPU的，运行（非阻塞）的线程可能会与就绪状态来回切换。**

**阻塞：在某些情况（1.共享资源被占用、2.等待I/O操作、3.调用了wait、sleep或suspend方法、4.尝试获得锁而该锁被其他线程持有）下，正在运行的线程会让出正在使用的CPU资源，进入阻塞状态；当引起阻塞的原因消除后，线程将进入就绪状态等待继续执行。**

**终止：达到终止状态的原因如下：1.线程run方法执行结束；2.线程通过某些方法（如destroy()）被提前终止；3.在run方法的执行期间发生了未捕获的异常；4.程序终止操作（如调用System.exit()）。**

（参考自 张杨著Java并行程序设计，清华大学出版社，13页~14页）

（3）创建方式

三种创建方式：

1.继承Thread类，重写run方法，启动的时候调用start方法；

2.实现Runnable接口，重写run方法，启动的时候new Thread(Runnable实例).start()；

3.实现Callable接口，定义返回值类型，重写call方法，启动方法为，将实例放入FutureTask实例中去，再将FutureTask实例放入一个Thread对象中去，使用Thread的start方法启动。好处：具有返回值（FutureTask可以通过调用get方法拿到异步任务的返回值）

（4）临界区

临界区就是在同一时刻只能有一个任务访问的代码区。在[**Java**](http://lib.csdn.net/base/java)中通常使用下面的方式来实现：

synchronized(syncObject) { //critical section }

可以使用Lock对象来实现临界区

**每个java对象都隐含有一把锁**。

使用临界区的执行效率要比使用synchronized方法的执行效率要高，因为其锁粒度更小。

（5）监视器

监视器（Monitor）有如下特征：

**一个监视器是只有一个私有属性的类；**

**每个监视器类的对象实力都有一个相关联的锁。这个锁将对对象的所有方法加锁。如：obj.method()方法调用开始时，自动获取obj对象的锁，方法调用结束时解锁。**

**Java中每个对象都有一把隐式的锁。**

（6）阻塞和非阻塞

前提： **如果线程请求某一种资源而得不到响应**，线程可以采用多种方式来决定接下来要采取的动作。根据动作的不同，分为阻塞和非阻塞

**非阻塞**：**若线程采用不断尝试的方式：在每次请求资源得不到满足的情况下，下次仍然继续请求，知道请求获得满足。这种方式就是非阻塞的。**

**阻塞**：**线程并非一直等待，而是被阻塞，这样可以让出CPU的资源来执行一些其他的操作。等待一段时间后，再去尝试获取资源。**

目前采用阻塞和非阻塞结合的方式：**首先以非阻塞的方式去尝试一定的次数，如果限定次数范围内，线程可以获得资源则线程继续运行，否则线程进入阻塞状态，等待一段时间后在尝试。**

### 锁

（1）锁机制的同步锁、 可重入锁和 读写锁以及这三种锁机制的比较

同步锁：使用同步方法或同步块，synchronized关键字修饰。同步方法的效率比同步块的低。

可重入锁：是一种无阻塞的同步机制，在java.util.concurrent.locks包下，ReentranLock类。比同步锁功能强大，有很多功能。支持公平和非公平策略。

读写锁：ReentrantReadWriteLock类，具有：锁的获取顺序（公平和非公平）、锁降级（允许将写锁降级为读锁，降级后允许更大程度的并发）、支持中断（读和写锁都支持中断，即如果锁整被其他线程持有，则可以被中断请求）

三种锁机制的比较：

同步锁是一种互斥锁，优点：形式简单使用方便，缺点：依赖于隐藏在对象后的内置监视器，不直观。

可重入锁是一种互斥锁，优点：在同步锁的基础上扩展了许多功能，如：非阻塞枷锁操作、在尝试获取锁时可中断、测试锁是否正在被持有、锁的获取顺序等。

读写锁出了提供可重入锁的一些特性外，还包含：把锁分为读锁和写锁，允许更大程度的并发

（2）锁的不足

**锁竞争问题**：多个线程试图访问被同一互斥锁保护的共享资源，并且此时有一个或多个线程等待获得该锁。此时就产生了锁竞争。

锁竞争问题是多核时代容易发生的问题。会导致只有一个CPU在运行，其他的CPU都处于空闲状态。解决办法：目前无法完全解决，有种方式是STM（软件事务型内存），其缺点是开销较大，且容易导致活锁。STM无法完全取代锁。

### 线程间通信

### 线程同步障栅

### 线程执行器

### Fork/Join框架

## 网络编程

## NIO

## 大数据

### Hadoop原理、HDFS实现原理、MapReduce原理、Yarn原理

### Hive、HBase、Storm、Flume、Zookeeper

### 面试题

在2.5亿个整数中找出不重复的整数，注，内存不足以容纳这2.5亿个整数。

答：采用2-Bitmap（每个数分配2bit，00表示不存在，01表示出现一次，10表示多次，11无意义）进行，共需内存2^32 \* 2 bit=1 GB内存，还可以接受。然后扫描这2.5亿个整数，查看Bitmap中相对应位，如果是00变01，01变10，10保持不变。所描完事后，查看bitmap，把对应位是01的整数输出即可。

## JavaWeb

### Session生命周期

（http://book.51cto.com/art/200912/169766.htm）

### HTTP协议

#### HTTP简介

HTTP协议是Hyper Text Transfer Protocol（超文本传输协议）的缩写,是用于从万维网（WWW:World Wide Web ）服务器传输超文本到本地浏览器的传送协议。

HTTP是一个基于TCP/IP通信协议来传递数据（HTML 文件, 图片文件, 查询结果等）。

HTTP是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。它于1990年提出，经过几年的使用与发展，得到不断地完善和扩展。目前在WWW中使用的是HTTP/1.0的第六版，HTTP/1.1的规范化工作正在进行之中，而且HTTP-NG(Next Generation of HTTP)的建议已经提出。

HTTP协议工作于客户端-服务端架构为上。浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送所有请求。Web服务器根据接收到的请求后，向客户端发送响应信息。



http请求-响应模型.jpg

#### 主要特点

1、简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。

2、灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记。

3.无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。

4.无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

5、支持B/S及C/S模式。

#### HTTP之URL

HTTP使用统一资源标识符（Uniform Resource Identifiers, URI）来传输数据和建立连接。URL是一种特殊类型的URI，包含了用于查找某个资源的足够的信息

URL,全称是UniformResourceLocator, 中文叫统一资源定位符,是互联网上用来标识某一处资源的地址。以下面这个URL为例，介绍下普通URL的各部分组成：

http://www.aspxfans.com:8080/news/index.asp?boardID=5&ID=24618&page=1#name

从上面的URL可以看出，一个完整的URL包括以下几部分：

1.协议部分：该URL的协议部分为“http：”，这代表网页使用的是HTTP协议。在Internet中可以使用多种协议，如HTTP，FTP等等本例中使用的是HTTP协议。在"HTTP"后面的“//”为分隔符

2.域名部分：该URL的域名部分为“www.aspxfans.com”。一个URL中，也可以使用IP地址作为域名使用

3.端口部分：跟在域名后面的是端口，域名和端口之间使用“:”作为分隔符。端口不是一个URL必须的部分，如果省略端口部分，将采用默认端口

4.虚拟目录部分：从域名后的第一个“/”开始到最后一个“/”为止，是虚拟目录部分。虚拟目录也不是一个URL必须的部分。本例中的虚拟目录是“/news/”

5.文件名部分：从域名后的最后一个“/”开始到“？”为止，是文件名部分，如果没有“?”,则是从域名后的最后一个“/”开始到“#”为止，是文件部分，如果没有“？”和“#”，那么从域名后的最后一个“/”开始到结束，都是文件名部分。本例中的文件名是“index.asp”。文件名部分也不是一个URL必须的部分，如果省略该部分，则使用默认的文件名

6.锚部分：从“#”开始到最后，都是锚部分。本例中的锚部分是“name”。锚部分也不是一个URL必须的部分

7.参数部分：从“？”开始到“#”为止之间的部分为参数部分，又称搜索部分、查询部分。本例中的参数部分为“boardID=5&ID=24618&page=1”。参数可以允许有多个参数，参数与参数之间用“&”作为分隔符。

（原文：http://blog.csdn.net/ergouge/article/details/8185219 ）

#### URI和URL的区别

**URI，是uniform resource identifier，统一资源标识符，用来唯一的标识一个资源。**

Web上可用的每种资源如HTML文档、图像、视频片段、程序等都是一个来URI来定位的

URI一般由三部组成：

①访问资源的命名机制

②存放资源的主机名

③资源自身的名称，由路径表示，着重强调于资源。

**URL是uniform resource locator，统一资源定位器，它是一种具体的URI，即URL可以用来标识一个资源，而且还指明了如何locate这个资源。**

URL是Internet上用来描述信息资源的字符串，主要用在各种WWW客户程序和服务器程序上，特别是著名的Mosaic。

采用URL可以用一种统一的格式来描述各种信息资源，包括文件、服务器的地址和目录等。URL一般由三部组成：

①协议(或称为服务方式)

②存有该资源的主机IP地址(有时也包括端口号)

③主机资源的具体地址。如目录和文件名等

**URN，uniform resource name，统一资源命名，是通过名字来标识资源，比如mailto:java-net@java.sun.com。**

URI是以一种抽象的，高层次概念定义统一资源标识，而URL和URN则是具体的资源标识的方式。URL和URN都是一种URI。笼统地说，每个 URL 都是 URI，但不一定每个 URI 都是 URL。这是因为 URI 还包括一个子类，即统一资源名称 (URN)，它命名资源但不指定如何定位资源。上面的 mailto、news 和 isbn URI 都是 URN 的示例。

在Java的URI中，一个URI实例可以代表绝对的，也可以是相对的，只要它符合URI的语法规则。而URL类则不仅符合语义，还包含了定位该资源的信息，因此它不能是相对的。

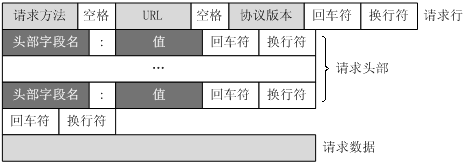
在Java类库中，URI类不包含任何访问资源的方法，它唯一的作用就是解析。

相反的是，URL类可以打开一个到达资源的流。

#### HTTP之请求消息Request

客户端发送一个HTTP请求到服务器的请求消息包括以下格式：

**请求行（request line）、请求头部（header）、空行**和**请求数据**四个部分组成。



Http请求消息结构.png

请求行以一个方法符号开头，以空格分开，后面跟着请求的URI和协议的版本。you

Get请求例子，使用Charles抓取的request：

GET /562f25980001b1b106000338.jpg HTTP/1.1

Host img.mukewang.com

User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36

Accept image/webp,image/\*,\*/\*;q=0.8

Referer http://www.imooc.com/

Accept-Encoding gzip, deflate, sdch

Accept-Language zh-CN,zh;q=0.8

**第一部分：请求行，用来说明请求类型,要访问的资源以及所使用的HTTP版本.**

GET说明请求类型为GET,[/562f25980001b1b106000338.jpg]为要访问的资源，该行的最后一部分说明使用的是HTTP1.1版本。

**第二部分：请求头部，紧接着请求行（即第一行）之后的部分，用来说明服务器要使用的附加信息**

从第二行起为请求头部，HOST将指出请求的目的地.User-Agent,服务器端和客户端脚本都能访问它,它是浏览器类型检测逻辑的重要基础.该信息由你的浏览器来定义,并且在每个请求中自动发送等等

**第三部分：空行，请求头部后面的空行是必须的**

即使第四部分的请求数据为空，也必须有空行。

**第四部分：请求数据也叫主体，可以添加任意的其他数据。**

这个例子的请求数据为空。

POST请求例子，使用Charles抓取的request：

POST / HTTP1.1

Host:www.wrox.com

User-Agent:Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)

Content-Type:application/x-www-form-urlencoded

Content-Length:40

Connection: Keep-Alive

name=Professional%20Ajax&publisher=Wiley

**第一部分：请求行，第一行明了是post请求，以及http1.1版本。**

**第二部分：请求头部，第二行至第六行。**

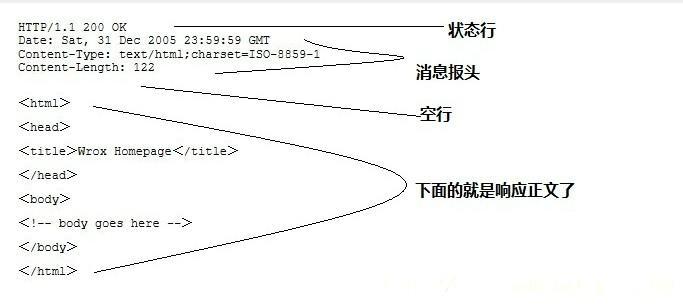
**第三部分：空行，第七行的空行。**

**第四部分：请求数据，第八行。**

#### HTTP之响应消息Response

一般情况下，服务器接收并处理客户端发过来的请求后会返回一个HTTP的响应消息。

**HTTP响应也由四个部分组成，分别是：状态行、消息报头、空行和响应正文。**



http响应消息格式.jpg

例子

HTTP/1.1 200 OK

Date: Fri, 22 May 2009 06:07:21 GMT

Content-Type: text/html; charset=UTF-8

<html>

<head></head>

<body>

<!--body goes here-->

</body>

</html>

**第一部分：状态行，由HTTP协议版本号， 状态码， 状态消息 三部分组成。**

第一行为状态行，（HTTP/1.1）表明HTTP版本为1.1版本，状态码为200，状态消息为（ok）

**第二部分：消息报头，用来说明客户端要使用的一些附加信息**

第二行和第三行为消息报头，

Date:生成响应的日期和时间；Content-Type:指定了MIME类型的HTML(text/html),编码类型是UTF-8

**第三部分：空行，消息报头后面的空行是必须的**

**第四部分：响应正文，服务器返回给客户端的文本信息。**

空行后面的html部分为响应正文。

#### HTTP之状态码

状态代码有三位数字组成，第一个数字定义了响应的类别，共分五种类别:

1xx：指示信息--表示请求已接收，继续处理

2xx：成功--表示请求已被成功接收、理解、接受

3xx：重定向--要完成请求必须进行更进一步的操作

4xx：客户端错误--请求有语法错误或请求无法实现

5xx：服务器端错误--服务器未能实现合法的请求

常见状态码：

**1**消息  
**2**成功  
**200** OK  
请求已成功，请求所希望的响应头或数据体将随此响应返回。  
**202** Accepted  
服务器已接受请求，但尚未处理。正如它可能被拒绝一样，最终该请求可能会也可能不会被执行。在异步操作的场合下，没有比发送这个状态码更方便的做法了。  
**3**重定向  
**4**请求错误  
**400** Bad Request  
1、语义有误，当前请求无法被服务器理解。除非进行修改，否则客户端不应该重复提交这个请求。  
2、请求参数有误。

**401** Unauthorized

请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用

**403** Forbidden  
服务器已经理解请求，但是拒绝执行它。与401响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交。如果这不是一个 HEAD 请求，而且服务器希望能够讲清楚为何请求不能被执行，那么就应该在实体内描述拒绝的原因。当然服务器也可以返回一个404响应，假如它不希望让客户端获得任何信息。  
**404** Not Found  
请求失败，请求所希望得到的资源未被在服务器上发现。没有信息能够告诉用户这个状况到底是暂时的还是永久的。假如服务器知道情况的话，应当使用410状态码来告知旧资源因为某些内部的配置机制问题，已经永久的不可用，而且没有任何可以跳转的地址。404这个状态码被广泛应用于当服务器不想揭示到底为何请求被拒绝或者没有其他适合的响应可用的情况下。出现这个错误的最有可能的原因是服务器端没有这个页面。  
**408** Request Timeout  
请求超时。客户端没有在服务器预备等待的时间内完成一个请求的发送。客户端可以随时再次提交这一请求而无需进行任何更改。  
**5**、**6** 服务器错误  
**500** Internal Server Error  
服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器端的源代码出现错误时出现。  
**502** Bad Gateway  
作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。  
**503** Service Unavailable  
由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。如果能够预计延迟时间，那么响应中可以包含一个 Retry-After 头用以标明这个延迟时间。如果没有给出这个 Retry-After 信息，那么客户端应当以处理500响应的方式处理它。  
注意：503状态码的存在并不意味着服务器在过载的时候必须使用它。某些服务器只不过是希望拒绝客户端的连接。  
**600** Unparseable Response Headers  
源站没有返回响应头部，只返回实体内容

更多状态码http://www.runoob.com/http/http-status-codes.html

#### HTTP请求方法

根据HTTP标准，HTTP请求可以使用多种请求方法。

HTTP1.0定义了三种请求方法： GET, POST 和 HEAD方法。

HTTP1.1新增了五种请求方法：OPTIONS, PUT, DELETE, TRACE 和 CONNECT 方法。

GET 请求指定的页面信息，并返回实体主体。

HEAD 类似于get请求，只不过返回的响应中没有具体的内容，用于获取报头

POST 向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。

PUT 从客户端向服务器传送的数据取代指定的文档的内容。

DELETE 请求服务器删除指定的页面。

CONNECT HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。

OPTIONS 允许客户端查看服务器的性能。

TRACE 回显服务器收到的请求，主要用于测试或诊断。

#### HTTP工作原理

HTTP协议定义Web客户端如何从Web服务器请求Web页面，以及服务器如何把Web页面传送给客户端。HTTP协议采用了请求/响应模型。客户端向服务器发送一个请求报文，请求报文包含请求的方法、URL、协议版本、请求头部和请求数据。服务器以一个状态行作为响应，响应的内容包括协议的版本、成功或者错误代码、服务器信息、响应头部和响应数据。

以下是 HTTP 请求/响应的步骤：

1、客户端连接到Web服务器

一个HTTP客户端，通常是浏览器，与Web服务器的HTTP端口（默认为80）建立一个TCP套接字连接。例如，http://www.oakcms.cn。

2、发送HTTP请求

通过TCP套接字，客户端向Web服务器发送一个文本的请求报文，一个请求报文由请求行、请求头部、空行和请求数据4部分组成。

3、服务器接受请求并返回HTTP响应

Web服务器解析请求，定位请求资源。服务器将资源复本写到TCP套接字，由客户端读取。一个响应由状态行、响应头部、空行和响应数据4部分组成。

4、释放连接TCP连接

若connection 模式为close，则服务器主动关闭TCP连接，客户端被动关闭连接，释放TCP连接;若connection 模式为keepalive，则该连接会保持一段时间，在该时间内可以继续接收请求;

5、客户端浏览器解析HTML内容

客户端浏览器首先解析状态行，查看表明请求是否成功的状态代码。然后解析每一个响应头，响应头告知以下为若干字节的HTML文档和文档的字符集。客户端浏览器读取响应数据HTML，根据HTML的语法对其进行格式化，并在浏览器窗口中显示。

例如：在浏览器地址栏键入URL，按下回车之后会经历以下流程：

1、浏览器向 DNS 服务器请求解析该 URL 中的域名所对应的 IP 地址;

2、解析出 IP 地址后，根据该 IP 地址和默认端口 80，和服务器建立TCP连接;

3、浏览器发出读取文件(URL 中域名后面部分对应的文件)的HTTP 请求，该请求报文作为 TCP 三次握手的第三个报文的数据发送给服务器;

4、服务器对浏览器请求作出响应，并把对应的 html 文本发送给浏览器;

5、释放 TCP连接;

6、浏览器将该 html 文本并显示内容;

#### GET和POST请求的区别

**GET请求**

GET /books/?sex=man&name=Professional HTTP/1.1

Host: www.wrox.com

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)

Gecko/20050225 Firefox/1.0.1

Connection: Keep-Alive

注意最后一行是空行

**POST请求**

POST / HTTP/1.1

Host: www.wrox.com

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)

Gecko/20050225 Firefox/1.0.1

Content-Type: application/x-www-form-urlencoded

Content-Length: 40

Connection: Keep-Alive

name=Professional%20Ajax&publisher=Wiley

1、GET提交，请求的数据会附在URL之后（就是把数据放置在HTTP协议头中），以?分割URL和传输数据，多个参数用&连接；例 如：login.action?name=hyddd&password=idontknow&verify=%E4%BD%A0 %E5%A5%BD。如果数据是英文字母/数字，原样发送，如果是空格，转换为+，如果是中文/其他字符，则直接把字符串用BASE64加密，得出如： %E4%BD%A0%E5%A5%BD，其中％XX中的XX为该符号以16进制表示的ASCII。

POST提交：把提交的数据放置在是HTTP包的包体中。上文示例中红色字体标明的就是实际的传输数据

因此，GET提交的数据会在地址栏中显示出来，而POST提交，地址栏不会改变

2、传输数据的大小：首先声明：HTTP协议没有对传输的数据大小进行限制，HTTP协议规范也没有对URL长度进行限制。

而在实际开发中存在的限制主要有：

GET:特定浏览器和服务器对URL长度有限制，例如 IE对URL长度的限制是2083字节(2K+35)。对于其他浏览器，如Netscape、FireFox等，理论上没有长度限制，其限制取决于操作系 统的支持。

因此对于GET提交时，传输数据就会受到URL长度的 限制。

POST:由于不是通过URL传值，理论上数据不受 限。但实际各个WEB服务器会规定对post提交数据大小进行限制，Apache、IIS6都有各自的配置。

3、安全性

POST的安全性要比GET的安全性高。比如：通过GET提交数据，用户名和密码将明文出现在URL上，因为(1)登录页面有可能被浏览器缓存；(2)其他人查看浏览器的历史纪录，那么别人就可以拿到你的账号和密码了，除此之外，使用GET提交数据还可能会造成Cross-site request forgery攻击

4、Http get,post,soap协议都是在http上运行的

（1）get：请求参数是作为一个key/value对的序列（查询字符串）附加到URL上的

查询字符串的长度受到web浏览器和web服务器的限制（如IE最多支持2048个字符），不适合传输大型数据集同时，它很不安全

（2）post：请求参数是在http标题的一个不同部分（名为entity body）传输的，这一部分用来传输表单信息，因此必须将Content-type设置为:application/x-www-form- urlencoded。post设计用来支持web窗体上的用户字段，其参数也是作为key/value对传输。

但是：它不支持复杂数据类型，因为post没有定义传输数据结构的语义和规则。

（3）soap：是http post的一个专用版本，遵循一种特殊的xml消息格式

Content-type设置为: text/xml 任何数据都可以xml化。

Http协议定义了很多与服务器交互的方法，最基本的有4种，分别是GET,POST,PUT,DELETE. 一个URL地址用于描述一个网络上的资源，而HTTP中的GET, POST, PUT, DELETE就对应着对这个资源的查，改，增，删4个操作。 我们最常见的就是GET和POST了。GET一般用于获取/查询资源信息，而POST一般用于更新资源信息.

我们看看GET和POST的区别

GET提交的数据会放在URL之后，以?分割URL和传输数据，参数之间以&相连，如EditPosts.aspx?name=test1&id=123456. POST方法是把提交的数据放在HTTP包的Body中.

GET提交的数据大小有限制（因为浏览器对URL的长度有限制），而POST方法提交的数据没有限制.

GET方式需要使用Request.QueryString来取得变量的值，而POST方式通过Request.Form来获取变量的值。

GET方式提交数据，会带来安全问题，比如一个登录页面，通过GET方式提交数据时，用户名和密码将出现在URL上，如果页面可以被缓存或者其他人可以访问这台机器，就可以从历史记录获得该用户的账号和密码.

## Struts2

## Hibernate

## Spring

## MyBatis

## SpringMVC

## 最佳实践

# 算法和数据结构

（涉及到的代码部分要求会默写）

## 链表、栈（及其应用，如表达式求值，走迷宫等）、队列（及其应用，如杨辉三角等）

## 二叉树（递归及非递归遍历、层次遍历），平衡二叉树、红黑树

### Treap（树堆）

### Trie树

## 图 （广度、深度、递归以及非递归）最小生成树（普里姆、克鲁斯卡尔），最短路径（地杰斯特拉，弗洛伊德）、拓扑排序

## 排序（插入、交换、选择、归并，外部）

## 查找（二分等）

## 常见算法的时空复杂度，小知识点（二叉树和森林的对应关系、离散数学关于图论的计算公式等）

## 串的模式匹配（KMP）

## 动态规划思想

## 贪心思想

## 分治思想

## 其他

### n个元素进栈，共有几种出栈顺序？

答：C(2n, n) / (n+1)种 （详见：http://blog.csdn.net/zyearn/article/details/778716）

### （三七互娱面试真题）已知一个单链表中的某一个节点，不知道该链表的头指针，请说出如何删除该节点。

答：将该节点的下一个节点的value值赋给当前节点的value，然后删除该节点的下一个节点即可。

# 数据库部分

## SQL基础

1. 例题
2. 【关于执行顺序】

<https://www.nowcoder.com/questionTerminal/7abd397a22054bbb99e2b3982ff81d01>  
对于满足SQL92标准的SQL语句：SELECT foo,count(foo) FROM pokes WHERE foo>10 GROUP BY foo HAVING ORDER BY foo，其执行的顺序应该为（ ）

1. FROM->WHERE->GROUP BY->HAVING->SELECT->ORDER BY
2. FROM->GROUP BY->WHERE->HAVING->SELECT->ORDER BY
3. FROM->WHERE->GROUP BY->HAVING->ORDER BY->SELECT
4. FROM->WHERE->ORDER BY->GROUP BY->HAVING->SELECT

解析：

标准的 SQL 的解析顺序为:

(1).FROM 子句, 组装来自不同数据源的数据

(2).WHERE 子句, 基于指定的条件对记录进行筛选

(3).GROUP BY 子句, 将数据划分为多个分组

(4).使用聚合函数进行计算

(5).使用 HAVING 子句筛选分组

(6).计算Select所有的表达式

(7).使用 ORDER BY 对结果集进行排序

答案：A

## 事务

## 锁

## NoSQL

## MySQL和Redis

## 三大范式

**第一范式(1NF)**

定义：如果关系模式R的每个关系r的属性都是不可分的数据项，那么就称R是第一范式的模式。

简单的说，**每一个属性都是原子项，不可分割。**

1NF是关系模式应具备的最起码的条件，如果数据库设计不能满足第一范式，就不称为关系型数据库。关系数据库设计研究的关系规范化是在1NF之上进行的。

例如(学生信息表)：

学生编号 姓名 性别 联系方式

20080901 张三 男 email:zs@126.com,phone:88886666

20080902 李四 女 email:ls@126.com,phone:66668888

以上的表就不符合，第一范式：联系方式字段可以再分，所以变更为正确的是：

学生编号 姓名 性别 电子邮件 电话

20080901 张三 男 zs@126.com 88886666

20080902 李四 女 ls@126.com 66668888

**第二范式（2NF）**

定义：如果关系模式R是1NF，且每个非主属性完全函数依赖于候选键，那么就称R是第二范式。

简单的说，**第二范式要满足以下的条件：首先要满足第一范式，其次每个非主属性要完全函数依赖与候选键，或者是主键。也就是说，每个非主属性是由整个主键函数决定的，而不能由主键的一部分来决定。**

例如(学生选课表):

学生 课程 教师 教师职称 教材 教室 上课时间

李四 Spring 张老师 java讲师 《Spring深入浅出》 ３０１ 08:00

张三 Struts 杨老师 java讲师 《Struts in Action》 ３０２ 13:30

这里通过（学生，课程）可以确定教师、教师职称，教材，教室和上课时间，所以可以把（学生，课程）作为主键。但是，教材并不完全依赖于（学生，课程），只拿出课程就可以确定教材，因为一个课程，一定指定了某个教材。这就叫不完全依赖，或者部分依赖。出现这种情况，就不满足第二范式。

修改后，选课表：

学生 课程 教师 教师职称 教室 上课时间

李四 Spring 张老师 java讲师 ３０１ 08:00

张三 Struts 杨老师 java讲师 ３０２ 13:30

课程表：

课程 教材

Spring 《Spring深入浅出》

Struts 《Struts in Action》

所以，第二范式可以说是消除部分依赖。第二范式可以减少插入异常，删除异常和修改异常。

**第三范式（3NF）**

定义：如果关系模式R是2NF，且关系模式R（U，F）中的所有非主属性对任何候选关键字都不存在传递依赖，则称关系R是属于第三范式。

简单的说，**第三范式要满足以下的条件：首先要满足第二范式，其次非主属性之间不存在函数依赖。由于满足了第二范式，表示每个非主属性都函数依赖于主键。如果非主属性之间存在了函数依赖，就会存在传递依赖，这样就不满足第三范式。**

上例中修改后的选课表中，一个教师能确定一个教师职称。这样，教师依赖于（学生，课程），而教师职称又依赖于教师，这叫传递依赖。第三范式就是要消除传递依赖。

修改后，选课表：

学生 课程 教师 教室 上课时间

李四 Spring 张老师 ３０１ 08:00

张三 Struts 杨老师 ３０２ 13:30

教师表：

教师 教师职称

张老师 java讲师

杨老师 java讲师

这样，新教师的职称在没被选课的时候也有地方存了，没人选这个教师的课的时候教师的职称也不至于被删除，修改教师职称时只修改教师表就可以了。

简单的说，

**第一范式就是原子性，字段不可再分割；**

**第二范式就是完全依赖，没有部分依赖；**

**第三范式就是没有传递依赖。**

# 网络部分

## OSI和TCP/IP架构，各层职能，所属协议

## 流量控制、三次握手，四次挥手、TCP和UDP区别、HTTP、HTTPS的五次握手

### HTTP与TCP/IP区别

TPC/IP协议是传输层协议，主要解决数据如何在网络中传输，而HTTP是应用层协议，主要解决如何包装数据。WEB使用HTTP协议作应用层协议，以封装HTTP 文本信息，然后使用TCP/IP做传输层协议将它发到网络上。

下面的图表试图显示不同的TCP/IP和其他的协议在最初OSI（Open System Interconnect）模型中的位置：



### CA证书是什么

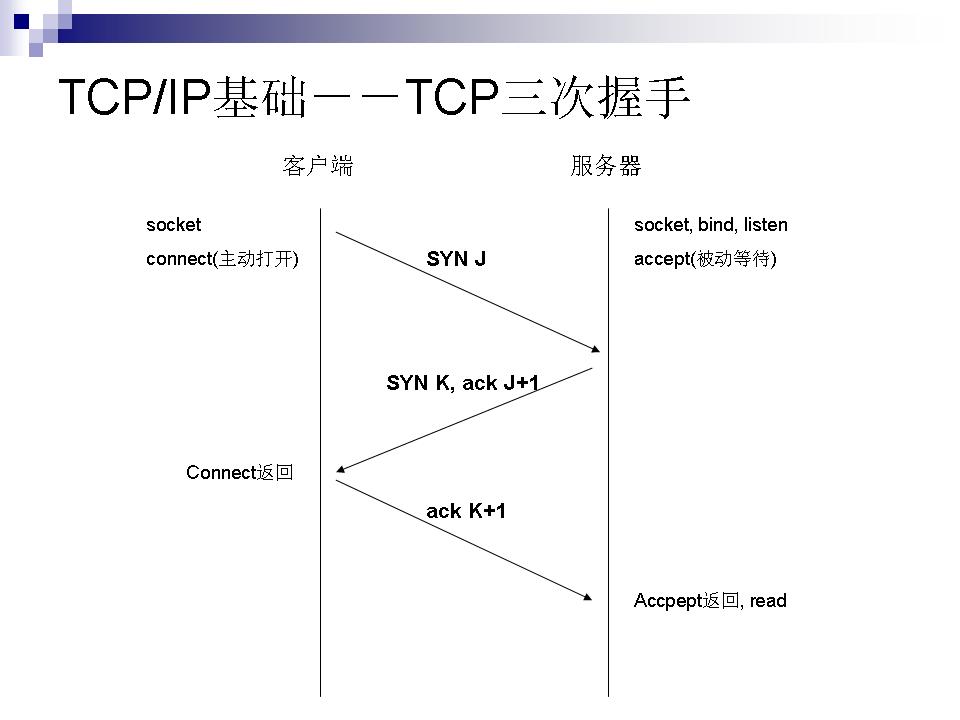
CA（Certificate Authority）是负责管理和签发证书的第三方权威机构，是所有行业和公众都信任的、认可的。

CA证书，就是CA颁发的证书，可用于验证网站是否可信（针对HTTPS）、验证某文件是否可信（是否被篡改）等，也可以用一个证书来证明另一个证书是真实可信，最顶级的证书称为根证书。除了根证书（自己证明自己是可靠），其它证书都要依靠上一级的证书，来证明自己。

### HTTP三次握手

HTTP（HyperText Transfer Protocol)超文本传输协议是互联网上应用最为广泛的一种网络协议。由于信息是明文传输，所以被认为是不安全的。而关于HTTP的三次握手，其实就是使用三次TCP握手确认建立一个HTTP连接。

如下图所示，SYN（synchronous）是TCP/IP建立连接时使用的握手信号、Sequence number（序列号）、Acknowledge number（确认号码），三个箭头指向就代表三次握手，完成三次握手，客户端与服务器开始传送数据。



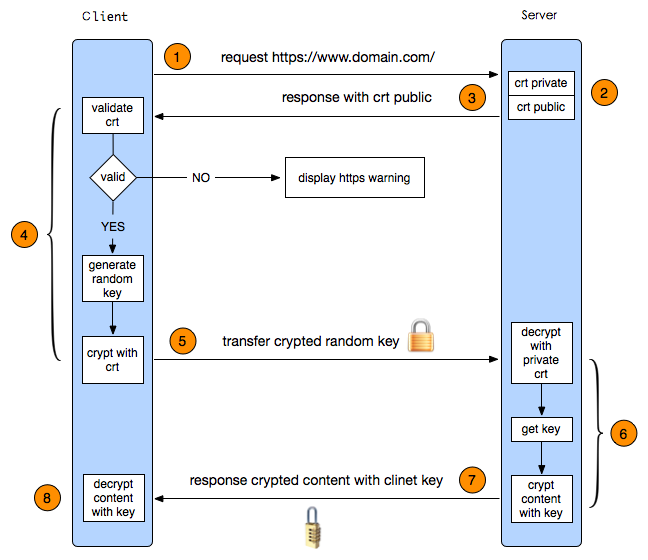
第一次握手：客户端发送syn包(syn=j)到服务器，并进入SYN\_SEND状态，等待服务器确认；

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=j+1），同时自己也发送一个SYN包（syn=k），即SYN+ACK包，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的SYN＋ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

### HTTPS握手过程

HTTPS在HTTP的基础上加入了SSL协议，SSL依靠证书来验证服务器的身份，并为浏览器和服务器之间的通信加密。具体是如何进行加密，解密，验证的，且看下图，下面的称为一次握手。



**1. 客户端发起HTTPS请求**

**2. 服务端的配置**

采用HTTPS协议的服务器必须要有一套数字证书，可以是自己制作或者CA证书。区别就是自己颁发的证书需要客户端验证通过，才可以继续访问，而使用CA证书则不会弹出提示页面。这套证书其实就是一对公钥和私钥。公钥给别人加密使用，私钥给自己解密使用。

**3. 传送证书**

这个证书其实就是公钥，只是包含了很多信息，如证书的颁发机构，过期时间等。

**4. 客户端解析证书**

这部分工作是有客户端的TLS来完成的，首先会验证公钥是否有效，比如颁发机构，过期时间等，如果发现异常，则会弹出一个警告框，提示证书存在问题。如果证书没有问题，那么就生成一个随即值，然后用证书对该随机值进行加密。

**5. 传送加密信息**

这部分传送的是用证书加密后的随机值，目的就是让服务端得到这个随机值，以后客户端和服务端的通信就可以通过这个随机值来进行加密解密了。

**6. 服务段解密信息**

服务端用私钥解密后，得到了客户端传过来的随机值(私钥)，然后把内容通过该值进行对称加密。所谓对称加密就是，将信息和私钥通过某种算法混合在一起，这样除非知道私钥，不然无法获取内容，而正好客户端和服务端都知道这个私钥，所以只要加密算法够彪悍，私钥够复杂，数据就够安全。

**7. 传输加密后的信息**

这部分信息是服务段用私钥加密后的信息，可以在客户端被还原。

**8. 客户端解密信息**

客户端用之前生成的私钥解密服务段传过来的信息，于是获取了解密后的内容。

PS: 整个握手过程第三方即使监听到了数据，也束手无策。

### 为什么HTTPS是安全的

在HTTPS握手的第四步中，如果站点的证书是不受信任的，会显示出现下面确认界面，确认了网站的真实性。另外第六和八步，使用客户端私钥加密解密，保证了数据传输的安全。



### HTTPS和HTTP的区别

1. https协议需要到ca申请证书或自制证书。

2. http的信息是明文传输，https则是具有安全性的ssl加密。

3. http是直接与TCP进行数据传输，而https是经过一层SSL（OSI表示层），用的端口也不一样，前者是80（需要国内备案），后者是443。

4. http的连接很简单，是无状态的；HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，比http协议安全。

**注意https加密是在传输层**

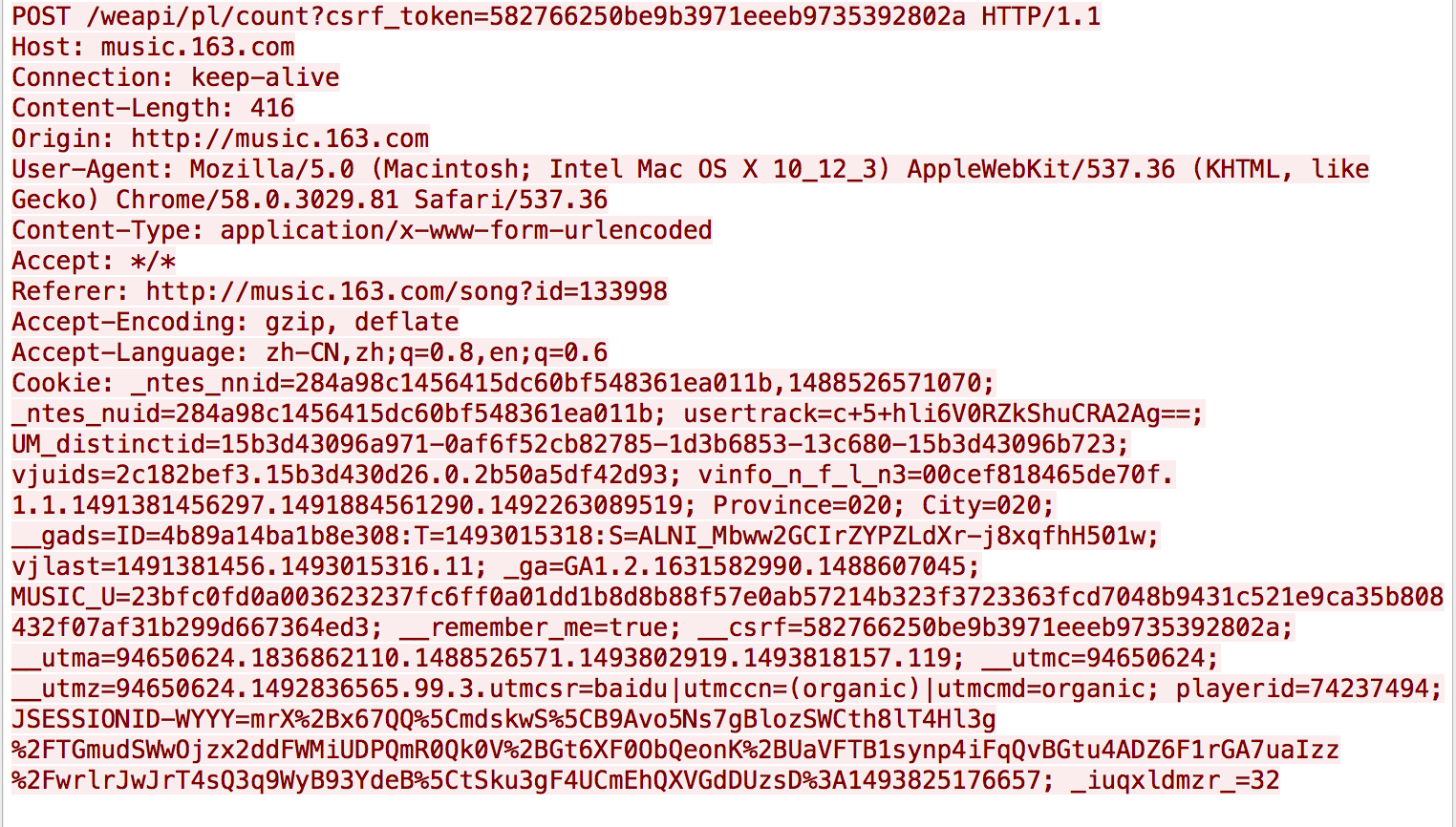
https报文在被包装成tcp报文的时候完成加密的过程，无论是https的**header域**也好，body域也罢都是会被加密的。

当使用**tcpdump或者wireshark**之类的tcp层工具抓包，获取是加密的内容，而如果用应用层抓包，使用**Charels(Mac)、Fildder(Windows)**抓包工具，那当然看到是明文的。

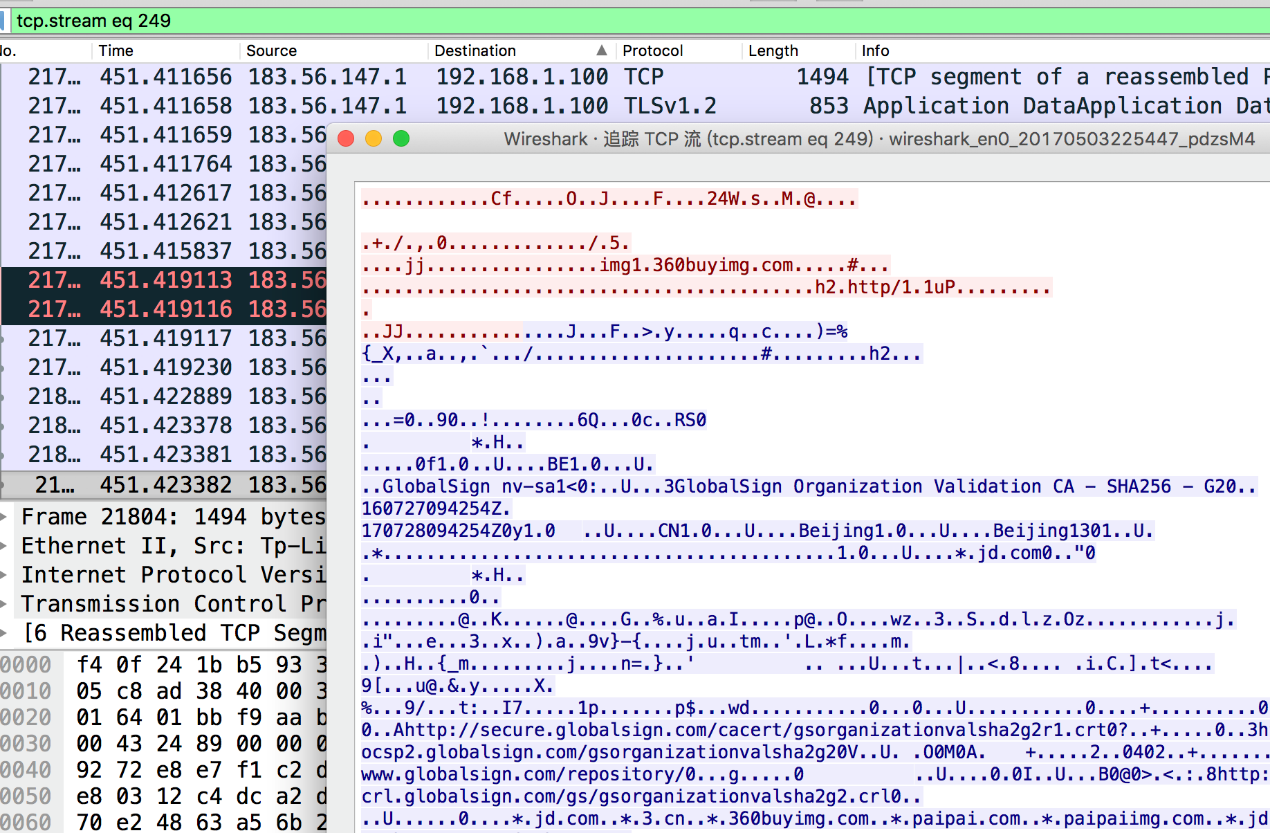
PS：HTTPS本身就是为了网络的传输安全。

例子，使用wireshark抓包：

http，可以看到抓到是明文的：



https，可以看到抓到是密文的：



### HTTPS一般使用的加密与HASH算法

非对称加密算法：RSA，DSA/DSS

对称加密算法：AES，RC4，3DES

HASH算法：MD5，SHA1，SHA256

## 其他面试题

(1) 如何将UDP协议修改成TCP协议？

答：修改报文的首部

# 操作系统部分

## Linux操作系统常用命令

1. 统计文件中出现次数最多的前10个单词

答：cat words.txt | sort | uniq -c | sort -kl, lnr | head -10

(详见：http://blog.sina.com.cn/s/blog\_5dce657a01012ddi.html)

2. linux下，系统资源占用相关命令？

答：top、free、uptime （详见：http://blog.51yip.com/linux/1162.html  以及  http://www.jb51.net/LINUXjishu/95640.html）

3. linux使用的进程间通信方式？

答：1.管道（pipe）、流管道（s\_pipe）、有名管道（FIFO）

       2.信号（signal）

       3.消息队列

       4.共享内存

       5.信号量

       6.套接字（socket）

       （详见：http://blog.csdn.net/gatieme/article/details/50908749）

4. 存储相关。磁盘调度算法有哪些？

答：先来先服务（FCFS）、最短寻道时间优先算法（SSTF）、扫描算法（SCAN）、循环扫描算法（CSCAN）。（详见：百度百科：磁盘调度算法）

5. linux阻塞与非阻塞，同步与异步、I/O模型

http://www.cnblogs.com/pang1567/p/4012834.html

# 软件工程部分

# 组成原理部分

# 机器学习部分

# HR面试问题汇总

## 人事面试一百问

来自网络：http://bbs.yingjiesheng.com/thread-1704164-1-1.html

### 请你自我介绍一下

回答提示：一般人回答这个问题过于平常，只说姓名、年龄、爱好、工作经验 ，这些在简历上都有，其实，企业最希望知道的是求职者能否胜任工作，包括：最强的技能、最深入研究的知识领域、个性中最积极的部分、做过的最成功的事，主要的成就等，这些都可以和学习无关，也可以和学习有关，但要突出积极的个性和做事的能力，说得合情合理企业才会相信。企业很重视一个人的礼貌，求职者要尊重考官，在回答每个问题之后都说一句“谢谢”。企业喜欢有礼貌的求职者。

### 你觉得你个性上最大的优点是什么？

回答提示：沉着冷静、条理清楚、立场坚定、顽强向上。  
乐于助人和关心他人、适应能力和幽默感、乐观和友爱。我在北大青鸟经过一到两年的培训及项目实战，加上实习工作，使我适合这份工作。我相信我能成功。

### 说说你最大的缺点？

回答提示：这个问题企业问的概率很大，通常不希望听到直接回答的缺点是什么等，如果求职者说自己小心眼、爱忌妒人、非常懒、脾气大、工作效率低，企业肯定不会录用你。绝对不要自作聪明地回答“我最大的缺点是过于追求完美”，有的人以为这样回答会显得自己比较出色，但事实上，他已经岌芨可危了。企业喜欢求职者从自己的优点说起，中间加一些小缺点，最后再把问题转回到优点上，突出优点的部分。企业喜欢聪明的求职者。

### 你对加班的看法？

回答提示：实际上好多公司问这个问题，并不证明一定要加班。 只是想测试你是否愿意为公司奉献。  
回答样本：如果是工作需要我会义不容辞加班。我现在单身，没有任何家庭负担，可以全身心的投入工作。但同时，我也会提高工作效率，减少不必要的加班

### 你对薪资的要求？

回答提示：如果你对薪酬的要求太低，那显然贬低自己的能力；  如果你对薪酬的要求太高，那又会显得你分量过重，公司受用不起。一些雇主通常都事先对求聘的职位定下开支预算，因而他们第一次提出的价钱往往是他们所能给予的最高价钱。他们问你只不过想证实一下这笔钱是否足以引起你对该工作的兴趣。  
回答样本一：“我对工资没有硬性要求。我相信贵公司在处理我的问题上会友善合理。我注重的是找对工作机会，所以只要条件公平，我则不会计较太多  
回答样本二：我受过系统的软件编程的训练，不需要进行大量的培训。而且我本人也对编程特别感兴趣。因此，我希望公司能根据我的情况和市场标准的水平，给我合理的薪水。  
回答样本三：如果你必须自己说出具体数目，请不要说一个宽泛的范围，那样你将只能得到最低限度的数字。最好给出一个具体的数字，这样表明你已经对当今的人才市场作了调查，知道像自己这样学历的雇员有什么样的价值。

### 在五年的时间内，你的职业规划？

回答提示：这是每一个应聘者都不希望被问到的问题，但是几乎每个人都会被问到。比较多的答案是“管理者”。但是近几年来，许多公司都已经建立了专门的技术途径。这些工作地位往往被称作“顾问”、“参议技师”或“高级软件工程师”等等。当然，说出其他一些你感兴趣的职位也是可以的，比如产品销售部经理，生产部经理等一些与你的专业有相关背景的工作。要知道，考官总是喜欢有进取心的应聘者，此时如果说“不知道”，或许就会使你丧失一个好机会。最普通的回答应该是“我准备在技术领域有所作为”或“我希望能按照公司的管理思路发展”。

### 你朋友对你的评价？

回答提示：想从侧面了解一下你的性格及与人相处的问题。  
?回答样本：“我的朋友都说我是一个可以信赖的人。因为，我一旦答应别人的事情，就一定会做到。如果我做不到，我就不会轻易许诺。  
?回答样本：”我觉的我是一个比较随和的人，与不同的人都可以友好相处。在我与人相处时，我总是能站在别人的角度考虑问题“

### 你还有什么问题要问吗？

回答提示：企业的这个问题看上去可有可无，其实很关键，企业不喜欢说“没有问题”的人，因为其很注重员工的个性和创新能力。企业不喜欢求职者问个人福利之类的问题，如果有人这样问：贵公司对新入公司的员工有没有什么培训项目，我可以参加吗？或者说贵公司的晋升机制是什么样的？企业将很欢迎，因为体现出你对学习的热情和对公司的忠诚度以及你的上进心。

### 如果通过这次面试我们单位录用了你，但工作一段时间却发现你根本不适合这个职位，你怎么办？

回答提示：一段时间发现工作不适合我，有两种情况：  
1、如果你确实热爱这个职业，那你就要不断学习，虚心向领导和同事学习业务知识和处事经验，了解这个职业的精神内涵和职业要求，力争减少差距；  
2、你觉得这个职业可有可无，那还是趁早换个职业，去发现适合你的，你热爱的职业，那样你的发展前途也会大点，对单位和个人都有好处。

### 在完成某项工作时，你认为领导要求的方式不是最好的，自己还有更好的方法，你应该怎么做？

回答提示：①.原则上我会尊重和服从领导的工作安排；同时私底下找机会以请教的口吻，婉转地表达自己的想法，看看领导是否能改变想法；  
②如果领导没有采纳我的建议，我也同样会按领导的要求认真地去完成这项工作；  
③.还有一种情况，假如领导要求的方式违背原则，我会坚决提出反对意见；如领导仍固执己见，我会毫不犹豫地再向上级领导反映。

### 如果你的工作出现失误，给本公司造成经济损失，你认为该怎么办？

回答提示：  
①我本意是为公司努力工作，如果造成经济损失，我认为首要的问题是想方设法去弥补或挽回经济损失。如果我无能力负责，希望单位帮助解决；  
②是责任问题。分清责任，各负其责，如果是我的责任，我甘愿受罚；如果是一个我负责的团队中别人的失误，也不能幸灾乐祸，作为一个团队，需要互相提携共同完成工作，安慰同事并且帮助同事查找原因总结经验。  
③总结经验教训，一个人的一生不可能不犯错误，重要的是能从自己的或者是别人的错误中吸取经验教训，并在今后的工作中避免发生同类的错误。检讨自己的工作方法、分析问题的深度和力度是否不够，以致出现了本可以避免的错误。

### 如果你在这次考试中没有被录用，你怎么打算？

回答提示：现在的社会是一个竞争的社会,从这次面试中也可看出这一点,有竞争就必然有优劣,有成功必定就会有失败.往往成功的背后有许多的困难和挫折,如果这次失败了也仅仅是一次而已,只有经过经验经历的积累才能塑造出一个完全的成功者。我会从以下几个方面来正确看待这次失败.  
第一、要敢于面对,面对这次失败不气馁,接受已经失去了这次机会就不会回头这个现实,从心理意志和精神上体现出对这次失败的抵抗力。要有自信,相信自己经历了这次之后经过努力一定能行.能够超越自我.  
第二、善于反思,对于这次面试经验要认真总结,思考剖析,能够从自身的角度找差距。正确对待自己,实事求是地评价自己,辩证的看待自己的长短得失,做一个明白人.  
第三、走出阴影,要克服这一次失败带给自己的心理压力,时刻牢记自己弱点,防患于未然,加强学习,提高自身素质.  
第四、认真工作,回到原单位岗位上后,要实实在在、踏踏实实地工作,三十六行,行行出状元,争取在本岗位上做出一定的成绩.  
第五、再接再厉,成为软件工程师或网络工程师一直是我的梦想,以后如果有机会我仍然后再次参加竞争.

### 如果你做的一项工作受到上级领导的表扬，但你主管领导却说是他做的，你该怎样？

回答提示：我首先不会找那位上级领导说明这件事，我会主动找我的主管领导来沟通，因为沟通是解决人际关系的最好办法，但结果会有两种：1.我的主管领导认识到自己的错误，我想我会视具体情况决定是否原谅他；2.他更加变本加厉的来威胁我，那我会毫不犹豫地找我的上级领导反映此事，因为他这样做会造成负面影响，对今后的工作不利。

### 谈谈你对跳槽的看法？

回答提示：（1）正常的"跳槽"能促进人才合理流动，应该支持；  
          （2）频繁的跳槽对单位和个人双方都不利，应该反对。

### 工作中你难以和同事、上司相处，你该怎么办？

回答提示：  
①我会服从领导的指挥，配合同事的工作。  
②我会从自身找原因，仔细分析是不是自己工作做得不好让领导不满意，同事看不惯。还要看看是不是为人处世方面做得不好。如果是这样的话 我会努力改正。  
③如果我找不到原因，我会找机会跟他们沟通，请他们指出我的不足。有问题就及时改正。  
④作为优秀的员工，应该时刻以大局为重，即使在一段时间内，领导和同事对我不理解，我也会做好本职工作，虚心向他们学习，我相信，他们会看见我在努力，总有一天会对我微笑的！

### 假设你在某单位工作，成绩比较突出，得到领导的肯定。但同时你发现同事们越来越孤立你，你怎么看这个问题？你准备怎么办？

回答提示：  
①成绩比较突出，得到领导的肯定是件好事情，以后更加努力  
②检讨一下自己是不是对工作的热心度超过同事间交往的热心了，加强同事间的交往及共同的兴趣爱好。  
③工作中，切勿伤害别人的自尊心  
④不再领导前拨弄是非  
⑤乐于助人对面

### 你最近是否参加了培训课程？谈谈培训课程的内容。是公司资助还是自费参加？

回答提示：是自费参加，就是北大青鸟的培训课程（可以多谈谈自己学的技术）。

### 你对于我们公司了解多少？

回答提示：在去公司面试前上网查一下该公司主营业务。如回答：贵公司有意改变策略，加强与国外大厂的OEM合作，自有品牌的部分则透过海外经销商。

### 请说出你选择这份工作的动机？

回答提示：这是想知道面试者对这份工作的热忱及理解度，并筛选因一时兴起而来应试的人，如果是无经验者，可以强调“就算职种不同，也希望有机会发挥之前的经验”。

### 你最擅长的技术方向是什么？

回答提示：说和你要应聘的职位相关的课程，表现一下自己的热诚没有什么坏处。

### 你能为我们公司带来什么呢？

回答提示：  
①假如你可以的话，试着告诉他们你可以减低他们的费用——“我已经接受过北大青鸟近两年专业的培训，立刻就可以上岗工作”。  
②企业很想知道未来的员工能为企业做什么，求职者应再次重复自己的优势，然后说：“就我的能力，我可以做一个优秀的员工在组织中发挥能力，给组织带来高效率和更多的收益”。企业喜欢求职者就申请的职位表明自己的能力，比如申请营销之类的职位，可以说：“我可以开发大量的新客户，同时，对老客户做更全面周到的服务，开发老客户的新需求和消费。”等等。

### 最能概括你自己的三个词是什么？

回答提示：  
我经常用的三个词是：适应能力强，有责任心和做事有始终，结合具体例子向主考官解释.

### 你的业余爱好是什么？

回答提示：找一些富于团体合作精神的，这里有一个真实的故事：有人被否决掉，因为他的爱好是深海潜水。主考官说：因为这是一项单人活动，我不敢肯定他能否适应团体工作。

### 作为被面试者给我打一下分.

回答提示：试着列出四个优点和一个非常非常非常小的缺点，（可以抱怨一下设施，没有明确责任人的缺点是不会有人介意的）。

### 你怎么理解你应聘的职位？

回答提示：把岗位职责和任务及工作态度阐述一下

### 喜欢这份工作的哪一点？

回答提示： 相信其实大家心中一定都有答案了吧！每个人的价值观不同，自然评断的标准也会不同，但是，在回答面试官这个问题时可不能太直接就把自己心理的话说出来，尤其是薪资方面的问题，不过一些无伤大雅的回答是不错的考虑，如交通方便，工作性质及内容颇能符合自己的兴趣等等都是不错的答案，不过如果这时自己能仔细思考出这份工作的与众不同之处，相信在面试上会大大加分。

### 为什么要离职?

回答提示：  
①回答这个问题时一定要小心，就算在前一个工作受到再大的委屈，对公司有多少的怨言，都千万不要表现出来，尤其要避免对公司本身主管的批评，避免面试官的负面情绪及印象；建议此时最好的回答方式是将问题归咎在自己身上，例如觉得工作没有学习发展的空间，自己想在面试工作的相关产业中多加学习，或是前一份工作与自己的生涯规划不合等等，回答的答案最好是积极正面的。  
②我希望能获得一份更好的工作，如果机会来临，我会抓住；我觉得目前的工作，已经达到顶峰，即沒有升迁机会。

### 说说你对行业、技术发展趋势的看法？

回答提示：企业对这个问题很感兴趣，只有有备而来的求职者能够过关。求职者可以直接在网上查找对你所申请的行业部门的信息，只有深入了解才能产生独特的见解。企业认为最聪明的求职者是对所面试的公司预先了解很多，包括公司各个部门，发展情况，在面试回答问题的时候可以提到所了解的情况，企业欢迎进入企业的人是“知己”，而不是“盲人”。

### 对工作的期望与目标何在？

回答提示：这是面试者用来评断求职者是否对自己有一定程度的期望、对这份工作是否了解的问题。对于工作有确实学习目标的人通常学习较快，对于新工作自然较容易进入状况，这时建议你，最好针对工作的性质找出一个确实的答案，如业务员的工作可以这样回答：“我的目标是能成为一个超级业务员，将公司的产品广泛的推销出去，达到最好的业绩成效；为了达到这个目标，我一定会努力学习，而我相信以我认真负责的态度，一定可以达到这个目标。”其他类的工作也可以比照这个方式来回答，只要在目标方面稍微修改一下就可以了。

### 说说你的家庭

回答提示：企业面试时询问家庭问题不是非要知道求职者家庭的情况，探究隐私，企业不喜欢探究个人隐私，而是要了解家庭背景对求职者的塑造和影响。企业希望听到的重点也在于家庭对求职者的积极影响。企业最喜欢听到的是：我很爱我的家庭！我的家庭一向很和睦，虽然我的父亲和母亲都是普通人，但是从小，我就看到我父亲起早贪黑，每天工作特别勤劳，他的行动无形中培养了我认真负责的态度和勤劳的精神。我母亲为人善良，对人热情，特别乐于助人，所以在单位人缘很好，她的一言一行也一直在教导我做人的道理。企业相信，和睦的家庭关系对一个**人的成长有潜移默化的影响。**

### 就你申请的这个职位，你认为你还欠缺什么？

回答提示：企业喜欢问求职者弱点，但精明的求职者一般不直接回答。他们希望看到这样的求职者：继续重复自己的优势，然后说：“对于这个职位和我的能力来说，我相信自己是可以胜任的，只是缺乏经验，这个问题我想我可以进入公司以后以最短的时间来解决，我的学习能力很强，我相信可以很快融入公司的企业文化，进入工作状态。”企业喜欢能够巧妙地躲过难题的求职者。

### 你欣赏哪种性格的人？

回答提示：诚实、不死板而且容易相处的人、有"实际行动"的人。

### 你通常如何处理別人的批评？

回答提示：①沈默是金。不必说什么，否则情况更糟，不过我会接受建设性的批评；②我会等大家冷靜下来再讨论。

### 你怎样对待自己的失敗？

回答提示：我们大家生来都不是十全十美的，我相信我有第二个机会改正我的错误。

### 什么会让你有成就感？

回答提示：为贵公司竭力效劳；尽我所能，完成一个项目

### 眼下你生活中最重要的是什么？

回答提示：对我来说，能在这个领域找到工作是最重要的；望能在贵公司任职对我说最重要。

### 你为什么愿意到我们公司来工作？

回答提示：对于这个问题，你要格外小心，如果你已经对该单位作了研究，你可以回答一些详细的原因，像“公司本身的高技术开发环境很吸引我。”，“我同公司出生在同样的时代，我希望能够进入一家与我共同成长的公司。”“你们公司一直都稳定发展，在近几年来在市场上很有竞争力。”或者“我认为贵公司能够给我提供一个与众不同的发展道路。”这都显示出你已经做了一些调查，也说明你对自己的未来有了较为具体的远景规划。

### 你和别人发生过争执吗？你是怎样解决的？

回答提示：这是面试中最险恶的问题。其实是考官布下的一个陷阱。千万不要说任何人的过错。应知成功解决矛盾是一个协作团体中成员所必备的能力。假如你工作在一个服务行业，这个问题简直成了最重要的一个环节。你是否能获得这份工作，将取决于这个问题的回答。考官希望看到你是成熟且乐于奉献的。他们通过这个问题了解你的成熟度和处世能力。在没有外界干涉的情况下，通过妥协的方式来解决才是正确答案。

### 问题：你做过的哪件事最令自己感到骄傲?

回答提示：这是考官给你的一个机会，让你展示自己把握命运的能力。这会体现你潜在的领导能力以及你被提升的可能性。假如你应聘于一个服务性质的单位，你很可能会被邀请去午餐。记住：你的前途取决于你的知识、你的社交能力和综合表现。

### 你新到一个部门,一天一个客户来找你解决问题,你努力想让他满意，可是始终达不到群众得满意,他投诉你们部门工作效率低,你这个时候怎么作?

回答提示：(1)首先，我会保持冷静。作为一名工作人员，在工作中遇到各种各样的问题是正常的，关键是如何认识它，积极应对，妥善处理。 (2)其次，我会反思一下客户不满意的原因。一是看是否是自己在解决问题上的确有考虑的不周到的地方，二是看是否是客户不太了解相关的服务规定而提出超出规定的要求，三是看是否是客户了解相关的规定，但是提出的要求不合理。 (3)再次，根据原因采取相对的对策。如果是自己确有不周到的地方，按照服务规定作出合理的安排，并向客户作出解释；如果是客户不太了解政策规定而造成的误解，我会向他作出进一步的解释，消除他的误会；如果是客户提出的要求不符合政策规定，我会明确地向他指出。 (4)再次，我会把整个事情的处理情况向领导作出说明，希望得到他的理解和支持。(5)我不会因为客户投诉了我而丧失工作的热情和积极性，而会一如既往地牢记为客户服务的宗旨，争取早日做一名领导信任、公司放心、客户满意的职员。

### 对这项工作，你有哪些可预见的困难？”

回答提示：：①不宜直接说出具体的困难，否则可能令对方怀疑应聘者不行；②可以尝试迂回战术，说出应聘者对困难所持有的态度——“工作中出现一些困难是正常的，也是难免的，但是只要有坚忍不拔的毅力、良好的合作精神以及事前周密而充分的准备，任何困难都是可以克服。”  
分析：一般问这个问题，面试者的希望就比较大了，因为已经在谈工作细节。但常规思路中的回答，又被面试官“骗”了。当面试官询问这个问题的时候，有两个目的。第一，看看应聘者是不是在行，说出的困难是不是在这个职位中一般都不可避免的问题。第二，是想看一下应聘者解决困难的手法对不对，及公司能否提供这样的资源。而不是想了解应聘者对困难的态度。

### 如果我录用你，你将怎样开展工作？”

回答提示：①如果应聘者对于应聘的职位缺乏足够的了解，最好不要直接说出自己开展工作的具体办法；②可以尝试采用迂回战术来回答，如“首先听取领导的指示和要求，然后就有关情况进行了解和熟悉，接下来制定一份近期的工作计划并报领导批准，最后根据计划开展工作。”  
分析：这个问题的主要目的也是了解应聘者的工作能力和计划性、条理性，而且重点想要知道细节。如果向思路中所讲的迂回战术，面试官会认为回避问题，如果引导了几次仍然是回避的话。此人绝对不会录用了。

### “你希望与什么样的上级共事？”

回答提示：①通过应聘者对上级的“希望”可以判断出应聘者对自我要求的意识，这既上一个陷阱，又是一次机会；②最好回避对上级具体的希望，多谈对自己的要求；③如“做为刚步入社会的新人，我应该多要求自己尽快熟悉环境、适应环境，而不应该对环境提出什么要求，只要能发挥我的专长就可以了  
分析：这个问题比较好的回答是，希望我的上级能够在工作中对我多指导，对我工作中的错误能够立即指出。总之，从上级指导这个方面谈，不会有大的纰漏。

### 在完成某项工作时，你认为领导要求的方式不是最好的，自己还有更好的方法，你应该怎么做？

回答提示：  
①.原则上我会尊重和服从领导的工作安排；同时私底下找机会以请教的口吻，婉转地表达自己的想法，看看领导是否能改变想法；  
②如果领导没有采纳我的建议，我也同样会按领导的要求认真地去完成这项工作；  
③.还有一种情况，假如领导要求的方式违背原则，我会坚决提出反对意见；如领导仍固执己见，我会毫不犹豫地再向上级领导反映。

### 与上级意见不一是，你将怎么办？”

回答提示：①一般可以这样回答“我会给上级以必要的解释和提醒，在这种情况下，我会服从上级的意见。”②如果面试你的是总经理，而你所应聘的职位另有一位经理，且这位经理当时不在场，可以这样回答：“对于非原则性问题，我会服从上级的意见，对于涉及公司利益的重大问题，我希望能向更高层领导反映。”

分析：这个问题的标准答案是思路1，如果用2的回答，必死无疑。你没有摸清楚改公司的内部情况，先想打小报告，这样的人没有人敢要。

### “你工作经验欠缺，如何能胜任这项工作？”

常规思路：①如果招聘单位对应届毕业生的应聘者提出这个问题，说明招聘公司并不真正在乎“经验”，关键看应聘者怎样回答；②对这个问题的回答最好要体现出应聘者的诚恳、机智、果敢及敬业；③如“作为应届毕业生，在工作经验方面的确会有所欠缺，因此在读书期间我一直利用各种机会在这个行业里做兼职。我也发现，实际工作远比书本知识丰富、复杂。但我有较强的责任心、适应能力和学习能力，而且比较勤奋，所以在兼职中均能圆满完成各项工作，从中获取的经验也令我受益非浅。请贵公司放心，学校所学及兼职的工作经验使我一定能胜任这个职位。” 点评：这个问题思路中的答案尚可。突出自己的吃苦能力和适应性以及学习能力（不是学习成绩）为好。

### 您在前一家公司的离职原因是什么？”

回答提示：①最重要的是：应聘者要使找招聘单位相信，应聘者在过往的单位的“离职原因”在此家招聘单位里不存在；②避免把“离职原因”说得太详细、太具体；③不能掺杂主观的负面感受，如“太辛苦”、“人际关系复杂”、“管理太混乱”、“公司不重视人才”、“公司排斥我们某某的员工”等；④但也不能躲闪、回避，如“想换换环境”、“个人原因”等；⑤不能涉及自己负面的人格特征，如不诚实、懒惰、缺乏责任感、不随和等；⑥尽量使解释的理由为应聘者个人形象添彩；⑦相关例子：如“我离职是因为这家公司倒闭；我在公司工作了三年多，有较深的感情；从去年始，由于市场形势突变，公司的局面急转直下；到眼下这一步我觉得很遗憾，但还要面对显示，重新寻找能发挥我能力的舞台。”同一个面试问题并非只有一个答案，而同一个答案并不是在任何面试场合都有效，关键在应聘者掌握了规律后，对面试的具体情况进行把握，有意识地揣摩面试官提出问题的心理背景，然后投其所好。  
分析：除非是薪资太低，或者是最初的工作，否则不要用薪资作为理由。“求发展”也被考官听得太多，离职理由要根据每个人的真实离职理由来设计，但是在回答时一定要表现得真诚。实在想不出来的时候，家在外地可以说是因为家中有事，须请假几个月，公司又不可能准假，所以辞职。这个答案一般面试官还能接受。

### “你工作经验欠缺，如何能胜任这项工作？”

回答提示：①如果招聘单位对应届毕业生的应聘者提出这个问题，说明招聘公司并不真正在乎“经验”，关键看应聘者怎样回答；②对这个问题的回答最好要体现出应聘者的诚恳、机智、果敢及敬业；③如“作为应届毕业生，在工作经验方面的确会有所欠缺，因此在读书期间我一直利用各种机会在这个行业里做兼职。我也发现，实际工作远比书本知识丰富、复杂。但我有较强的责任心、适应能力和学习能力，而且比较勤奋，所以在兼职中均能圆满完成各项工作，从中获取的经验也令我受益非浅。请贵公司放心，学校所学及兼职的工作经验使我一定能胜任这个职位。”  
分析：这个问题思路中的答案尚可。突出自己的吃苦能力和适应性以及学习能力（不是学习成绩）为好。

### 为了做好你工作份外之事，你该怎样获得他人的支持和帮助？

回答提示：每个公司都在不断变化发展的过程中；你当然希望你的员工也是这样。你希望得到那些希望并欢迎变化的人，因为这些人明白，为了公司的发展，变化是公司日常生活中重要组成部分。这样的员工往往很容易适应公司的变化，并会对变化做出积极的响应。此外，他们遇到矛盾和问题时，也能泰然处之。下面的问题能够考核应聘者这方面的能力。  
据说有人能从容避免正面冲突。请讲一下你在这方面的经验和技巧。  
有些时候，我们得和我们不喜欢的人在一起共事。说说你曾经克服了性格方面的冲突而取得预期工作效果的经历。

### 如果你在这次面试中没有被录用，你怎么打算？

回答提示：现在的社会是一个竞争的社会,从这次面试中也可看出这一点,有竞争就必然有优劣,有成功必定就会有失败.往往成功的背后有许多的困难和挫折,如果这次失败了也仅仅是一次而已,只有经过经验经历的积累才能塑造出一个完全的成功者。我会从以下几个方面来正确看待这次失败.  
第一、要敢于面对,面对这次失败不气馁,接受已经失去了这次机会就不会回头这个现实,从心理意志和精神上体现出对这次失败的抵抗力。要有自信,相信自己经历了这次之后经过努力一定能行.能够超越自我.  
第二、善于反思,对于这次面试经验要认真总结,思考剖析,能够从自身的角度找差距。正确对待自己,实事求是地评价自己,辩证的看待自己的长短得失,做一个明白人.  
第三、走出阴影,要克服这一次失败带给自己的心理压力,时刻牢记自己弱点,防患于未然,加强学习,提高自身素质.  
第四、认真工作,回到原单位岗位上后,要实实在在、踏踏实实地工作,三十六行,行行出状元,争取在本岗位上做出一定的成绩.  
第五、再接再厉,成为国家公务员一直是我的梦想,以后如果有机会我仍然后再次参加竞争.

### 假如你晚上要去送一个出国的同学去机场，可单位临时有事非你办不可，你怎么办？

回答提示：我觉得工作是第一位的，但朋友间的情谊也是不能偏废的。这个问题我觉得要按照当时具体的情况来决定。  
（1）、如果我的朋友晚上9点中的飞机，而我的 加班八点就能够完成的话，那就最理想了，干完工作去机场，皆大欢喜。  
（2）、如果说工作不是很紧急，加班仅仅是为了明天上班的时候能把报告交到办公室，那完全可以跟领导打声招呼，先去机场然后回来加班，晚点睡就是了。  
（3）、如果工作很紧急，两者不可能兼顾的情况下，我觉得可以由两种选择。1）如果不是全单位都加班的话，是不是可以要其他同事来代替以下工作，自己去机场，哪怕就是代替你离开的那一会儿。2）如果连这一点都做不到的话，  
那只好忠义不能两全了，打电话给朋友解释一下，小心他会理解，毕竟工作做完了就完了，朋友还是可以再见面的。

### 如果通过这次面试我们单位录用了你，但工作一段时间却发现你根本不适合这个职位，你怎么办？

回答提示：一段时间发现工作不适合我，有两种情况：  
1、如果你确实热爱这个职业，那你就要不断学习，虚心向领导和同事学习业务知识和处事经验，了解这个职业的精神内涵和职业要求，力争减少差距；  
2、你觉得这个职业可有可无，那还是趁早换个职业，去发现适合你的，你热爱的职业，那样你的发展前途也会大点，对单位和个人都有好处。

### 你做过的哪件事最令自己感到骄傲?

回答提示：这是考官给你的一个机会，让你展示自己把握命运的能力。这会体现你潜在的领导能力以及你被提升的可能性。假如你应聘于一个服务性质的单位，你很可能会被邀请去午餐。记住：你的前途取决于你的知识、你的社交能力和综合表现。

### 谈谈你过去做过的成功案例

回答提示：举一个你最有把握的例子，把来龙去脉说清楚，而不要说了很多却没有重点。切忌夸大其词，把别人的功劳到说成自己的，很多主管为了确保要用的人是最适合的，会打电话向你的前一个主管征询对你的看法及意见，所以如果说谎，是很容易穿梆的。

### 谈谈你过去的工作经验中，最令你挫折的事情

回答提示：曾经接触过一个客户，原本就有耳闻他们以挑剔出名，所以事前的准备功夫做得十分充分，也投入了相当多的时间与精力，最后客户虽然并没有照单全收，但是接受的程度已经出乎我们意料之外了。原以为从此可以合作愉快，却得知客户最后因为预算关系选择了另一家代理商，之前的努力因而付诸流水。尽管如此，我还是从这次的经验学到很多，如对该产业的了解，整个team的默契也更好了。  
分析：借此了解你对挫折的容忍度及调解方式。

### 如何安排自己的时间？会不会排斥加班？

回答提示：基本上，如果上班工作有效率，工作量合理的话，应该不太需要加班。可是我也知道有时候很难避免加班，加上现在工作都采用责任制，所以我会调配自己的时间，全力配合。  
分析：虽然不会有人心甘情愿的加班，但依旧要表现出高配合度的诚意。

### 为什么我们要在众多的面试者中选择你？

回答提示：根据我对贵公司的了解，以及我在这份工作上所累积的专业、经验及人脉，相信正是贵公司所找寻的人才。而我在工作态度、ＥＱ上，也有圆融、成熟的一面，和主管、同事都能合作愉快。  
分析：别过度吹嘘自己的能力，或信口开河地乱开支票，例如一定会为该公司带来多少钱的业务等，这样很容易给人一种爱说大话、不切实际的感觉。

### 对这个职务的期许？

回答提示：希望能借此发挥我的所学及专长，同时也吸收贵公司在这方面的经验，就公司、我个人而言，缔造“双赢”的局面。  
分析：回答前不妨先询问该公司对这项职务的责任认定及归属，因为每一家公司的状况不尽相同。以免说了一堆理想抱负却发现牛头不对马嘴。

### 为什么选择这个职务？

回答提示：：这一直是我的兴趣和专长，经过这几年的磨练，也累积了一定的经验及人脉，相信我一定能胜任这个职务的。  
分析：适时举出过去的“丰功伟业”，表现出你对这份职务的熟稔度，但避免过于夸张的形容或流于炫耀。

### 为什么选择我们这家公司？

回答提示：曾经在报章杂志看过关于贵公司的报道，与自己所追求的理念有志一同。而贵公司在业界的成绩也是有目共睹的，而且对员工的教育训练、升迁等也都很有制度。  
分析：去面试前先做功课，了解一下该公司的背景，让对方觉得你真的很有心想得到这份工作，而不只是探探路。

### 你认为你在学校属于好学生吗？

回答提示：企业的招聘者很精明，问这个问题可以试探出很多问题：如果求职者学习成绩好，就会说：“是的，我的成绩很好，所有的成绩都很优异。当然，判断一个学生是不是好学生有很多标准，在学校期间我认为成绩是重要的，其他方面包括思想道德、实践经验、团队精神、沟通能力也都是很重要的，我在这些方面也做得很好，应该说我是一个全面发展的学生。”如果求职者成绩不尽理想，便会说：“我认为是不是一个好学生的标准是多元化的，我的学习成绩还可以，在其他方面我的表现也很突出，比如我去很多地方实习过，我很喜欢在快节奏和压力下工作，我在学生会组织过 ××活动，锻炼了我的团队合作精神和组织能力。” 有经验的招聘者一听就会明白，企业喜欢诚实的求职者。

### 请谈谈如何适应办公室工作的新环境？

回答提示①办公室里每个人有各自的岗位与职责，不得擅离岗位。  
②根据领导指示和工作安排，制定工作计划，提前预备，并按计划完成。  
③多请示并及时汇报，遇到不明白的要虚心请教。  
④抓间隙时间，多学习，努力提高自己的政治素质和业务水平。

### 在工作中学习到了些什么？

回答提示：这是针对转职者提出的问题，建议此时可以配合面试工作的特点作为主要依据来回答，如业务工作需要与人沟通，便可举出之前工作与人沟通的例子，经历了哪些困难，学习到哪些经验，把握这些要点做陈述，就可以轻易过关了

### 有想过创业吗？

回答提示：这个问题可以显示你的冲劲，但如果你的回答是“有”的话，千万小心，下一个问题可能就是“那么为什么你不这样做呢？”

### 最能概括你自己的三个词是什么？

回答提示：我经常用的三个词是：适应能力强，有责任心和做事有始终，结合具体例子向主考官解释，使他们觉得你具有发展潜力

### 你认为你在学校属于好学生吗？

回答提示：企业的招聘者很精明，问这个问题可以试探出很多问题：如果求职者学习成绩好，就会说：“是的，我的成绩很好，所有的成绩都很优异。当然，判断一个学生是不是好学生有很多标准，在学校期间我认为成绩是重要的，其他方面包括思想道德、实践经验、团队精神、沟通能力也都是很重要的，我在这些方面也做得很好，应该说我是一个全面发展的学生。”如果求职者成绩不尽理想，便会说：“我认为是不是一个好学生的标准是多元化的，我的学习成绩还可以，在其他方面我的表现也很突出，比如我去很多地方实习过，我很喜欢在快节奏和压力下工作，我在学生会组织过 ××活动，锻炼了我的团队合作精神和组织能力。” 有经验的招聘者一听就会明白，企业喜欢诚实的求职者。

### 除了本公司外，还应聘了哪些公司？

回答提示：很奇怪，这是相当多公司会问的问题，其用意是要概略知道应徵者的求职志向，所以这并非绝对是负面答案，就算不便说出公司名称，也应回答“销售同种产品的公司”，如果应聘的其他公司是不同业界，容易让人产生无法信任的感觉。

### 何时可以到职？

回答提示：大多数企业会关心就职时间，最好是回答\’如果被录用的话，到职日可按公司规定上班”，但如果还未辞去上一个工作、上班时间又太近，似乎有些强人所难，因为交接至少要一个月的时间，应进一步说明原因，录取公司应该会通融的

### 你并非毕业于名牌院校？

回答提示：是否毕业于名牌院校不重要，重要的是有能力完成您交给我的工作，我接受了北大青鸟的职业培训，掌握的技能完全可以胜任贵公司现在工作，而且我比一些名牌院校的应届毕业生的动手能力还要强，我想我更适合贵公司这个职位。

### 你怎样看待学历和能力？

回答提示：学历我想只要是大学专科的学历，就表明觉得我具备了根本的学习能力。剩下的，你是学士也好，还是博士也好，对于这一点的讨论，不是看你学了多少知识，而是看你在这个领域上发挥了什么，也就是所说的能力问题。一个人工作能力的高低直接决定其职场命运，而学历的高低只是进入一个企业的敲门砖，如果贵公司把学历卡在博士上，我就无法进入贵公司，当然这不一定只是我个人的损失，如果一个专科生都能完成的工作，您又何必非要招聘一位博士生呢？

### 你经历太单纯，而我们需要的是社会经验丰富的人？

回答提示：经历丰富的人也未必适合这个职位，如果他在以前的经历中养成的是一个良好的职业习惯还好，如果是不好的职业习惯呢？我在这方面是一片空白，更可以尽快地融入贵公司的企业文化，养成良好的职业习惯，一个人具有良好的职业习惯，更会发挥自己的长处为公司做更多的事。

### 你性格过于内向，这恐怕与我们的职业不合适？

回答提示：没关系呀，性格内向可以踏实地完成技术工作。况且性格内向并不能说明我无法讲述清楚我做的项目，克林顿小时候性格还内向呢，并不能阻止他成为美国总统。

### 假如领导派你和一个有矛盾的同志一起出差，你如何处理?在日常生活中，出现这样的事情你是如何处理的？试举例说明。

回答提示：在日常生活中，由于每个人的观点和立场，看待和分析问题的方法不同，矛盾是不可避免的。假如我和一个和我有矛盾的同志一起出差，我想首先应该开诚布公，因为有矛盾，就把许多东西隐藏起来，这只会加深误会，假如互相坦诚相见，以一种客观，不带个人情绪的态度看问题，你会发现你原先自认为十分得意的想法并不完全正确，你先前反对的看法和观点可能只是一个事物的另一侧面，你们原来在许多方面可以互相补充，互相完善。我个人认为在处理矛盾问题上要有一种宽容的态度，俗话说的好：宰相肚里能撑船。心胸狭隘是化解矛盾的大敌，而一个心胸狭隘的人是绝不可能成就一番大事业的。  
我在上大学时，同寝室一个同学喜欢在寝室随地吐痰，我很不喜欢这一点，但他脾气比较暴躁，如果我直接向他说，矛盾就会激化，对此我采取了以下方法：一是通过别的同学委婉地表达我对他随地吐痰的反对态度；二是经常拿一些有关随地吐痰危害自己和他人健康的书籍放在寝室的桌子上，使他能够看到；三是我在他在寝室的时候也吐痰，但是不在寝室里吐，而是在外边吐，暗示其应到外边吐痰.

### 如果你遇到了挫折你将怎么办?

回答提示：事业有成一帆风顺时许多人的美好想法,其实很难做到一帆风顺,要接受这样一个现实,  
人的一生不可能是一帆风顺的,成功的背后会有许许多多的艰辛,痛苦甚至挫折.在人生的一段时期遇到一些挫折是很正常的.只有经验知识和经历的积累才能塑造出一个成功者.我觉得面对挫折要做到以下几点:  
1）第一要敢于面对.哪里跌倒要从哪里爬起来,小平同志还是三起三落呢,不要惧怕困难,要敢于向困难挑战.  
2）再者要认真分析失败的原因,寻根究源,俗话说失败乃成功之母,在挫折中掌握教训,为下一次奋起提供经验.  
3）还有在平时的工作生化中要加强学习,人的一生是有限的,不可能经历所有的事,要在别人的经验吸取教训.  
4）最后可能由于当局者迷或者知识经历的不足,自己对于挫折并没有特别好的处理方法,这是可以求教自己的亲人朋**友,群策群力渡过难关。**

### 你最喜欢的一本书是那本?

回答提示：我喜欢读书,一个人最早看的一本书可能会对个人的一生产生很大的影响,我小时候最早看的一本书是三国演义,三国演义这本书博大精深,书中描写的一些人物我对我的成长起了许多潜移默化的作用,现在看来我还是最喜欢三国演义如果我说我喜欢关羽,可能俗了一点,但从关羽身上表现出来的诚信和忠诚一直是我很推崇的。我觉得诚心是立身之本,而对单位的忠诚是你能不能做出一番事业的前提条件。当然这个忠诚还包括对领导的忠诚.  
从周瑜身上我学到对别人要宽容,不要又嫉妒心；从诸葛亮身上学到要加强自己学习等等。三国演义这本书博大精深,对我的影响也是全方位的，时间原因我不再赘述。

### 请告知你的工作观？

回答提示：常被问到“你的\*\*观是什么？”时，可别把它想得太复杂，可回答“为何而工作”、“从工作方面得到了什么”“\*年後想变成怎样”等的话。

### 如果你有一位固执武断的领导，你会经常提合理化建议吗？

回答提示：①在一般情况下，领导和同事是不能选择的，每个人有每个人的个性和脾气，要学会适应和相处。  
②领导脾气直也好，悠也好，固执也好，只要是出自为公，为工作，应该尊重和原谅他，并且按他的安排去做。  
③适当的时候，可以用谈心、汇报思想等方式委婉地提出自己的看法，但点到为止。  
④在有合理化的建议时，照提不误。因为那是对自己和工作负责。

### 假如在一次学习讨论会上，领导对你的发言很不满意，并当场批评了你，你怎么办?

回答提示：  
好：能谈到在情绪产生波动时，自己的自我控制方法，并取得良好的效果。  
中：有控制自我情绪的努力，但方法不够有效。  
差：思前想后，顾虑重重，言语中流露出气愤、委屈，或长时间沉默，显得心情难以言表。

### 针对你们单位业务工作中出现的问题，你提出了一些很好的建议，得到了同事们的赞同，但你的领导并不满意。在这种情况下，你怎么办?

**回答提示：**好：能认真分析，仔细推敲自己建议存在的问题，认真领会领导意图，找出两者之间的共同点，圆满解决问题，考生心理冷静，自制力强，方法正确，陈述问题条理清楚，有说服力。  
中：能找出问题的根源并冷静对待，处理问题有一定方法，陈述问题有一定说服力。  
差：行为偏激，自制力差，方法不正确，或长时间沉默，显得无以应对。

### 假设你手头上有好几项工作没有完成，可是上级又给你安排了一项任务。你感到自己完成这项工作有困难。你如何处理这个矛盾?

回答提示：出题思路：情境性问题。该题用于考查考生的人际交往的意识与技巧，主要是在组织中处理权属关系的能力。  
　　参考评分标准：  
　　好：能够很好地与人沟通，有很好的交往方法和技巧；能够在尊重他人的前提下恰当地表达自己的意见。如：能用适当方法让领导了解到自己现有任务已很重，并能向领导提出完成该任务的可行建议。  
　　中：能与人沟通，交往中有一些方法、技巧，能适当地表达出自己的意见。如：能让领导认识到自己的困难。  
　　差：不能与他人沟通，交往中缺乏技巧；难以表达自己的意见。如：无法让领导了解自己的困难，或直接顶撞领导。

### 当前对有些单位实施的‘末位淘汰制’，有不同争议，你怎么看待这种用人措施？”

回答提示： “末位淘汰制”是一种向竞争机制发展的过渡性措施，可以试行；但要因情况而异，不能一刀切。再说“末位淘汰制”也不完全等同于竞争机制。对于规模较大、人数较多的单位最初实行，然后实施竞争机制，未尝不可。如果在规模小、人数少的单位实行，效果就不一定好，因为也确有些单位人数不多，几乎所有人员都很努力，成绩都不错，甚至难分上下，如果实行就会造成人心惶惶、人际关系紧张的不利局面。

### 物质待遇和工作条件是人们选择工作的重要因素之一，这次报考谈一谈你在选择工作时都考虑哪些因素?为什么?

回答提示：好：既能够坦率承认在择业过程中的个人利益因素，又能够超越物质利益因素，有正确的择业观，积极向上，叙述有条理，具有说服力。  
　　中：基本能够正视现实，实事求是地考虑择业问题，叙述问题基本清楚，有一定说理性。差：隐瞒自己的真实想法，假、大、空，或者过分注重自己的个性需求，叙述无条理，说理性差。

### 直接领导要求你在30日内完成一项工作，你会怎样去完成？

回答提示：提前做准备，制订完备的计划，会准时甚至更早地完成工作任务。

### 你认为这份工作最重要的是什么？

回答提示：叙述工作特性的同时，也要加上自己的看法，如果是有工作经验的人，最好说明自己的基本心态。

### 上班的时候，往往有多件事需要你处理，你会如何安排？

回答提示：事有三件，紧要处着手。要先处理重要事件，其他依次统筹安排

### 找这份工作，你最重要的考虑因素为何？

回答提示：如回答：“工作的性质是否能让我发挥所长，并不断成长。”因为公司要找工作表现好、能够真正有贡献的人，而非纯粹慕名、求利而来的人。

### 在你参加我们这次面试前，你做了哪些准备工作？

回答提示：公司的背景知识及岗位要求；可能要问的问题（技术+技巧）；调整状态、保持一颗平常心等

### 你争取成功的动力是什么？

回答提示：自我实现，展现自身的价值。

### 怎样理解团队？请举例并说明启示。

回答提示：团队成员共同承担领导职能，共同努力，以各自独特的方式，在所处的环境中共同完成预先设定的目标。（明确的目标、勇于负责、协作增效、能力资源各异）  
例如：大雁团队，其合作精神体现在以下几个方面：  
（1）大雁会共同“拍动翅膀”。拍翅膀是大雁的本能，但只要排成人字队形，就可以提高飞行效率。（2）所有的大雁都愿意接受团体的飞行队形，而且都实际协助队形的建立。如果有一只大雁落在队形外面，它很快就会感到自己越来越落后，便会立即回到雁群中。（3）大雁的领导工作是由群体共同分担的。虽然有带头雁出来整队，但是当它疲倦时，便会自动退到队伍之中，另一只大雁马上替补领头的位置。（4）队形后边的大雁不断发出鸣叫，目的是为了给前方的伙伴打气激励。（5）如果一只大雁生病或被猎人击伤，雁群中就会有两只大雁脱离队形，靠近这只遇到困难的同伴，协助它降落在地面上，直至它能够重回群体，或是不幸死亡。  
通过分析大雁的合作行为，我们可以得出以下启示：  
（1）每个人都要忠诚于自己的团队，忠诚于自己的事业，做好自己的本职工作。如果你不拍翅膀，他不拍翅膀，这个团体还会存在吗？（2）如果我们与大雁一样聪明的话，我们就会留在与自己的事业目标一致的队伍里，而且乐意接受他人的协助，也愿意协助他人。（3）我们必须确定从背后传来的是鼓励的叫声，而不是其他声音。想要在职业生涯中生存和发展，需要把工作伙伴变成啦啦队，一队快乐的工作伙伴是成功最好的助手。你的工作伙伴散播的有利消息远比你个人所有的努力更有助于你职业生涯的发展

### 为什么下水道的井盖是圆的？

回答提示：圆的井盖立起来时不会掉到井里。

### 如果你是公司老板，最不喜欢的是哪种员工？

回答提示：

1.遇到问题事不关己。  
2.凡事三缄其口。  
3.总是墨守城规，没有创新。  
4.背后议论上司。  
5.随遇而安。  
6.追名逐利。  
7.拒绝加班。  
8.不善交际。  
9.既然预定了假期就不可改变。

### 面试进行时，大脑出现短路怎么办？

回答提示：

1、坦白的讲，我对这个问题不太了解。但我对XX问题倒是有一些研究。我可以讲讲这方面的内容吗？  
2、您的意思是………？（进一步确认）  
3、”不知道您看出来没有，我太重视这次面试，因此有些紧张，您能给我一分种的时间让我考虑一下这个问题好吗”（如果你确信，多给你一些时间，你可以考虑出来）  
4、很坦白的讲，对于这个问题我不太了解。（没有人可以知道所有的知识）  
给出解决问题的方法所设计的知识点。（虽然不知道怎么解决，你会按照什么思路解决）

### 怎样与上级沟通?

回答提示：  
1）仔细聆听领导的命令  
2）与领导探讨目标的可行性  
3）拟定详细的工作计划  
4）在工作进行之中随时向领导汇报  
5）在工作完成后及时总结汇报

### 你为什么觉得自己能够在这个职位上取得成就？

回答提示：分析这是一个相当宽泛的问题，它给求职者提供了一个机会，可以让求职者表明自己的热情和挑战欲。对这个问题的回答将为面试人在判断求职者是否对这个职位有足够的动力和自信心方面提供关键信息。  
错误回答我不知道。我擅长做很多事情。如果我能得到并且决定接受这份工作，我确信自己可以把它做得相当好,因为我过去一直都很成功。  
正确回答从我的经历来看，这是我的职业生涯中最适合我的一份工作。几年来，我一直在研究这个领域并且关注贵公司，一直希望能有这样的面试机会。我拥有必备的技能（简单讲述一个故事来加以说明），我非常适合这一职位，也确实能做好这份工作。

### 你最大的长处和弱点分别是什么？这些长处和弱点对你在企业的业绩会有什么样的影响？

回答提示：这个问题的最大陷阱在于，第一个问题实际上是两个问题，而且还要加上一个后续问题。这两个问题的陷阱并不在于你是否能认真地看待自己的长处，也不在于你是否能正确认识自己的弱点。记住，你的回答不仅是向面试人说明你的优势和劣势，也能在总体上表现你的价值观和对自身价值的看法。  
     错误回答从长处来说，我实在找不出什么突出的方面，我认为我的技能是非常广泛的。至于弱点，我想，如果某个项目时间拖得太久，我可能会感到厌倦。  
　　正确回答从长处来说，我相信我最大的优点是我有一个高度理性的头脑，能够从混乱中整理出头绪来。我最大的弱点是，对那些没有秩序感的人，可能缺乏足够的耐心。我相信我的组织才能可以帮助企业更快地实现目标，而且有时候，我处理复杂问题的能力也能影响我的同事。  
      评论这个回答做到了“一箭三雕”。首先，它确实表明了求职者的最大长处。其次，它所表达的弱点实际上很容易被理解为长处。最后，它指出了这个求职者的长处和弱点对企业和其他员工的好处。

### 请谈谈你个人的最大特色。

回答提示：“我的坚持度很高，事情没有做到一个令人满意的结果，绝不罢手。这种回答，最能和工作结合，能够与工作表现相结合的优点、特质，才是面谈者比较感兴趣的回答。

### 关于你的个人隐私(是否有男/女朋友，或是否结婚)

回答提示：在大多数情况下，面谈者会竭力地打探证明你不稳定或不可靠的信息，还有其他一些可能使某个雇主关注的问题，这些只是对某些性格的人的推测，都是些不相关的问题，但是，如果雇主想以此来了解你是否可靠，你就得全力以赴地去应付了。要记住即使是随意地闲谈也要避免提及隐晦的问题。在回答个人情况时，要态度友好而且自信

### 你说一下对\*\*专业内容的了解，对此专业应用领域的了解？

回答提示：如果是精通的，一定要详细阐述；如果只是了解，也要对其中掌握的进行一下简单描述；如果确实不知道，就说“对不起，不是很了解”。因为没有一个什么都知道。

### 你认为求职，穿什么去面试合适？

回答提示：  
女生：建议穿白色有领衬衣，配长过膝盖的中裙或西装裤，可准备一件质地较好的外套方便穿脱，色彩以淡雅为宜。  
    关键词：优雅、美丽、精致  
    忌：嘻哈风、黑白灰、低胸露背装  
    宜：淡妆、裙装、明快色系、精致配饰  
男生：在色彩和款式上，要重在表达亲近感，增强亲和力。  
　　关键词：稳重、整洁  
　　忌：T恤衫＆牛仔裤  
    　宜：衬衫＆西裤

# 面经汇总

左鹏的涂鸦移动1、 2面：https://www.nowcoder.com/discuss/38363

20个高级Java面试题汇总

<http://www.imooc.com/article/20702>

## 涂鸦移动面经

一面:

1. 给定一个大于10的整数N，要求随机分为5个数，这五个数的和为N
2. 给定一个有序整数数组arr，给定一个整数N，求在这个数组arr中的两个数a和b使得a+b=N。

二面：

1. 一个小根堆， 来一个数，如何加入这个小根堆。
2. 给定完全二叉树的根节点，求该树的节点数。（复杂度尽量低）
3. 给定一个字符串s， 另一个字符串t，判断s中是否存在t。复杂度尽量低。
4. 一个数轴上，有n个点， 求距离这n个点的平方和最小的点。
5. 给定一个矩阵（m\*n），每个元素存放一个字符，再给定一个字典。从矩阵指定位置的字母到第i，j位置字母所经历的路径中，字母组成的单词，求这些单词所构成的集合。

## 阿里巴巴面经

早上九点，先是做一小段自我介绍，然后面试开始,先确定了最熟悉的语言是JAVA，

1ArrayList底层是如何实现的

2集合类有了解吗，说一说map这个类，有几种实现，包括了HashMap，TreeMap，HashTable等等，HashMap底层如何实现

3说一说了解的排序算法，然后问了各种算法的时间复杂度，包括最坏情况以及平均情况，快排

4二叉树，讲一讲有哪些性质，二叉平衡树（AVL），红黑树的性质，以及红黑树高度最高是多少，可以用性质中的哪一条来证明

5算法问题，一个整数数组里面，只有一个数字只出现过一次，其他的数字都是出现两次，如何最快的找到这个数字是什么，不用找到index

6看到项目上写了一个Kaggle竞赛，然后问了一下是单人完成的吗，讲了讲项目的实现过程，以及运用到了哪些模型，学习过哪些模型，然后说一说贝叶斯模型的公式，也就是那个条件概率的公式，然后问到KNN模型需不需要训练，GBDT是什么，有哪些性质，Decision Tree有哪些判断的标准，ID3，C4.5， CART 包括了熵，GAIN RATIO, GINI系数等等，

7问我有什么问题

8推荐两本书，一本技术，一本非技术，介绍一下书中的基本内容

9闲聊。。。

面试主要考察JAVA的一些基础知识和一些基本的算法，然后看看你对自己项目相关知识是否了解清楚，整个过程大概80分钟，不过感觉最后闲聊了20分钟。

## 腾讯面经

1、手写代码 判断是否是浏览器环境。

2、http和https https的整个过程画一下

3、httPS需要多少时间 比http慢多少 怎么优化

4、https有什么缺点

5、手写快速排序 复杂度 最慢是多少 为什么取第一个值作为中间值 如果这个值是最大的或者最小的怎么办

6、http2有哪些特性 头部压缩怎么回事

7、100层 1个花瓶仍 找到n层不碎 n + 1层碎了的情况 2个花瓶呢

8、node了解多少 全局对象是什么

9、事件机制说一下 事件绑定说一下 手写如何把div中的a放在平级其中的事件会怎么样 冒泡会到document还是window

10、推荐一下身边厉害的人，你觉得你比他厉害吗。为什么

11、xss怎么防范 尖括号应该怎么过滤，一直问确定吗，最后也没说清楚

12、用过哪些框架

13、移动端如何真机调试

14、tap是怎么回事、和click的区别。。。

## 百度面经

一面:

1.进程和线程的区别；

2.手写一段代码；

3.tcp三次握手；

4.openGL渲染管线；

5.25台电脑，4口排插，一个电源，要多少个排插？n个呢？

6.放盘子问题；

7.openGL渲染管线，五个坐标系；

8.10个线程都对1个变量进行加加操作，结果是多少；1000个呢？

9.死锁的原因和解决方案；

10.知道哪些排序算法，复杂度分别是多少，对一个单链表升序，选哪个好？

二面：

1.violate关键字;

2.Activity启动模式；

3.进程间通讯的方式；

4.http和https

5.service的两种模式;

6.view事件分发机制；

7.线程池；

8.布局优化；

9.Android热修复

10.C++的类在内存中的大小

三面：

1.手写自己项目的框架;

2.性能优化；

3.插件化开发；

4.jni优化；

5.ndk守护进程；

6.职业规划；

7.看过哪些书，有没真正沉下心来看过一本书；

8.第三方框架再封装的必要性；

9.跟同事方案有冲突怎么办？

10.给你一个没接触过的项目，一个星期内完成，怎么做；

11.手头有没有offer；

12.对百度文化的了解；

## 三七互娱面经

一面：

1. 集合
   1. list，map，set的区别，有哪些实现类
   2. 线程安全的list和map有哪些
   3. treeset和hashset的区别
   4. 使用treeset和hashset的场景
   5. linkedhashmap的使用场景，问我linkedhashmap的使用场景，我说可以用于LRU算法，竟然被面试官反驳了。。。根据数据结构进行解释，还是被反驳。。。
2. 设计模式
   1. 画一画代理模式的uml图
   2. 工厂模式和建造者模式的区别
   3. 简单工厂模式、抽象工厂模式、工厂方法模式的区别。
3. Spring
   1. 介绍一下spring ioc和aop，实现原理
4. JVM
   1. jvm怎样判断一个对象是否可回收，怎样的对象才能作为GC root
5. java并发
   1. java中有哪几种锁，同步锁和显式锁的区别，使用场景(自旋锁那些)
   2. ExecutorService的实现类有哪些，只介绍了ThreadExecutorPool
6. java基础
   1. 泛型的T,E,?和空的区别
   2. 怎样自定义注解
7. 计算机基础
   1. tcp/ip、http和https的区别
   2. 什么是死锁，用代码写一个死锁
   3. 带宽和传输速度的关系
   4. 一般局域网的带宽有多少
   5. 硬盘和内存读写速度是多少

可能计算机基础学得不好，硬件和内存的读写速度真答不上来，面试官还问我是什么专业的，尴尬

接着等了半个钟，才二面，技术总监面

二面：

1. 自我介绍
2. 说说项目中的难点
3. 两题算法题：
   1. 关于括号的算法题，用栈
   2. 最长字符串匹配，用tire树
4. spring ioc与aop的实现
5. 原对象中两个方法，方法a与方法b。使用spring aop对该对象进行增强处理，增强处理都为都是输出一条日志。在代理对象的a方法中调用b方法，会输出几条日志。（答案是一条日志，因为代理对象a方法调用的是原对象的b方法，而不是代理对象的b方法）
6. 有了解过你应聘的这个岗位的工作吗？与你的匹配吗
7. 了解过游戏公司的工作机制吗
8. 你认为公司加班的主要原因是什么
9. 如何避免加班

Hr面：

1. 期待薪资
2. 自己的缺点
3. 不能忍受舍友的哪些做法
4. 什么时候能入职
5. 还会回原来实习的那家公司吗

相对来说，互联网公司的hr面还是比较干脆的，没有那么多套路。

不会像某些公司一样，会问一些“阿里和我的公司选哪个”，“什么情况会让你拒绝我们公司的offer“这种问题。

## 杭州有赞面经

 自我介绍

 项目，简单聊了聊

 介绍单例模式及其实现

 介绍四次挥手的过程

 TCP和UDP是这么实现的？不知道，回答了TCP和UDP的区别

 同步代码块问题？具体记不清了，说了synchronized和ReentrantLock的用法和区别

 HTTP请求，说一下每部分包含哪些内容

 POST大小限制

 Linux方面，top作用，ps的作用？（这个没听清）

 介绍一下B树和B+树

 OOM什么时候会发生？

 新生代和老年代的区别

 垃圾回收算法有哪些？说一下复制算法

 介绍一下隔离界别，什么是幻读？

 使用 like 模糊查询所以会失效吗？

 有什么想了解的？

## 美团面经

美团面试还是有深度的。

1、你还转载过美团的文章？嗯 （很多面试官还是会看博客的，谢谢博客总是好的）

2、cookie和localStorage区别，如何把cookie写在一个对象中，其属性就是键值对

3、实现水平竖直居中的几中方法（高度和宽度不确定）

4、flex是什么

5、深拷贝和浅拷贝的区别，手写深拷贝 （博客里总结过，遂让手撕）

6、Oject的几中方式，如何区分 （这个也是常考的）

7、兼容性说一说，你做的PC端兼容性是到哪的

8、如何看待最近react事件 （百度、wordpress弃用事件。可以多关注前端的一些动态）

9、JQuery和三大框架的区别，react和vue的区别

10、说一说CSS3中的动画，animation中可以取哪些值 （css3动画也很重要）

11、flex布局是什么，默认的方向是什么，如何改变方向

12、你的博客都是自己写的吗？ 你所有所学都会写成博客吗？

13、说一说原型和原型链，object是最上面的吗？

14、vue中的数据双向绑定是如何实现的、

15、display有哪些，回答了7 8个，还是不满意

16、你的node项目中有几个线程，有几个进程，如果node进程崩溃了，如何知道

17、node中的process是什么

18、node中如何创建子进程/19讲一讲你遇到的比较困难的事情，问了两个，最后是怎么解决的

19、CSS中的单位有哪些分别说一说 （提到了ex）

20、你未来的发展规划是怎么样的

21、跨域，举一个例子，安全性问题到底在哪里

22、100万个数据中，找出前10大数据，最快的方法是什么，堆排序怎么写

23、登录状态怎么使用cookie保持，最好的方法是什么

24、页面中一个video，可能格式不支持，那么前端如何判断并给出提示？

25、通过比较img的onerror是一种方法， 通过服务器端也可以保存一些属性来标识哪些浏览器支持，哪些不支持

26、为什么要用vue和react，而不用JQuery，给你一个项目，什么情况下使用单页面应用，什么情况下使用多页面应用。

27、你所知道的前端大会有哪些，比如腾讯web前端大会，CSS conf大会，js conf大会。

28、你学习的前端号有哪些，比如前端早读课，那么维护者是谁。

29、前端的表单中如何设置表单的方式，如multipart, www等，对于multipart具体是如何区分其中的不同的格式的

30、http2的所有特性。

31、http2中的头部压缩和多路复用。

三面主要是拓展，会比较难。

## 猎豹移动面经

java搞一个LRU

写一个gc，重点是思路

java搞一个优先队列（别用现成的）

手写一个死锁

## 小米面经

TCP/IP各层协议报头标志位；

MySQL的InnoDB引擎关于索引的实现，包括B+树的每个节点存放的是什么东西；

Trie树；

跳跃表。

## CVTE面经

一面：

1.Activity启动模式;

2.view绘制流程;

3.measure三种模式，textview属于哪种；

4.Activity A跳转到Activity B，生命周期变化，哪些情况不会调用onStop,onStart和onResume有什么区别。

5.service两种模式，如果用bind的模式，Activity被销毁，但是这之前没unbind会怎样；

6.设计模式；

7.两种广播的注册，有序广播；

8.自定义view；

9.view事件分发机制,父view拦截子view事件，子view拦截父view事件

10.二叉树的遍历;

二面：

1.长连接和短连接；

2.httpClient框架；

3.MVP,MVC;

4.怎么分析内存抖动，内存泄露，cpu

5.Message消息池；

6.OpenGL ES点和片段着色器作用，1.0和2.0区别；

7.OpenGL五个坐标系；

8.远程写代码，实现栈，考虑扩容，线程安全，泛型；

9.职业规划；

10.view事件分发机制；

11.handler机制;

12.leackcanary原理;

13.设计模式；

14.线程池有哪几种，参数意义；

15.说项目框架；

16.有没github开源项目,写过博客没;

17.滑动冲突