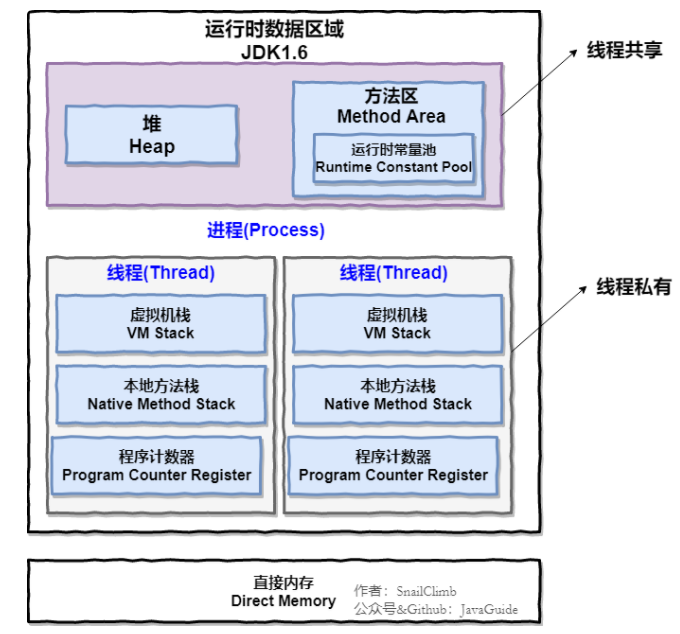
1、进程与线程

线程与进程相似，但线程是一个比进程更小的执行单位。一个进程在其执行的过程中可以产生多个线程。与进程不同的是同类的多个线程共享进程的堆和方法区资源，但每个线程有自己的程序计数器、虚拟机栈和本地方法栈，



程序计数器私有主要是为了**线程切换后能恢复到正确的执行位置**。

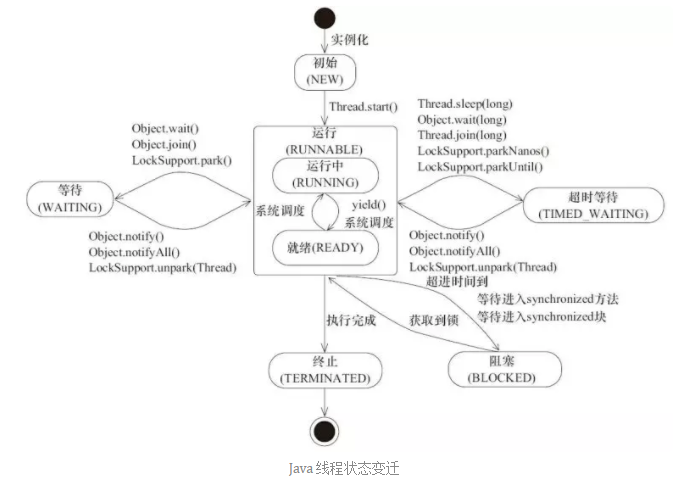
为了**保证线程中的局部变量不被别的线程访问到**，虚拟机栈和本地方法栈是线程私有的。

2、堆和方法区

堆和方法区是所有线程共享的资源，其中堆是进程中最大的一块内存，主要用于存放新创建的对象 (所有对象都在这里分配内存)，方法区主要用于存放已被加载的类信息、常量、静态变量、即时编译器编译后的代码等数据。

3、线程的生命周期

**线程的生命周期**



线程的五种状态：

初始化：实例化，并没有调用start()方法。

就绪：调用了start()方法，但是上下文还未切换到该线程。

运行: 上下文还未切换到该线程,并执行run方法中的代码。

阻塞: 线程正在运行的时候，被暂停。

终止：线程生命期结束。

4、死锁

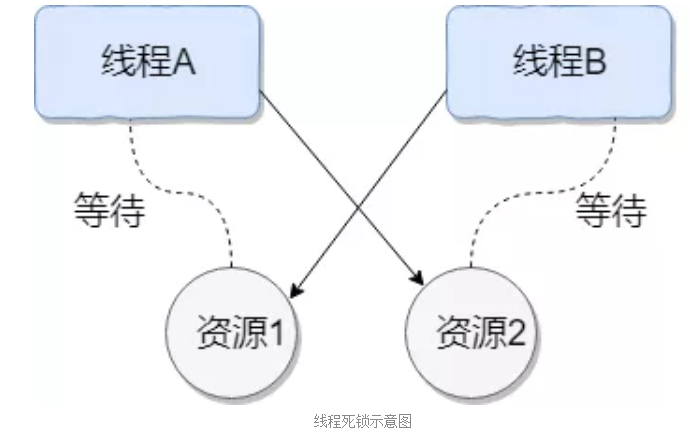
多个线程同时被阻塞，它们中的一个或者全部都在等待某个资源被释放。由于线程被无限期地阻塞，因此程序不可能正常终止。

5. zookeeper为什么至少3台服务器？

在zookeeper的选举过程中，为了保证选举过程最后能选出leader，就一定不能出现两台机器得票相同的僵局，所以一般的，要求zk集群的server数量一定要是奇数，也就是2n+1台。

6. 基本类型为什么要定义包装类

Java是一个面相对象的编程语言，基本类型并不具有对象的性质，为了让基本类型也具有对象的特征，就出现了包装类型（如我们在使用集合类型Collection时就一定要使用包装类型而非基本类型），它相当于将基本类型“包装起来”，使得它具有了对象的性质，并且为其添加了属性和方法，丰富了基本类型的操作。



7、说说 sleep() 方法和 wait() 方法区别和共同点?

（1）两者最主要的区别在于：sleep 方法没有释放锁，而 wait 方法释放了锁 。

（2）两者都可以暂停线程的执行。

（3）Wait 通常被用于线程间交互/通信，sleep 通常被用于暂停执行。

（4）wait() 方法被调用后，线程不会自动苏醒，需要别的线程调用同一个对象上的 notify() 或者 notifyAll() 方法。sleep() 方法执行完成后，线程会自动苏醒。或者可以使用 wait(long timeout)超时后线程会自动苏醒。

8、为什么我们调用 start() 方法时会执行 run() 方法，为什么我们不能直接调用 run() 方法？

总结：调用 start 方法方可启动线程并使线程进入就绪状态，而 run 方法只是 thread 的一个普通方法调用，还是在主线程里执行。

9.(面试有信网络技术有限公司的问题) 一个进程(process)最多可以开辟多少个线程(thread)？

默认情况下，一个线程的栈要预留1M的内存空间，而一个进程中可用的内存空间只有2G，所以理论上一个进程中最多可以开2048个线程。但是内存当然不可能完全拿来作线程的栈，所以实际数目要比这个值要小。

10. RESTful规则

RESTful是一种架构的规范与约束、原则，符合这种规范的架构就是RESTful架构。

11. @requestBody @RequestParam @PathValiable的理解与区别

@RequestMapping注解中用{ }来表明它的变量部分：

@PathValiable 路径变量

@RequestParam 查询字符串

@RequestBody主要用来接收前端传递给后端的json字符串中的数据的(请求体中的数据的)；一般用于POST请求。

12. 什么情况下sql查询不会走索引？

（1）模糊查询 %like

（2）索引列参与计算,使用了函数

（3）非最左前缀顺序

（4）where对null判断

（5）where不等于

（6）or操作有至少一个字段没有索引

（7）需要回表的查询结果集过大（超过配置的范围）

13.SQL的执行顺序

(1)from

　　(2) join

　　(3) on

　　(4) where

　　(5)group by

　　(6) avg,sum.... （组函数）

　　(7)having

　　(8) select

　　(9) distinct

(10) order by

14.Redis和memcached的区别

存储方式上：memcache会把数据全部存在内存之中，断电后会挂掉，数据不能超过内存大小。redis有部分数据存在硬盘上，这样能保证数据的持久性。

数据支持类型上：memcache对数据类型的支持简单，只支持简单的key-value，，而redis支持五种数据类型。

用底层模型不同：它们之间底层实现方式以及与客户端之间通信的应用协议不一样。redis直接自己构建了VM机制，因为一般的系统调用系统函数的话，会浪费一定的时间去移动和请求。

value的大小：redis可以达到1GB，而memcache只有1MB。