

浅析Java多线程技术中所使用方法Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/582/2021_2022__E6_B5_85_E6_9E_90Java_c104_582218.htm 一、run()和start() 这两个方法应该都比较熟悉，把需要并行处理的代码放在run()方法中

，start()方法启动线程将自动调用run()方法，这是由Java的内存机制规定的。并且run()方法必须是public访问权限，返回值类型为void. 二、关键字Synchronized 这个关键字用于保护共享数据，当然前提是要分清哪些数据是共享数据。每个对象都有一个锁标志，当一个线程访问该对象时，被Synchronized修饰的数据将被“上锁”，阻止其他线程访问。当前线程访问完这部分数据后释放锁标志，其他线程就可以访问了。

```
public ThreadTest implements Runnable { public synchronized void run(){ for(int i=0;i<10;i ) { System.out.println(" " i). } } public static void main(String[] args) { Runnable r1 = new ThreadTest(). Runnable r2 = new ThreadTest(). Thread t1 = new Thread(r1). Thread t2 = new Thread(r2). t1.start(). t2.start(). } }
```

以上这段程序中的i变量并不是共享数据，也就是这里的Synchronized关键字并未起作用。因为t1，t2两个线程是两个对象(r1，r2)的线程。不同的对象其数据是不同的，所以r1和r2两个对象的i变量是并不是共享数据。当把代码改成如下：Synchronized关键字才会起作用 Runnable r = new ThreadTest(). Thread t1 = new Thread(r). Thread t2 = new Thread(r). t1.start(). t2.start(). 三

、sleep() 使当前线程(即调用该方法的线程)暂停执行一段时间，让其他线程有机会继续执行，但它并不释放对象锁。也就是如果有Synchronized同步块，其他线程仍然不同访问共享数

据。注意该方法要捕获异常 比如有两个线程同时执行(没有Synchronized)，一个线程优先级为MAX_PRIORITY，另一个为MIN_PRIORITY，如果没有Sleep()方法，只有高优先级的线程执行完成后，低优先级的线程才能执行.但当高优先级的线程sleep(5000)后，低优先级就有机会执行了。总之，sleep()可以使低优先级的线程得到执行的机会，当然也可以让同优先级、高优先级的线程有执行的机会。四、join() join()方法使调用该方法的线程在此之前执行完毕，也就是等待调用该方法的线程执行完毕后再往下继续执行。注意该方法也要捕获异常。五、yield() 它与sleep()类似，只是不能由用户指定暂停多长时间，并且yield()方法只能让同优先级的线程有执行的机会。六、wait()和notify()、notifyAll() 这三个方法用于协调多个线程对共享数据的存取，所以必须在Synchronized语句块内使用这三个方法。前面说过Synchronized 这个关键字用于保护共享数据，阻止其他线程对共享数据的存取。但是这样程序的流程就很不灵活了，如何才能在当前线程还没退出Synchronized数据块时让其他线程也有机会访问共享数据呢？此时就用这三个方法来灵活控制。wait()方法使当前线程暂停执行并释放对象锁标志，让其他线程可以进入Synchronized数据块，当前线程被放入对象等待池中。当调用notify()方法后，将从对象的等待池中移走一个任意的线程并放到锁标志等待池中，只有锁标志等待池中的线程能够获取锁标志；如果锁标志等待池中没有任何线程，则notify()不起作用。notifyAll()则从对象等待池中移走所有等待那个对象的线程并放到锁标志等待池中。更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详

细资料 100Test 下载频道开通，各类考试题目直接下载。详细
请访问 www.100test.com