### 线程

#### 1.进程与线程

##### 1.1基本概念

* 程序：用某种计算机语言编写的可执行代码。
* 进程：进程就是**应用程序在内存中分配的空间，也就是正在运行的程序**。

##### 1.2进程和线程的区别

* 进程占有一定的内存空间，进程间内存隔离，数据共享比较复杂，数据同步简单，一个进程出现问题不影响其他进程，可靠性高。
* 线程共享所属进程的内存资源，数据共享简单，数据同步复杂，线程间耦合度高，一个线程崩溃可能影响整个进程的运行，可靠性低。

**进程是操作系统进行资源分配的基本单位，而线程是操作系统进行调度的基本单位。**

#### 2.线程实现

* 继承Tread类
* 例子：MyThreadOne
* 实现Runnable接口
* 例子：MyThreadTwo
* 实现Callable接口
* 例子：MyThreadThree

#### 3.线程状态

* NEW：创建之后尚未启动的线程。
* RUNNABLE：在Java虚拟机中执行的线程处于此状态。
* BLOCKED：被阻塞等待监视器锁定的线程处于此状态。
* WAITING：正在等待另一个线程执行特定动作的线程处于此状态。
* TIME\_WAITING：正在等待另一个线程执行动作达到指定等待时间的线程处于此状态。
* TERMINATED：已推出的线程处于此状态。

##### 3.1观测线程状态和线程状态转换

例子：TestState.test1();

例子：TestState.test2();

例子：TestState.test3();

##### 3.2线程休眠

线程休眠会立马交出CPU，让CPU去执行其他的任务。sleep方法不会释放锁。

例子：TestSleep

##### 3.3线程礼让

调用yield方法会让当前线程交出CPU权限，让CPU去执行其他的线程。它跟sleep方法类似，同样不会释放锁。

不会立即交出CPU权限，有可能会礼让失败

例子：TestYield

##### 3.4线程强制执行

t.join()方法只会使主线程(或者说调用t.join()的线程)进入等待池并等待t线程执行完毕后才会被唤醒。

例子：TestJoin

#### 4.线程同步

同步方法、同步代码块、锁

例子：SyncThreadTest

#### 6.线程通信

* 同步：例子：SyncThreadTest
* while轮询：例子：ThreadCommunicationTest.test1()
* wait/notify机制 ：例子：ThreadCommunicationTest.test2()
* 管道通信：例子：PipeStreamTest