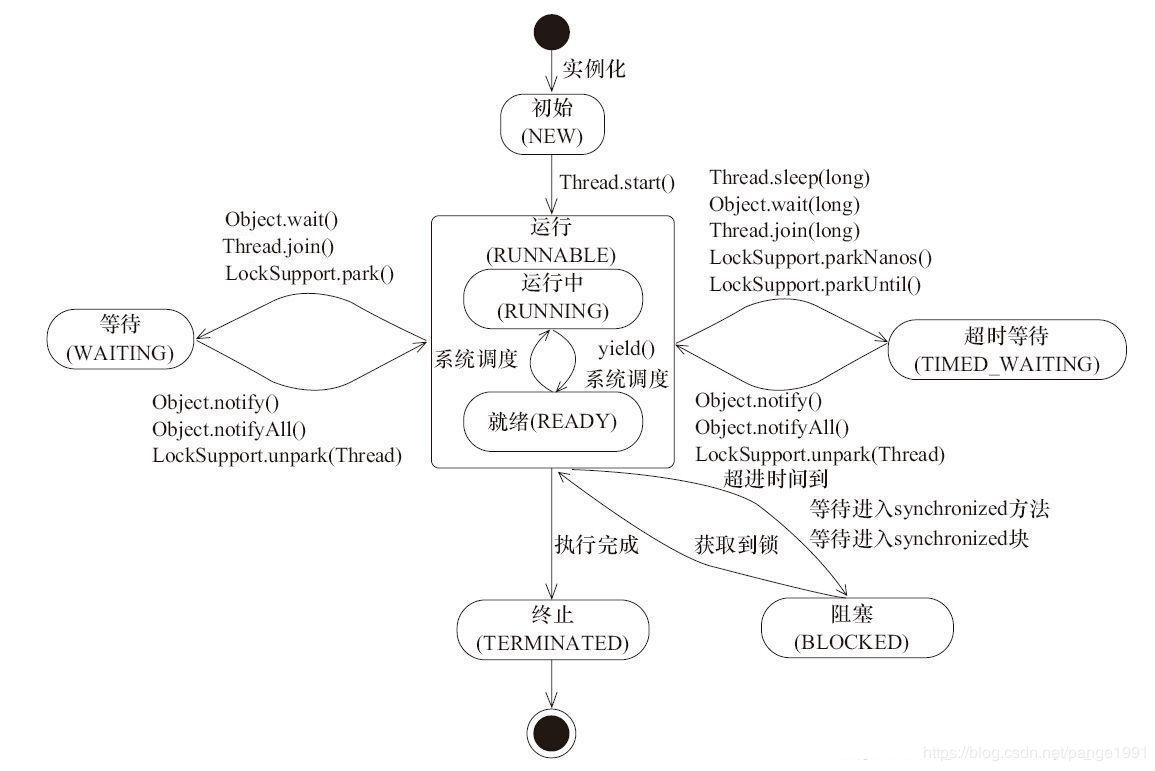
## 线程的状态



## 状态解释

初始(NEW)：新创建一个线程，调用new Thread()， 但没有调用start()方法

运行（RUNNABLE）: java 线程中的运行中（RUNNING）和(Ready)统称为运行状态

Ready: 就是主线程调用start()方法，但是线程位于可运行的线程池中，等待被系统调度，获得cpu执行权限，

Running: 系统调度，获取到cpu的执行权，

1. 阻塞(BLOCKED): 线程阻塞于锁
2. 等待（WATING）:调用wait().join(),LockSupport.park()，需要其他线程中断或是通知
3. 超时等待（TIMED\_WATING）:调用wait(long time), join(long time), LockSupport.park(long time)， 不同于WATIING ,可以在指定时间内返回
4. 终止(TERMINATED):表示该线程执行完毕

## 状态转换

### ****1. 初始状态****

实现Runnable接口和继承Thread可以得到一个线程类，new一个实例出来，线程就进入了初始状态

### ****2.1. 就绪状态****

就是有资格运行，但是系统还没有挑选到你，start()后

线程调用sleep(), 其他线程join(),等待用户输入完毕，拿到对象锁，进入就绪状态

当前线程的时间片用完了，但是还没有执行完毕，进入就绪状态

线程池里线程拿到对象锁后，进入就绪状态

### ****2.2. 运行中状态****

线程调度程序从可运行池中选择一个线程作为当前线程时线程所处的状态。这也是线程进入运行状态的唯一一种方式。

### ****3. 阻塞状态****

阻塞状态是线程阻塞在进入synchronized关键字修饰的方法或代码块(获取锁)时的状态。

### **4. 等待**

处于这种状态的线程不会被分配CPU执行时间，它们要等待被显式地唤醒，否则会处于无限期等待的状态。

6. 终止状态

当线程的run()方法完成时，或者主线程的main()方法完成时，我们就认为它终止了。这个线程对象也许是活的，但是，它已经不是一个单独执行的线程。线程一旦终止了，就不能复生。

在一个终止的线程上调用start()方法，会抛出java.lang.IllegalThreadStateException异常。

线程的sleep(millons), 进入超时等待状态，释放cpu, 但是不释放锁，millions后进入就绪状态，

Yield(), 释放cpu, 但是不释放锁, ， 进入就绪状体

hread.join()/thread.join(long millis)，当前线程里调用其它线程t的join方法，当前线程进入WAITING/TIMED\_WAITING状态，当前线程不会释放已经持有的对象锁。线程t执行完毕或者millis时间到，当前线程进入就绪状态。

obj.wait()，当前线程调用对象的wait()方法，当前线程释放对象锁，进入等待队列。依靠notify()/notifyAll()唤醒或者wait(long timeout) timeout时间到自动唤醒。

obj.notify()唤醒在此对象监视器上等待的单个线程，选择是任意性的。notifyAll()唤醒在此对象监视器上等待的所有线程。

LockSupport.park()/LockSupport.parkNanos(long nanos),LockSupport.parkUntil(long deadlines), 当前线程进入WAITING/TIMED\_WAITING状态。对比wait方法,不需要获得锁就可以让线程进入WAITING/TIMED\_WAITING状态，需要通过LockSupport.unpark(Thread thread)唤醒。