注：From thinking in Java。

注：A thread can be in any of four states：New，Runnable，Dead，Blocked。New是创建线程，调用start后才处于Runnable状态，run（）运行完后变Dead状态。Blocked则有五种情况，如sleep、wait等，但CPU都会跳过而不执行

注：重点标色

blocking

A thread can be in any of four states:

1. New: the thread object has been created but it hasn’t been started yet so it cannot run.

2. Runnable: This means that a thread can be run, when the time-slicing mechanism has CPU cycles available for the thread. Thus the thread may or may not be running, but there’s nothing to prevent it from being run if the scheduler can arrange it: that is, it’s not dead or blocked.

3. Dead: the normal way for a thread to die is by returning from it’s run( ) method. You can also call stop( ), but this throws an exception that’s a subclass of Error (which means you normally don’t catch it). Remember that throwing an exception should be a special event and not part of normal program execution; thus the use of stop( ) is discouraged. There’s also a destroy( ) method that you should never call if you can avoid it, since it’s drastic（急剧的） and doesn’t release object locks.

4. Blocked: the thread could be run but there’s something that prevents it. While a thread is in the blocked state the scheduler will simply skip over it and not give it any CPU time. Until a thread re-enters the runnable state it won’t perform any operations.

becoming blocked

The blocked state is the most interesting and is worth further examination. A thread can become blocked for five reasons:

1. You’ve put the thread to sleep by calling sleep(milliseconds) in which case it will not be run for the specified time.

2. You’ve suspended the execution of the thread with suspend( ). It will not become runnable again until the thread gets the resume( ) message.

3. You’ve suspended the execution of the thread with wait( ). It will not become runnable again until the thread gets the notify( ) or notifyAll( ) message (yes, this looks just like number 2, but there’s a distinct difference that will be revealed).

4. The thread is waiting for some IO to complete.

5. The thread is trying to call a synchronized method on another object and that object’s lock is not available. You can also call yield( )（按：yield：让步；放弃） (a method of the Thread class) to voluntarily（自愿地） give up the CPU so other threads may run. However, this is no different than if the scheduler decides that your thread has had enough time and jumps to another thread. That is, nothing prevents the scheduler from re-starting your thread. When a thread is blocked, there’s some reason that it cannot continue running. The following example shows all five ways of becoming blocked. It all exists in a single file called

按：代码去找Thinking in java书籍查出名字后，可到Java收集资料项目（JavaCodeCollection）找出案例来看。