Java开发中线程是经常用到的技术，那么让我们来回顾一下一些线程中经常考虑的问题吧

1、线程分为用户线程和守护线程，这两种线程有什么区别

2、线程的运行状态有哪些

3、线程的上下文切换是什么，切换过程中发生了什么

1、什么是线程

线程，一个执行实体，正在执行的程序，担当分配系统资源（CPU、内存）的实体。

对于我们来说一个完整的线程包括，需要运行的逻辑和需要运行需要的资源。

2、线程的状态

线程有哪些状态呢，在Thread的代码中的State枚举已经很清楚了

public enum State { *NEW*, *RUNNABLE*, *BLOCKED*,  
 *WAITING*,  
 *TIMED\_WAITING*, *TERMINATED*;  
}

线程的状态实际是7种，新建、就绪、运行、阻塞、等待、超时等待、终止，不过一般将就绪和运行都算作运行中

新建：新建的线程对象，还未调用start方法

就绪：已经具备了运行条件，等待CPU服务，这个状态下，线程在就绪队列中等待

运行：线程具备运行条件，并获的了CPU资源

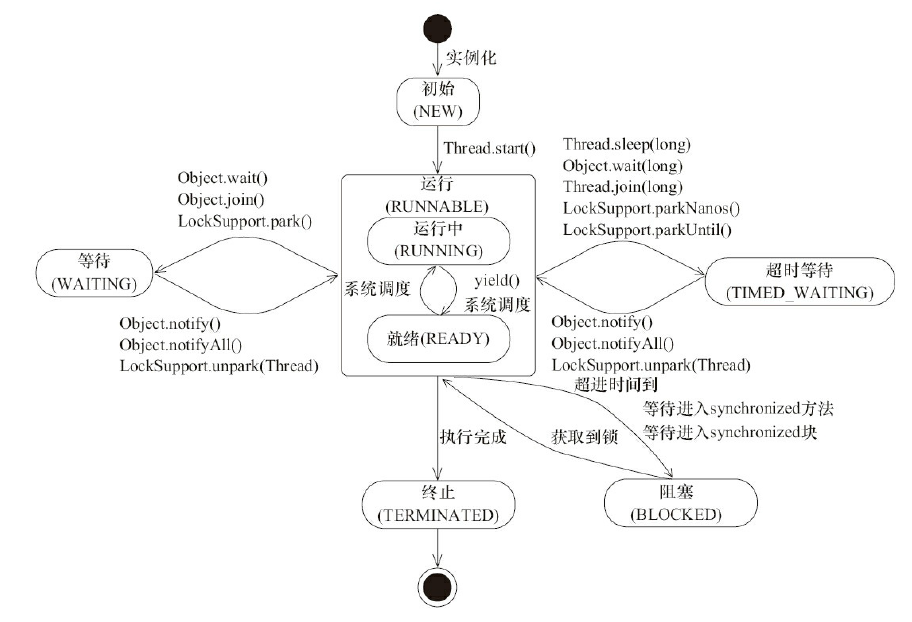
阻塞：线程被挂起，一般是加锁获取同步状态被阻塞，放到阻塞队列中，阻塞状态消除后，会进入就绪状态

等待：等待状态表示当前线程需要等待其他线程做出一些特定动作（等待通知机制）

超时等待：在等待的基础上，加上一定时间后返回就绪状态

终止：线程已经运行完毕

那么这些状态是如何切换的呢，借用java并发编程艺术中的一张图



start方法调用后进入运行状态，wait、sleep等方法进入等待状态或，yeild方法进入就绪状态，lock和synchronized进入阻塞状态

3、线程的分类和使用

线程可以分为守护线程和用户线程两种，那么什么是用户线程，什么是守护线程呢？

1、用户线程

用户线程就是我们平时使用的用来处理逻辑的线程。

特点：

2、守护线程

守护线程，是服务线程，程序运行时在后台提供的一种通用服务的线程。最常见的就是jvm的垃圾回收线程。

1、什么是线程

2、线程的状态，每个状态的线程在做什么和状态的切换

3、线程分类和作用

4、线程太多为什么会消耗资源

5、线程的创建方法

6、线程停止的方式和比较

5、sleep和wait的比较