

并行&伪并行

在许许多多道程序系统中，CPU 会在进程间快速切换，使每个程序运行几十或者几百毫秒。然而，严格意义来说，在某一个瞬间，CPU 只能运行一个进程，然而我们如果把时间定位为 1 秒内的话，它可能运行多个进程。这样就会让我们产生并行的错觉。有时候人们说的 伪并行(pseudoparallelism) 就是这种情况，以此来区分多处理器系统(该系统由两个或多个 CPU 来共享同一个物理内存)

进程&线程执行方式

单核CPU在某一时间点只能执行一个线程，多核执行多个，线程数=CPU核数

某一时间段内，单核CPU可执行多个进程，谁拿到时间片谁就执行，其他进程阻塞

进程创建

- (1) 系统初始化 init：操作系统启动时创建一些看不见的进程
- (2) 运行的程序执行创建进程的调用：启动某程序后续创建一些相关进程
- (3) 用户请求：点点点
- (4) 初始化批处理：用户写了批处理的文件并执行，系统执行完某一个命令，满足下一条命令执行的条件，自动创建进程执行下一条命令

进程终止

- (1) 正常退出

(2) 错误退出：进程报错并退出，一般会弹出对话框让用户选择重试还是退出

(3) 严重错误：执行非法指令，比如除数是0，有点像开发过程中的异常

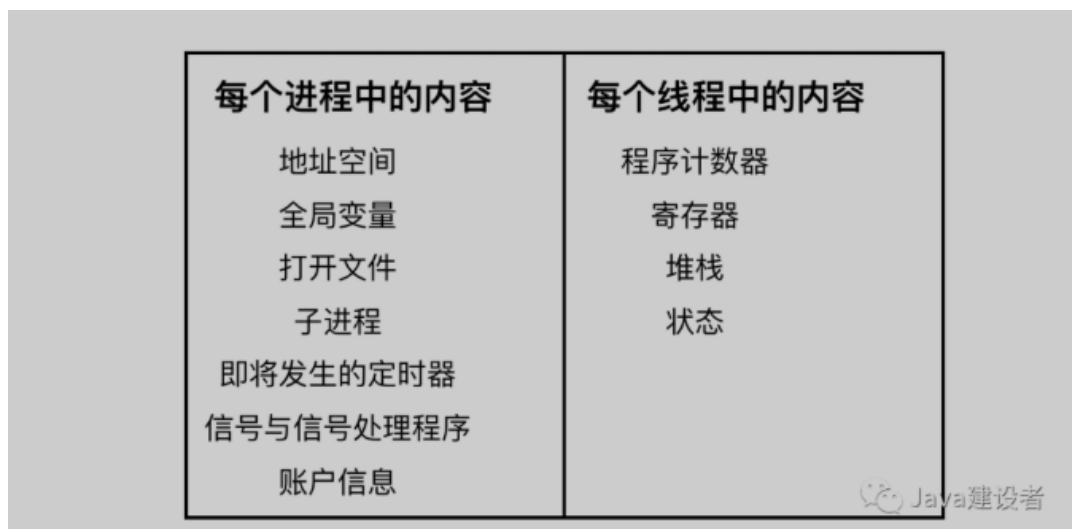
(4) 被其他进程杀死：kill命令

进程状态

阻塞-就绪-执行

就绪和执行很像，就绪态在获得时间片之后可立即执行，阻塞态不可以，即使有时间片也不能执行

进程与线程关系

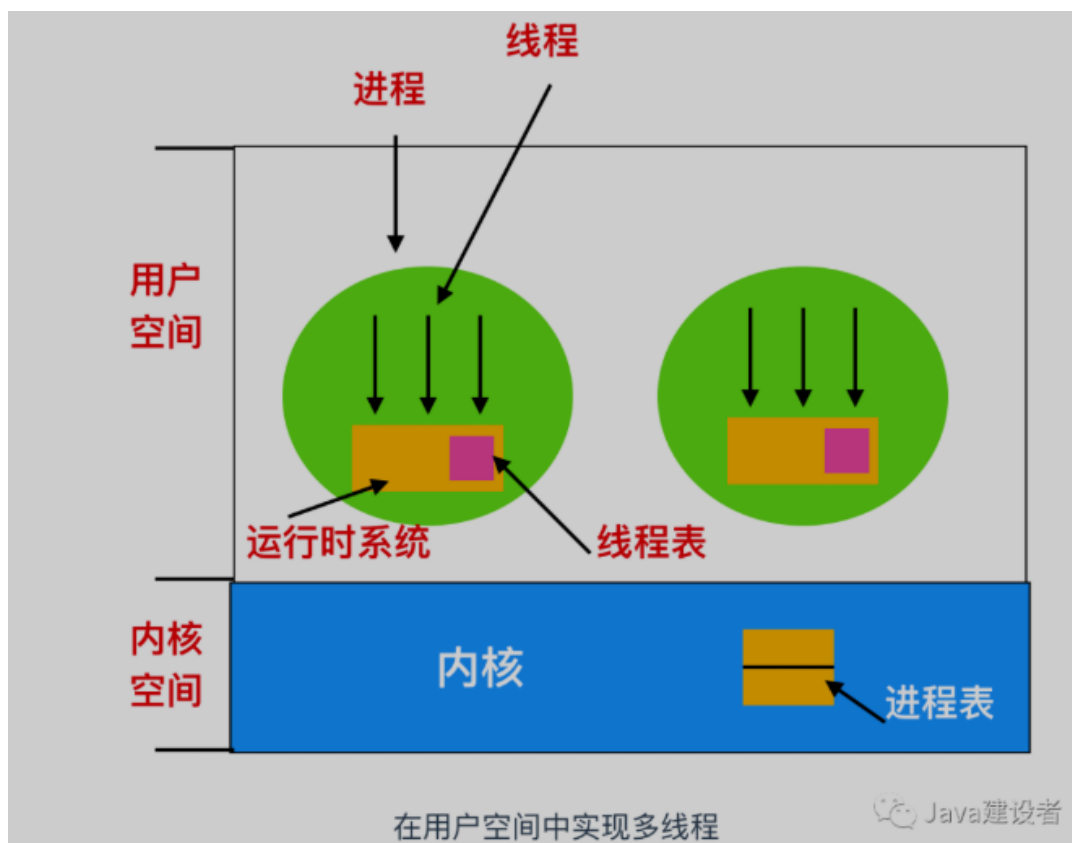


线程状态：运行-阻塞-就绪-终止

线程实现方式

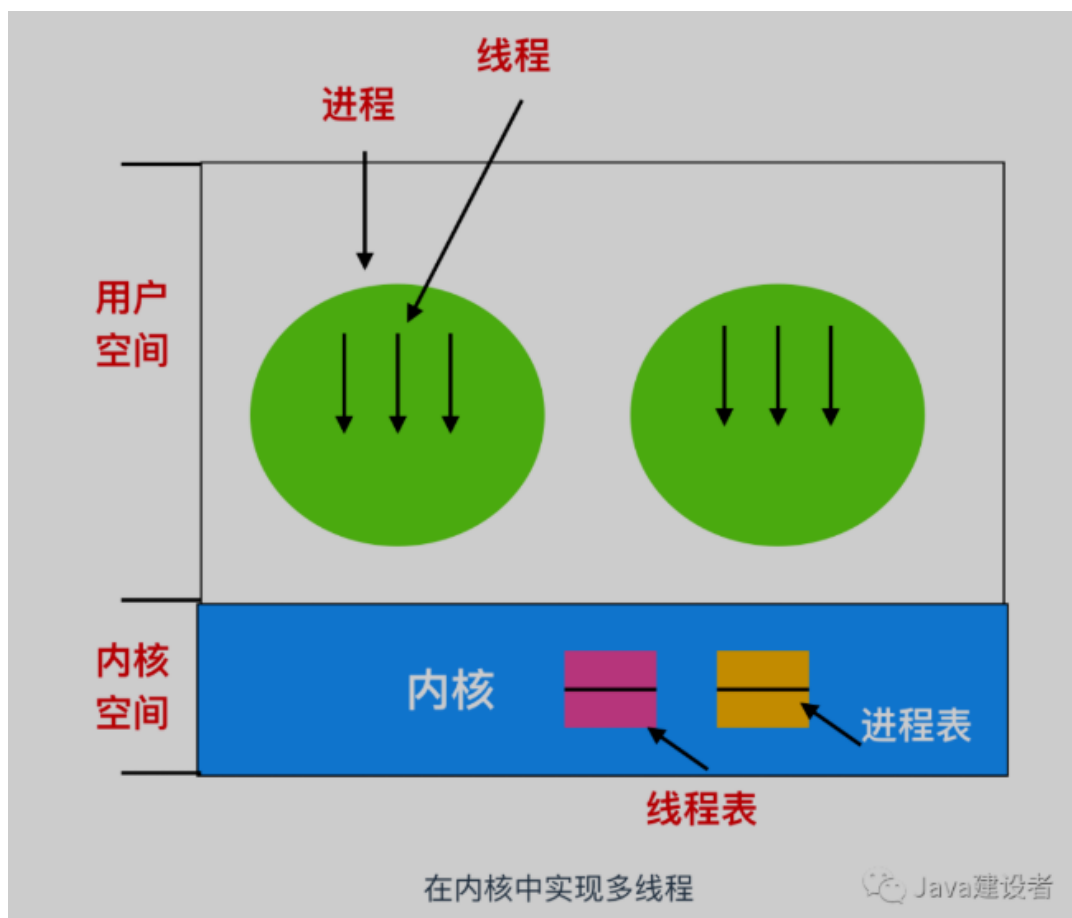
(1) 用户态：每个线程有线程表，记录信息

优点，效率高，缺点在线程内部一个线程运行其他都不能运行，没有时钟中断，线程表占用空间，线程多了出现问题

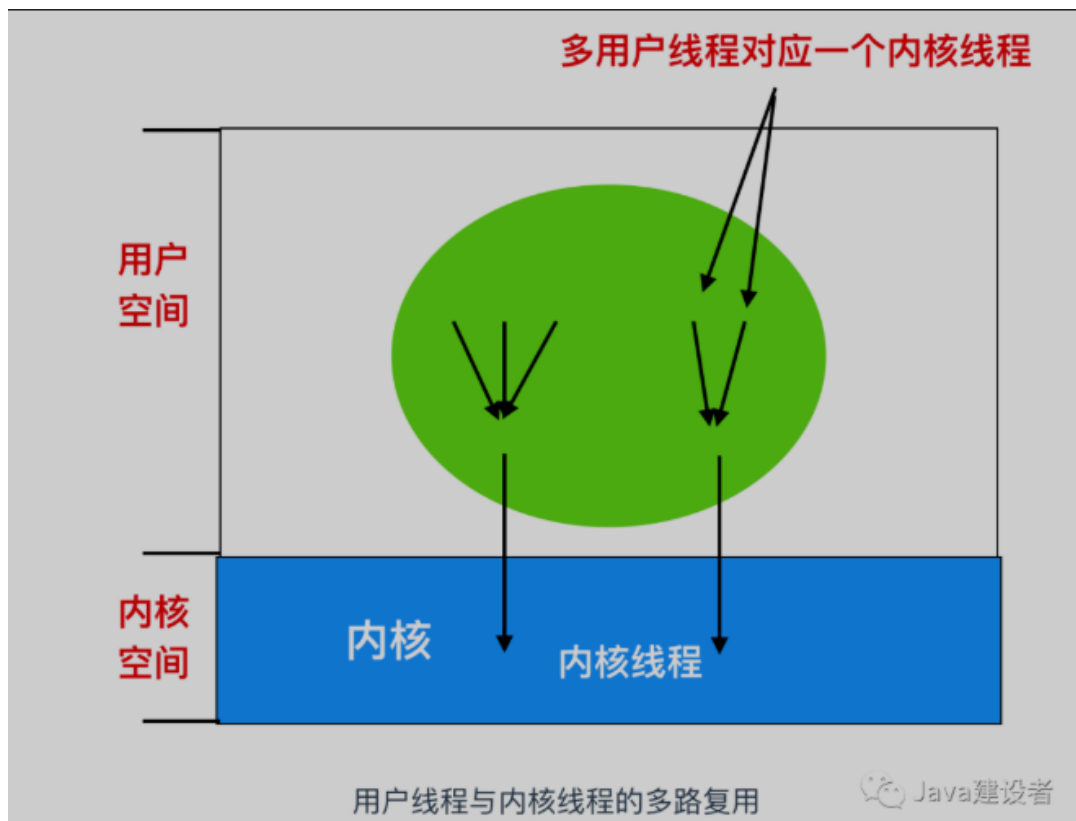


(2) 内核态：当某个线程希望创建一个新线程或撤销一个已有线程时，它会进行一个系统调用，这个系统调用通过对线程表的更新来完成线程创建或销毁工作。

缺点，创建销毁进程开销大



(3) 混合态



进程间通信

需考虑如何传递，互相干扰，顺序问题