# 基本概念

## 程序 - 进程 - 线程

### 程序和进程

**程序(program)：**

指令的集合；

静态资源。如：没开的QQ

**进程(process)：（执行中的程序）**

程序并不能单独执行，只有将程序加载到内存中，系统为他分配资源后才能够执行，这种执行的程序称之为进程；

动态过程（静态资源被编译后）：有它自身的产生、存在和消亡的过程。如：运行中的QQ

### 进程缺陷：

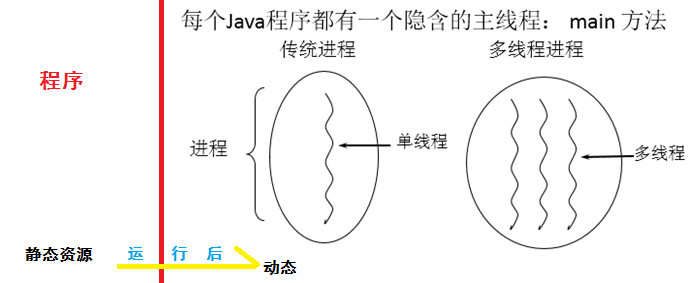
(1).进程只能在同一时间干一件事情，如果想同时干两件事或多件事情，进程就无能为力了。   
(2).进程在执行的过程中如果由于某种原因阻塞了，例如等待输入，整个进程就会挂起，其他与输入无关的工作也必须等待输入结束后才能顺序执行。

### 解决方法： 引入线程

线程(thread)是进程的一个实体,也是CPU调度和分派的基本单位,它是比进程更小的能独立运行的基本单位，有时又被称为轻权进程或轻量级进程

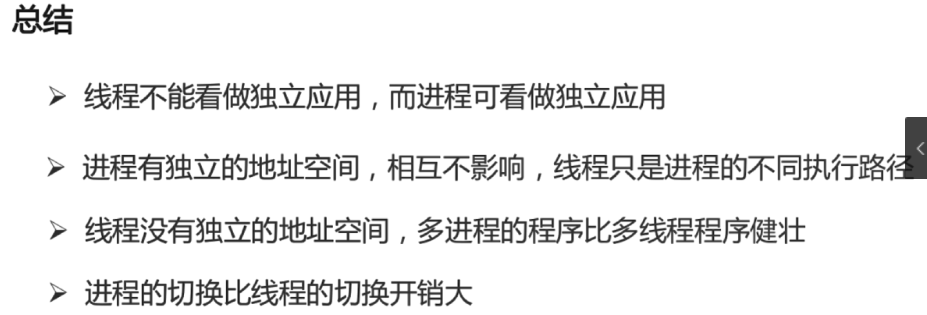
## 关系



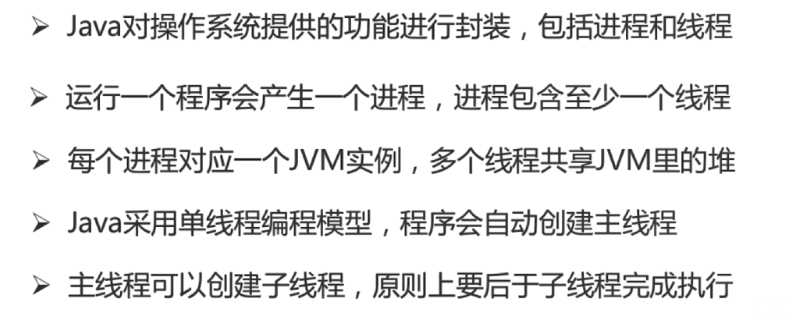


## 进程和线程

### 区别



### 关系



# 多线程

## 何时需要多线程

程序需要同时执行两个或多个任务。

程序需要实现一些需要等待的任务时，如用户输入、文件读写操作、网络操作、搜索等。

需要一些后台运行的程序时。

## 并行和并发

并行就是两个任务同时运行，就是甲任务进行的同时，乙任务也在进行。(需要多核CPU)

比如我跟两个网友聊天，左手操作一个电脑跟甲聊，同时右手用另一台电脑跟乙聊天，这就叫并行

并发是指两个任务都请求运行，而处理器只能按受一个任务，就把这两个任务安排轮流进行，由于时间间隔较短，使人感觉两个任务都在运行。

如果用一台电脑我先给甲发个消息，然后立刻再给乙发消息，然后再跟甲聊，再跟乙聊。这就叫并发。

## 优缺点

### 多线程的好处：

1. 解决了一个进程里面可以同时运行多个任务（执行路径）。改善程序结构。将既长又复杂的进程分为多个线程，独立运行，利于理解和修改
2. 提供资源（计算机系统CPU的利用率）的利用率，而不是提供效率。
3. 提高应用程序的响应。对图形化界面更有意义，可增强用户体验。

### 多线程的弊端:

1. 降低了一个进程里面的线程的执行频率。
2. 对线程进行管理要求额外的 CPU开销。线程的使用会给系统带来上下文切换的额外负担。
3. 公有变量的同时读或写。当多个线程需要对公有变量进行写操作时,后一个线程往往会修改掉前一个线程存放的数据，发生线程安全问题。
4. 线程的死锁。即较长时间的等待或资源竞争以及死锁等多线程症状。

## 注意

### Java程序运行原理

Java命令会启动java虚拟机，启动JVM，等于启动了一个应用程序，也就是启动了一个进程。该进程会自动启动一个 “主线程” ，然后主线程去调用某个类的 main 方法。

### JVM的启动是多线程的吗

JVM启动至少启动了垃圾回收线程和主线程，所以是多线程的。