进程、线程与程序的区别

程序、进程与线程

# 进程、线程与程序

## 进程(Process)

**进程**：就是正在进行的程序，主要是用来开辟一片内存空间；是程序的一次动态执行，它对应了从**代码加载、执行至执行完毕**的一个完整过程，这个进程也是进程本身从产生、发展至消亡的过程，是一个动态概念。

## 线程(Thread)---又可以称为“轻量级进程”

**线程**：就是进程中一个负责程序执行的控制单元或者**叫做一个执行路径**。是比进程更小的执行单位。一个进程在其执行过程中，可以产生过个线程，形成多条执行路径，每条路径，即每个线程也有它自身的产生、存在和消亡的过程，**也是一个动态的概念**。

## 程序（静态的）

**程序**：**一段静态的代码**、**指令集**，是应用软件执行的蓝本。



# 进程与线程的区别

## 根本区别：

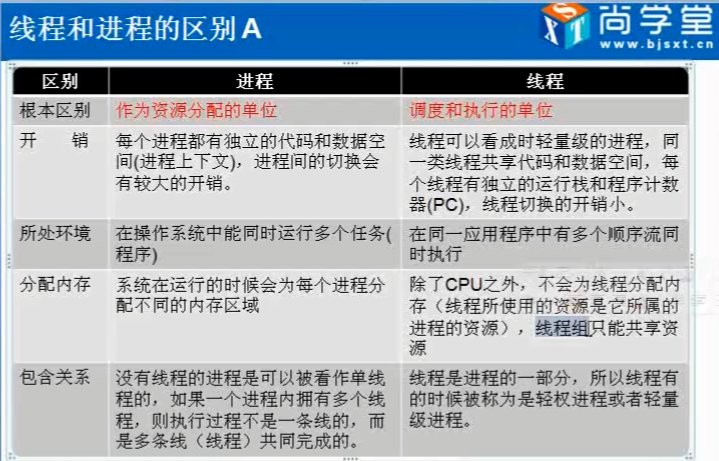
进程作为**资源分配的单位**；线程作为**调度和执行的单元**。

## 开销

## 所处环境

## 分配内存

## 包含关系



# 线程的特点：

### 一个进程中至少有一个线程。

### 一个进程中允许有多个执行路径，即多线程。

### 开启多个线程是为了同时运行多部分代码；

### 任务： 每个线程都具有自己运行的内容，这个内容就可以称为线程要执行的任务。

### 多线程的好处：实现了多部分代码同时运行的目的；

### 多线程的弊端：多线程过多，效率降低。因为程序的执行都是在CPU中做着快速的切换完成的，且这个切换是随机的。

# Java虚拟机对多线程的支持

## JVM启动时就已经启动了至少两个线程：

### 主线程： 执行main方法的线程，该线程的任务代码都定义在main方法里；

### 负责垃圾回收的线程。

## 创建新线程的目的

目的：为了**开启一条新的执行路径**，能够实现此代码和其他代码的**同时运行**，而运行此代码就是这个执行路径的任务。

## Java虚拟机支持多线程，且多线程间可以相互通信。

### Java应用程序总是从主类的main（）方法开始执行。

### 执行main（）方法就会启动一个线程，称为主线程。创建其他线程，必须在main（）方法中创建。

### 等到所有的线程都结束完毕，JVM才会结束该应用程序。

### 各个线程之间是相互独立的；

### 共享数据需要考虑并发问题：面试时经常会问到此类问题。