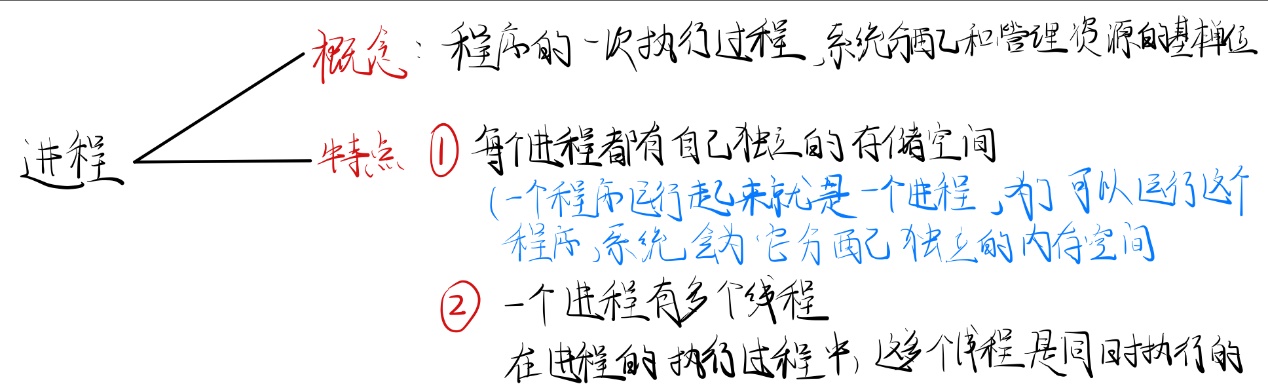
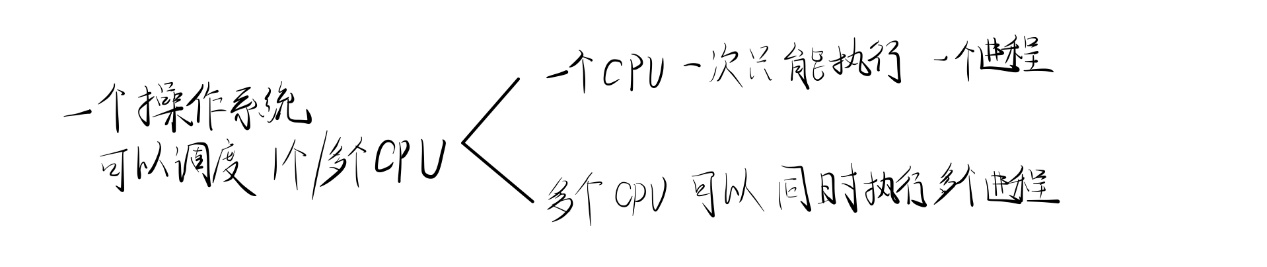
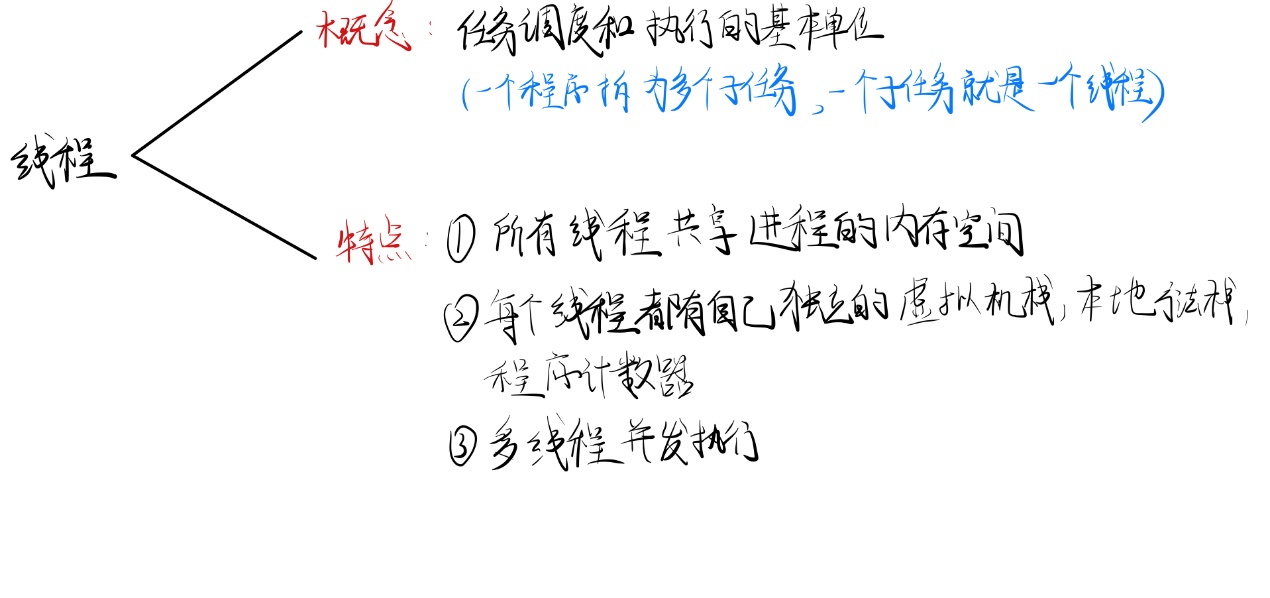
**一，进程**



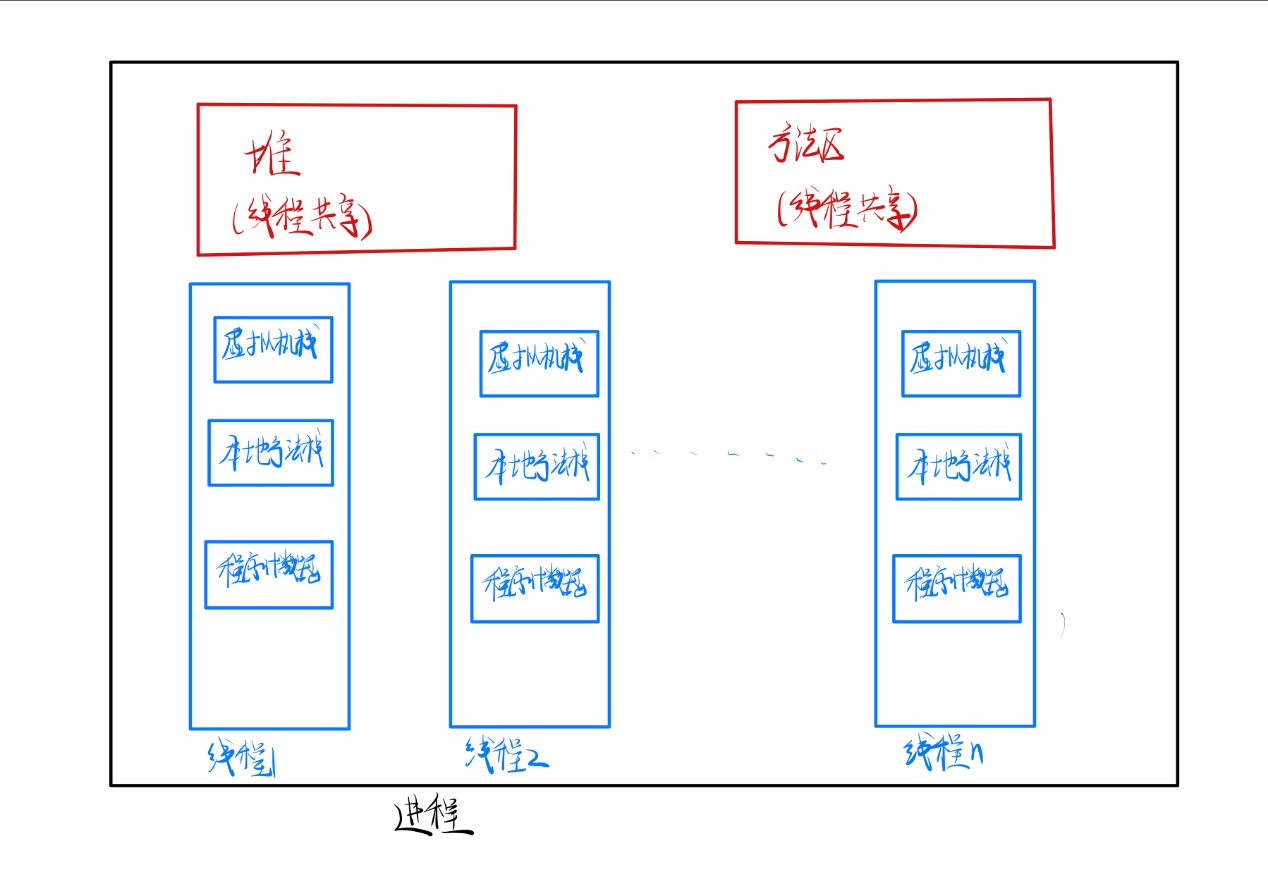
**二，进程与CPU**



**三，线程**



**四，结构图**



**堆**  ——**存放实例对象**

**方法区**——**存放已被加载的类信息、常量、静态变量、即时编译器编译后的代码等数据**

**五，进程与线程的区别**

**①每个进程都有自己独立的内存空间——各个进程进行切换时，花销大**

**每个进程都有自己独立的程序计数器，本地方法栈，虚拟机栈，但是所有线程共享堆和方法区——各个线程之间进行切换时，花销小**

②**一个进程崩溃后，在保护模式下不会对其他进程产生影响，但是一个线程崩溃整个进程都死掉。所以多进程要比多线程健壮**

**六，单线程，多线程，多进程——多线程的实质**

**1，多线程——操作系统同时执行多个程序**

**2，单线程与多线程的区别**

**单线程——依次执行各个线程**

**多线程——多个线程同时执行**

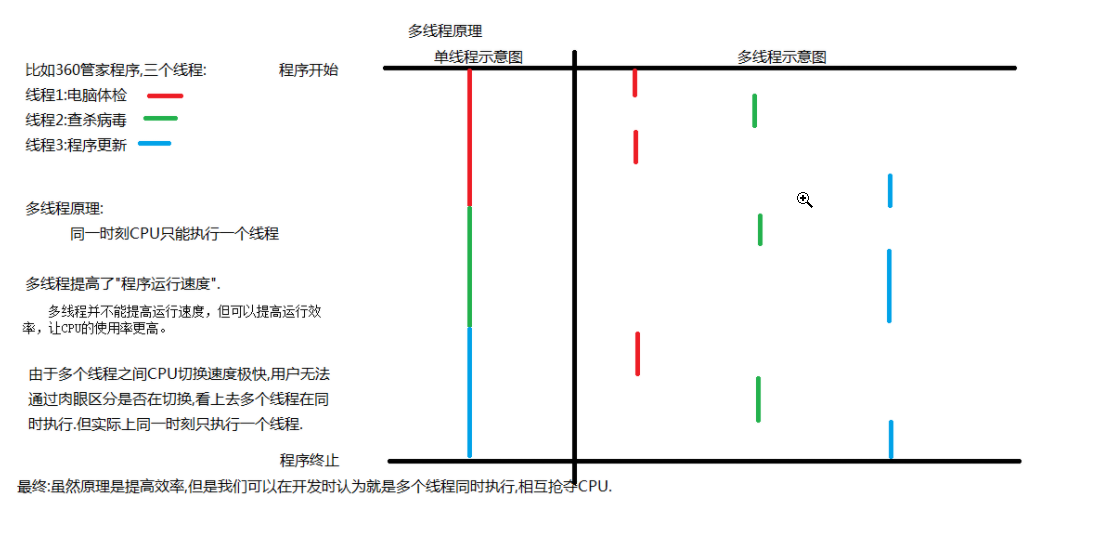
**大前提：同一时刻CPU只能执行一个线程**

**一个CPU对应一个进程**

**一个进程有许多个线程，我们说 执行进程的时候，多个线程是同时执行的**

**！But，尽管我们说进程的执行是多线程的，但是如果我们把时间缩小到非常小，我们会发现，在一个瞬间，CPU只会执行一个线程，那为什么说进程的执行是多线程的呢？**

**这是因为CPU会在不同的线程之间来回切换执行，切换速度太快了，让我们觉得这多个线程是在同时执行，所以叫多线程，如图所示**



**七，Java程序中的多线程**

**Java程序的进程里有几个线程：主线程，垃圾回收线程(后台线程)等**

**在 Java 中，当我们启动 main 函数时其实就是启动了一个 JVM 进程，而 main 函数所在的线程就是这个进程中的一个线程，也称主线程。**

**Java支持多线程，当Java程序执行main方法的时候，就是在执行一个名字叫做main的线程，可以在main方法执行时，开启多个线程A,B,C，多个线程 main,A,B,C同时执行，相互抢夺CPU**

**Thread类是java.lang包下的一个常用类,每一个Thread类的对象，就代表一个处于某种状态的线程**

**八，例子**

例子1：

一个车间——一个CPU

一个工厂，有多个车间——多CPU计算机，有多个CPU

一个程序——一个进程

一个车间一次只能运行一个程序（一个CPU一次只能运行一个进程）

一个工厂有多个车间，可以同时运行多个程序（多个CPU可以同时运行多个进程）

车间中的一个工人——一个线程（一个车间有多个工人——一个进程有多个线程）

车间内的空间是工人们共享的——进程的内存空间被所有线程共享

例子2：

开了QQ——开了一个进程，开了迅雷——开了一个进程

在QQ这个进程里，发送消息——开了一个线程，传输语音——开了一个线程

传送文件——开始一个线程

QQ同时发送消息，传送文件，就是QQ这个进程中同时执行多个线程