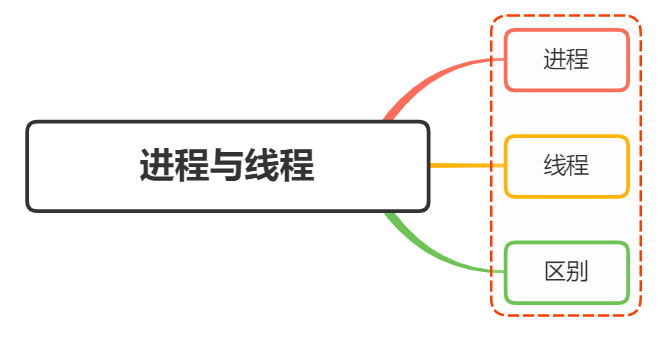
# 面试官：说说什么是进程？什么是线程？区别？



## 一、进程

操作系统中最核心的概念就是进程，进程是对正在运行中的程序的一个抽象，是系统进行资源分配和调度的基本单位

操作系统的其他所有内容都是围绕着进程展开的，负责执行这些任务的是CPU



进程是一种抽象的概念，从来没有统一的标准定义看，一般由程序、数据集合和进程控制块三部分组成：

* 程序用于描述进程要完成的功能，是控制进程执行的指令集
* 数据集合是程序在执行时所需要的数据和工作区
* 程序控制块，包含进程的描述信息和控制信息，是进程存在的唯一标志

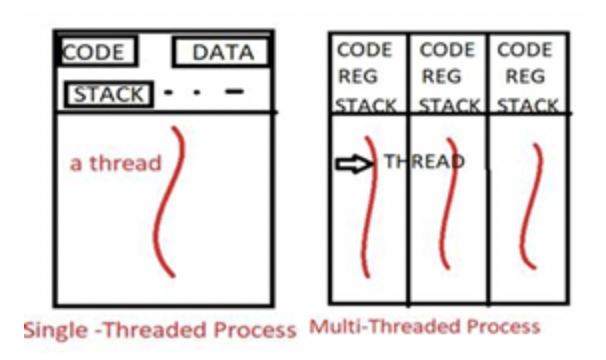
## 二、线程

**线程**（thread）是操作系统能够进行**运算调度**的最小单位，其是进程中的一个执行任务（控制单元），负责当前进程中程序的执行

一个进程至少有一个线程，一个进程可以运行多个线程，这些线程共享同一块内存，线程之间可以共享对象、资源，如果有冲突或需要协同，还可以随时沟通以解决冲突或保持同步

举个例子，假设你经营着一家物业管理公司。最初，业务量很小，事事都需要你亲力亲为。给老张家修完暖气管道，立马再去老李家换电灯泡——这叫单线程，所有的工作都得顺序执行

后来业务拓展了，你雇佣了几个工人，这样，你的物业公司就可以同时为多户人家提供服务了——这叫多线程，你是主线程



但实际上，并不是线程越多，进程的工作效率越高，这是因为在一个进程内，不管你创建了多少线程，它们总是被限定在一颗CPU内，或者多核CPU的一个核内

这意味着，多线程在宏观上是并行的，在微观上则是分时切换串行的，多线程编程无法充分发挥多核计算资源的优势

这导致使用多线程做任务并行处理时，线程数量超过一定数值后，线程越多速度反倒越慢的原因

## 三、区别

* **本质区别**：进程是操作系统资源分配的基本单位，而线程是任务调度和执行的基本单位
* **在开销方面**：每个进程都有独立的代码和数据空间（程序上下文），程序之间的切换会有较大的开销；线程可以看做轻量级的进程，同一类线程共享代码和数据空间，每个线程都有自己独立的运行栈和程序计数器（PC），线程之间切换的开销小
* **所处环境**：在操作系统中能同时运行多个进程（程序）；而在同一个进程（程序）中有多个线程同时执行（通过CPU调度，在每个时间片中只有一个线程执行）
* **内存分配方面**：系统在运行的时候会为每个进程分配不同的内存空间；而对线程而言，除了CPU外，系统不会为线程分配内存（线程所使用的资源来自其所属进程的资源），线程组之间只能共享资源
* **包含关系**：没有线程的进程可以看做是单线程的，如果一个进程内有多个线程，则执行过程不是一条线的，而是多条线（线程）共同完成的；线程是进程的一部分，所以线程也被称为轻权进程或者轻量级进程

举个例子：进程=火车，线程=车厢

* 线程在进程下行进（单纯的车厢无法运行）
* 一个进程可以包含多个线程（一辆火车可以有多个车厢）
* 不同进程间数据很难共享（一辆火车上的乘客很难换到另外一辆火车，比如站点换乘）
* 同一进程下不同线程间数据很易共享（A车厢换到B车厢很容易）
* 进程要比线程消耗更多的计算机资源（采用多列火车相比多个车厢更耗资源）
* 进程间不会相互影响，一个线程挂掉将导致整个进程挂掉（一列火车不会影响到另外一列火车，但是如果一列火车上中间的一节车厢着火了，将影响到所有车厢）

## 参考文献

* https://zhuanlan.zhihu.com/p/106283969
* https://blog.csdn.net/ThinkWAon/article/details/102021274
* https://www.zhihu.com/question/25532384