

# 进程 Vs 线程、并发 Vs 并行

作者: [DongXiaokai0819](#)

原文链接: <https://ld246.com/article/1572011109419>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 进程Vs线程、并发Vs并行

## CPU核心数跟线程数的关系？

对于一个CPU，线程数 $\geq$ 核心数

也就是说，一个核心最少可以对应一个线程，但通过超线程技术，一个CPU核心可以同时运行两个线程。

- 线程数可等同于在某一时刻，CPU并行执行任务的数量。

## 进程Vs线程

### 何为进程？

- 进程是一个具有独立功能的程序，在一个数据集上的一次动态执行的过程，是系统进行资源分配和度度的基本单位。
- 在早期面向进程设计的计算机结构中，进程是程序的基本执行实体。
- 在当代面向线程设计的计算机结构中，进程是线程的容器。
- 程序是指令、数据及其组织形式的描述，进程是程序的实体。

#### 第一点

进程是一个实体。每一个进程都有它自己的地址空间，一般情况下，包括文本区域（text region）、数据区域（data region）和堆栈（stack region）。文本区域存储处理器执行的代码；数据区域存储变化和进程执行期间使用的动态分配的内存；堆栈区域存储着活动过程调用的指令和本地变量。

#### 第二点

进程是一个“执行中的程序”。程序是一个没有生命的实体，只有处理器赋予程序生命时（操作系统运行之），它才能成为一个活动的实体，我们称其为进程。

简而言之呢，可概括为：

- 进程是程序的一次执行过程，是系统运行程序的基本单位，因此进程是动态的。
- 系统运行一个程序即是一个进程从创建，运行到消亡的过程。

### 何为线程？

- 线程是操作系统能够进行运算调度的最小单位。
- 被包含在进程之中，是进程中的实际运作单位。
- 一条线程指的是进程中一个单一顺序的控制流，一个进程中可以并发多个线程，每条线程并行执行不同的任务。

#### 重要的一点

各个线程之间共享程序的内存空间(也就是所在进程的内存空间)。

一个标准的线程由线程ID, 当前指令指针PC, 寄存器和堆栈组成。而进程由内存空间(代码, 数据, 程序空间, 打开的文件)和一个或多个线程组成。

## Java中的进程与线程

之所以要整理这么多, 是因为, 我觉得只总结几句话的话真的很难理解什么叫进程什么叫线程, 大家读读上述概念自然就明白了。

接下来我们回到JAVA中来。

```
public class MultiThread {
    public static void main(String[] args) {
        // 获取 Java 线程管理 MxBean
        ThreadMXBean threadMXBean = ManagementFactory.getThreadMXBean();
        // 不需要获取同步的 monitor 和 synchronizer 信息, 仅获取线程和线程堆栈信息
        ThreadInfo[] threadInfos = threadMXBean.dumpAllThreads(false, false);
        // 遍历线程信息, 仅打印线程 ID 和线程名称信息
        for (ThreadInfo threadInfo : threadInfos) {
            System.out.println("[ " + threadInfo.getThreadId() + " ] " + threadInfo.getThreadName())
        }
    }
}
```

以上程序输出如下, 打印了参与的所有线程名

```
[5] Attach Listener //添加事件
[4] Signal Dispatcher // 分发处理给 JVM 信号的线程
[3] Finalizer //调用对象 finalize 方法的线程
[2] Reference Handler //清除 reference 线程
[1] main //main 线程,程序入口
```

在Java中, 当我们启动main函数时, 其实就是启动了一个JVM的进程, 而main函数所在的线程以及上述的几个线程都是属于这一个进程的。

## 进程与线程的区别

1. 线程是程序执行的最小单位, 而进程是操作系统分配资源的最小单位;
2. 一个进程由一个或多个线程组成, 线程是一个进程中代码的不同执行路线
3. 进程之间相互独立, 但同一进程下的各个线程之间共享程序的内存空间(包括代码段, 数据集, 堆等及一些进程级的资源(如打开文件和信号等), 某进程内的线程在其他进程不可见;
4. 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; 线程可以做轻量级的进程, 同一个进程内的线程共享一定的代码和数据空间, 每个线程都有自己独立的运行栈程序计数器 (PC), 线程之间切换的开销小。

## 从JVM角度看Java中进程与线程的关系

### 并发Vs并行

### 并发

- 当有多个线程在操作时,如果系统只有一个CPU,则它根本不可能真正同时进行一个以上的线程,它能把CPU运行时间划分成若干个时间段,再将时间段分配给各个线程执行,在一个时间段的线程代码运行时,其它线程处于挂起状。这种方式我们称之为并发(Concurrent)。

## 并行

- 当系统有一个以上CPU时,则线程的操作有可能非并发。当一个CPU执行一个线程时,另一个CPU可执行另一个线程,两个线程互不抢占CPU资源,可以同时进行,这种方式我们称之为并行(Parallel)。

## 二者区别

- 并发是指两个或多个事件在同一时间间隔内发生,并行是指两个或者多个事件在同一时刻发生。

## 以上部分内容参考自

- 百度百科
- <https://snailclimb.top/JavaGuide/#/?id=%e5%b9%b6%e5%8f%91>
- <https://blog.csdn.net/kuangsonghan/article/details/80674777>

站在巨人的肩膀上~