## 进程

进程：正在进行中的程序（直译）

线程：就是进程中一个负责程序执行单元的控制单元（执行路径）

一个进程中可以有多执行路径，称之为多线程

一个进程中至少有一个线程

开启多个线程是为了同时运行多部分代码

每一个线程都有自己运行的内容，这个内容可以称为线程要执行的任务。

## 多线程的好处与弊端

多线程的好处：解决了多部分同时运行的问题

多线程的弊端：线程太多回到效率降低

运行程序越多，cpu切换（切换随机）越慢，效率低

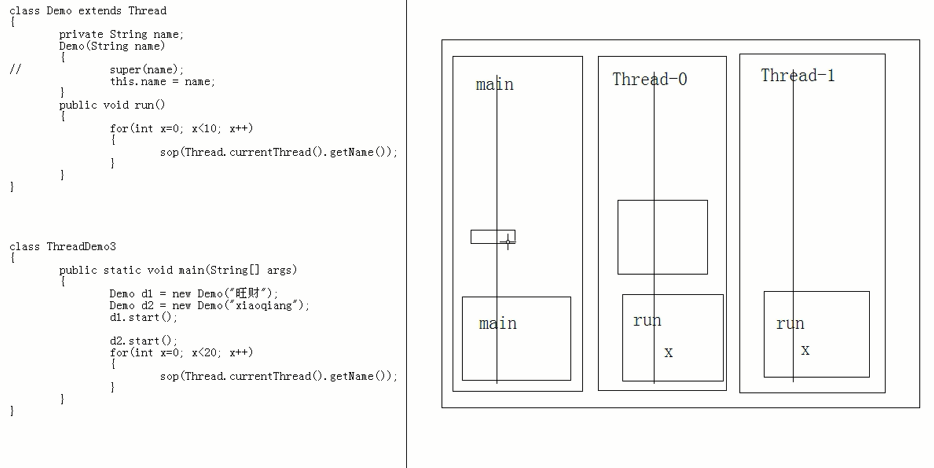
## JVM中的多线程解析

JVM启动时就启动了多个线程，至少有两个可以分析出来

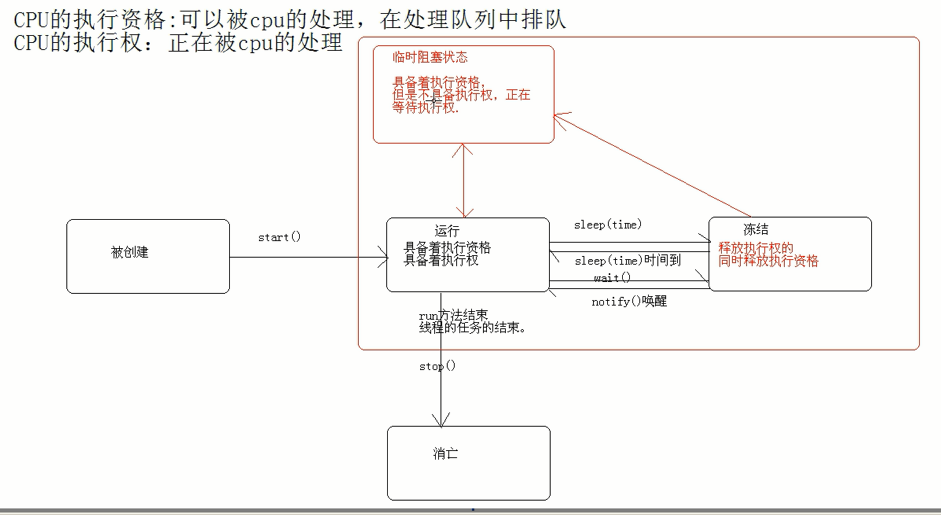
1. 执行main函数的线程

该线程的任务代码都是定义在main函数中。

1. 负责垃圾回收的线程



## 线程的状态



第一种创建进程的方式

\* 创建线程方式一：继承Thread类

\*

\* 步骤：

\* 1、定义一个类继承Thread类

\* 2、覆盖Thread类中的run方法

\* 3、直接创建Thread的子类对象创建线程

\* 4、调用start方法开启线程并调用线程的任务的run方法执行。

\*

\* 可以通过Thread的getName获取线程的名称 Thread-编号（从0开始）

\* Thread.currentThread().getName();//可以获取当前主线程名称mian

第二种创建进程的方式

/\*

\* 创建线程方式二：实现Runnable接口

\*

\* 步骤：

\* 1、定义一个类实现Runnable接口

\* 2、实现接口类中的run方法

\* 3、直接创建该类对象创建线程

\* 4、调用start方法开启线程并调用线程的任务的run方法执行。

\*

\* 可以通过Thread的getName获取线程的名称 Thread-编号（从0开始）

\* Thread.currentThread().getName();//可以获取当前主线程名称mian

\*/

## 多线程第二种方式的好处

1. 将线程的任务从线程的子类中分离出来，进行了单独的封装

按照面向对象的思想将任务封装成对象

1. 避免了java的单继承的特点

所以，创建线程的第二种方式较为常用

## 同步函数和同步代码块的区别

同步函数的锁是固定的this

同步代码块的锁是任意的对象

静态的同步函数使用的锁是该函数所属字节码文件对象，可以用getClass方法获取，也可以用当前 类名.class表示

建议使用同步代码块