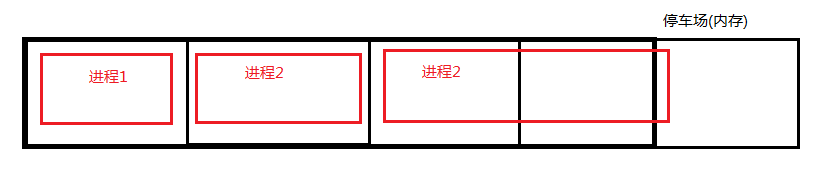
进程（Process）：

进程（Process）是计算机中的程序关于某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位，是[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F" \t "https://baike.baidu.com/item/%E8%BF%9B%E7%A8%8B/_blank)结构的基础。

（1个程序至少1个进程）



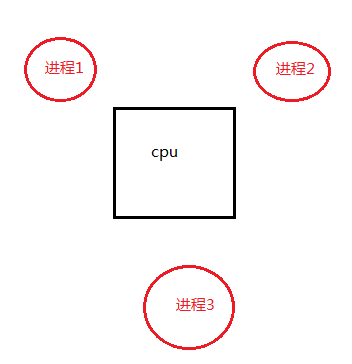
（1个程序至少1个进程，创建1个进程，系统就需要为它分配各种资源，每个进程具有独立性）

状态：

[就绪状态](https://baike.baidu.com/item/%E5%B0%B1%E7%BB%AA%E7%8A%B6%E6%80%81" \t "https://baike.baidu.com/item/%E8%BF%9B%E7%A8%8B/_blank) **运行状态** [阻塞状态](https://baike.baidu.com/item/%E9%98%BB%E5%A1%9E%E7%8A%B6%E6%80%81" \t "https://baike.baidu.com/item/%E8%BF%9B%E7%A8%8B/_blank)

（单个cpu只能执行一个进程）

切换存在间隔



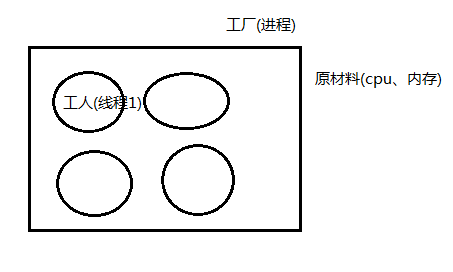
早期系统最小单元是进程。

弊端: 创建、撤消与切换占用资源

线程：

引入轻型进程，是程序执行流的最小单元，资源可共享。

在单个程序中同时运行多个线程完成不同的工作，称为[多线程](https://baike.baidu.com/item/%E5%A4%9A%E7%BA%BF%E7%A8%8B" \t "https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B/_blank)。



工厂相互独立、工人之间相互配合

例子：

进程是一个工厂，工厂有它的独立资源

- 工厂之间相互独立

- 线程是工厂中的工人，多个工人协作完成任务

- 工厂内有一个或多个工人

- 工人之间共享空间

工厂的资源 -> 系统分配的内存（独立的一块内存）

- 工厂之间的相互独立 -> 进程之间相互独立

- 多个工人协作完成任务 -> 多个线程在进程中协作完成任务

- 工厂内有一个或多个工人 -> 一个进程由一个或多个线程组成

- 工人之间共享空间 -> 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）

（1个程序至少1个进程，创建1个进程，系统就需要为它分配各种资源，每个进程具有独立性;

1个进程至少有1个线程，线程可以并发执行、资源共享。

）

并行：同时执行多个任务(多cpu),当一个CPU执行一个线程时，另一个CPU可以执行另一个线程，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)。

并发：同时请求多个任务，队列执行。

案例：

浏览器

浏览器是多进程的

页面、视频、动画、游戏、插件、测试

1，主进程（Browser进程）

2.第三方插件进程

3.GPU进程

4.浏览器渲染进程（浏览器内核，Render进程）

IE:Trident -ms-

火狐:Gecko -moz-

Safari/Chrome:Webkit -webkit-

Render进程

1.GUI渲染线程 （解析HTML，CSS，构建DOM树和RenderObject树，布局和绘制等）

2.JS引擎线程，[V8](http://www.966266.com/jishu/v8.html" \t "https://blog.csdn.net/mr_gly/article/details/_blank) (js解析主要靠它)

(不能同时进行)

3.事件触发线程 ,click、scroll

4.定时触发器线程 settimeout

5.异步http请求线程 xmlhttprequest