



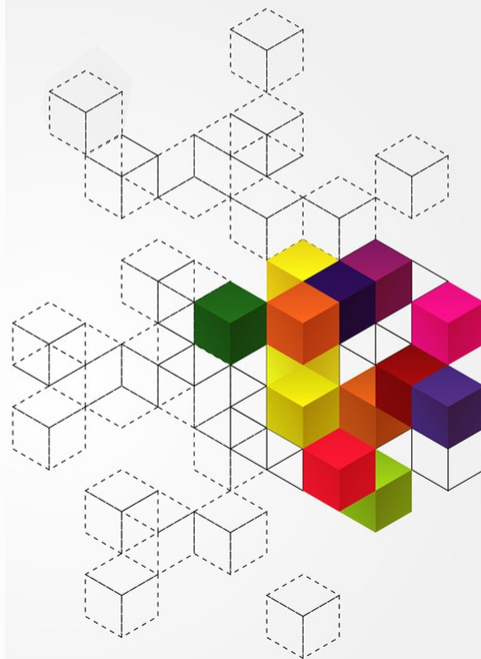
操作系统

Operating system

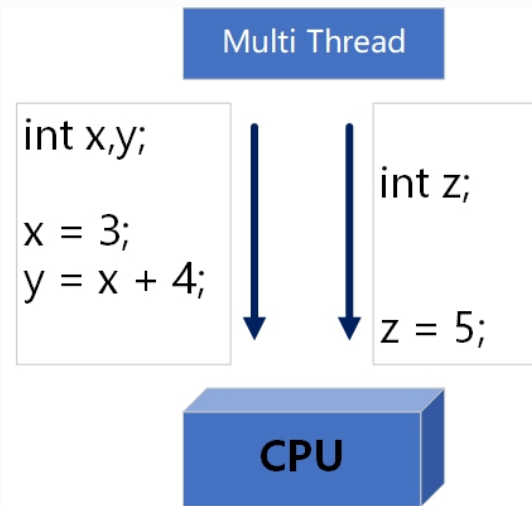
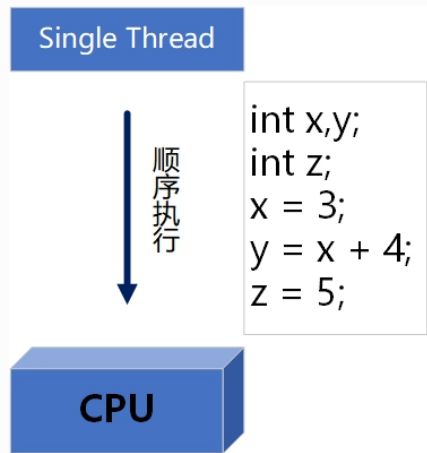
孔维强

大连理工大学

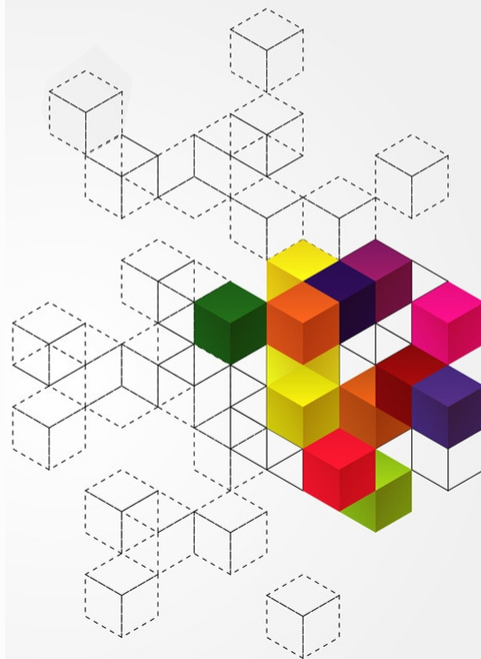
- 一、什么是线程
- 二、线程结构
- 三、线程 v.s. 进程
- 四、线程应用场景



一、什么是线程？

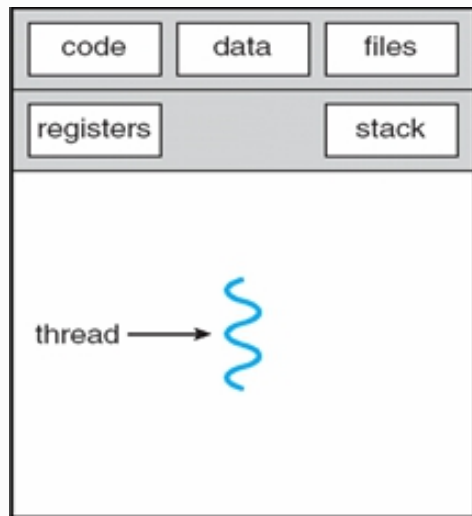


- **线程**是将进程的计算任务进一步细分后得到的更细粒度的计算单位

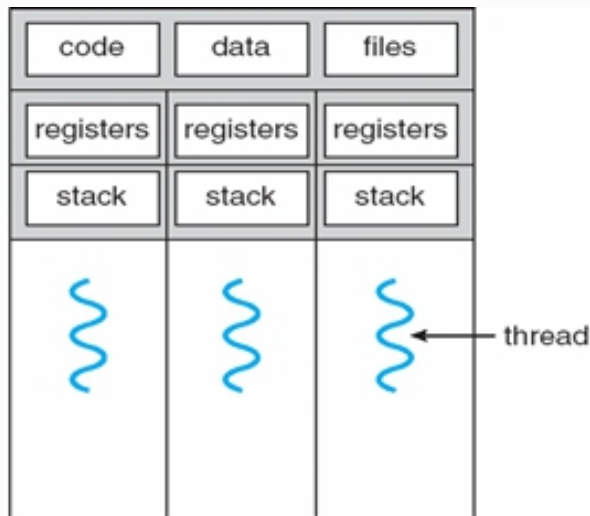


一、什么是线程？

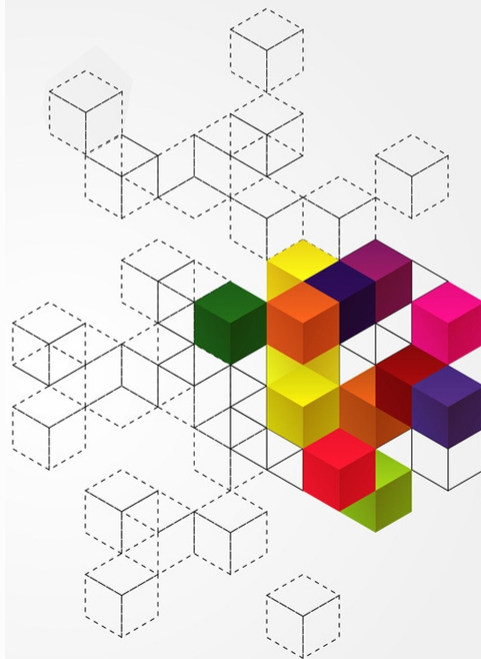
线程概念示意图：



单线程进程



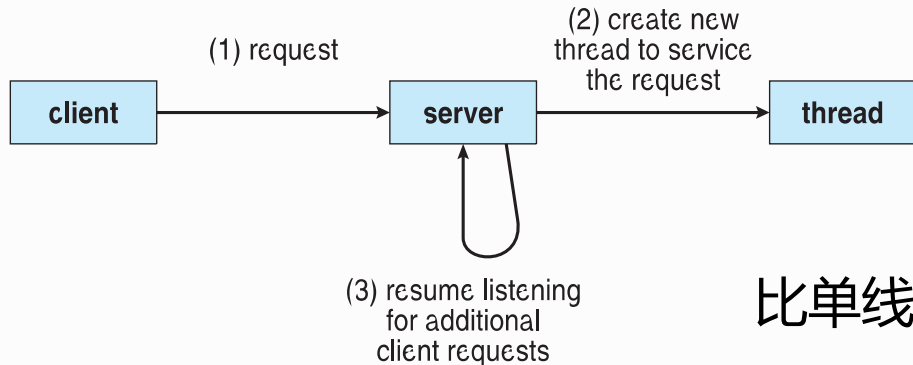
多线程进程



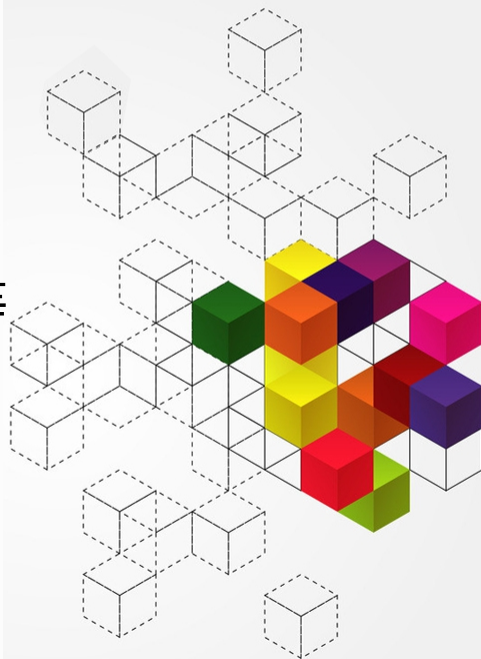
一、什么是线程？

为什么需要线程？

1. 现今应用大多是多线程的。
2. 应用内的多线程可独立完成不同功能
(word引用中：更新、读取数据、输入错误检查等)
3. 进程创建heavy-weight，线程创建light-weight
进程创建：分配资源、复制代码、进程上下文切换等

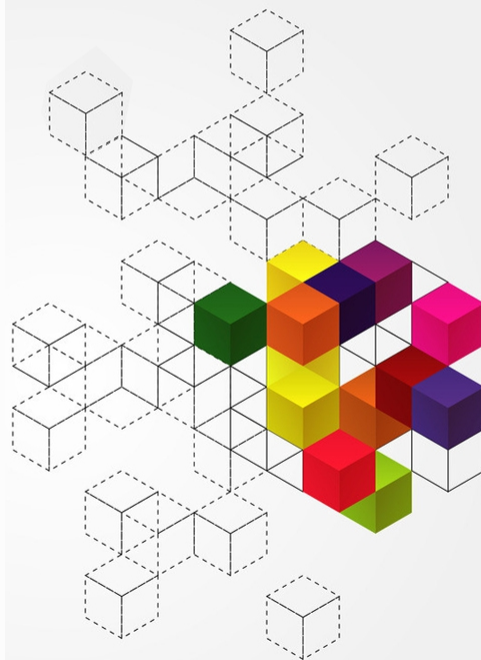
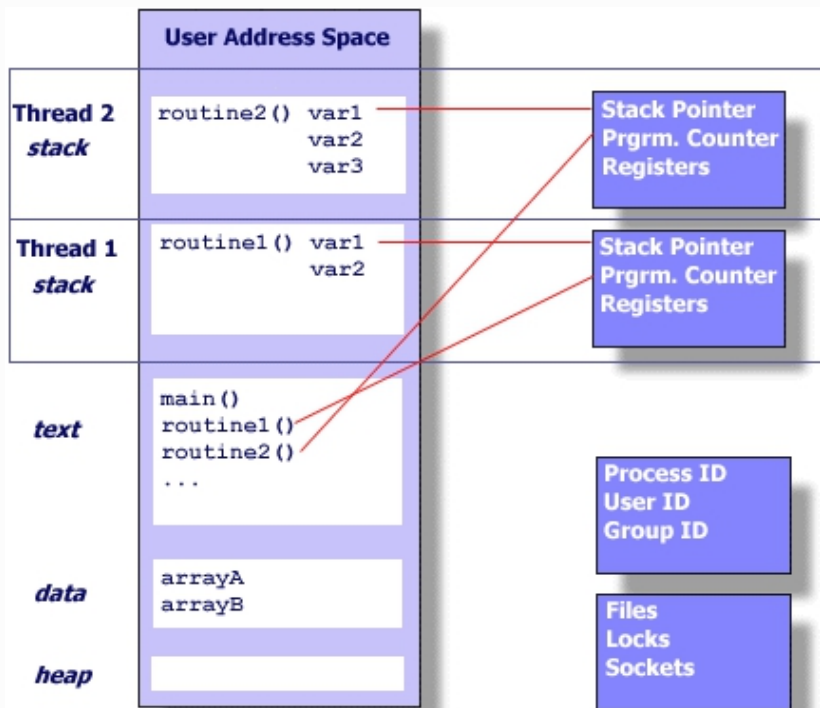


比单线程更高效



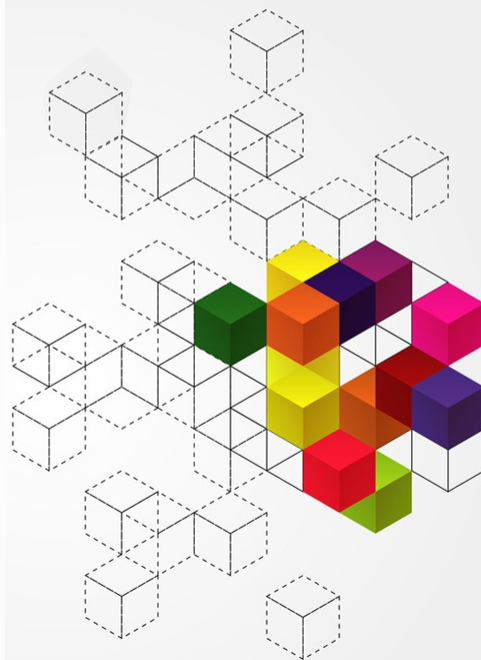
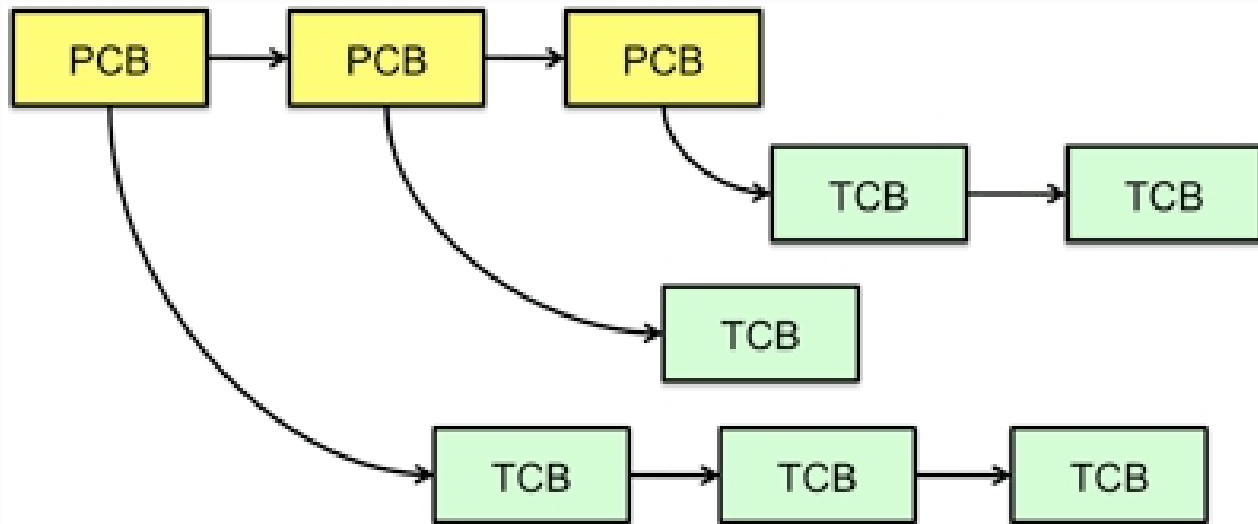
一、什么是线程？

多线程的进程地址空间示意



二、线程结构

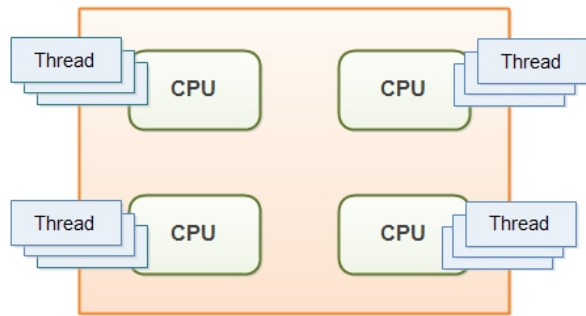
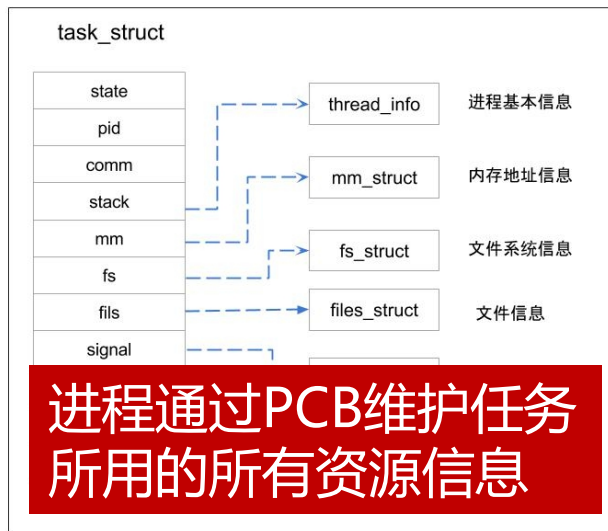
- OS内核中的线程管理数据结构：TCB
 - TCB(Thread Control Block)



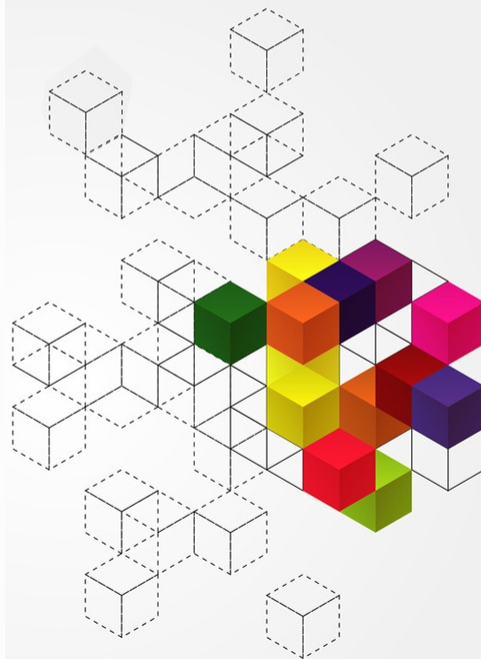
三、线程 v.s. 进程

• 线程与进程之间的联系

- 进程：拥有资源
- 线程：使用所隶属进程的资源进行计算

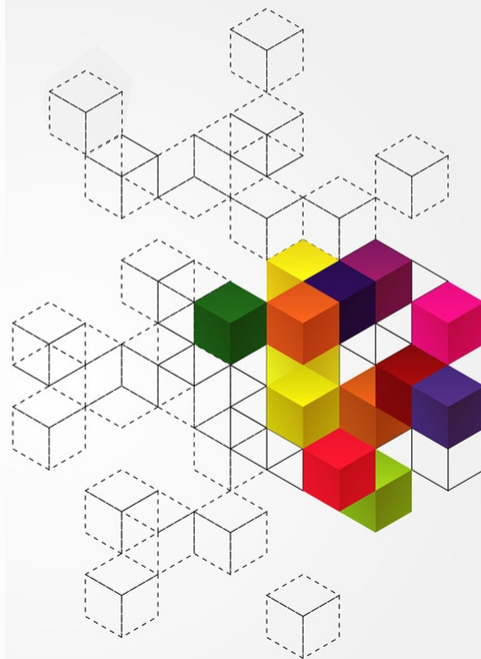
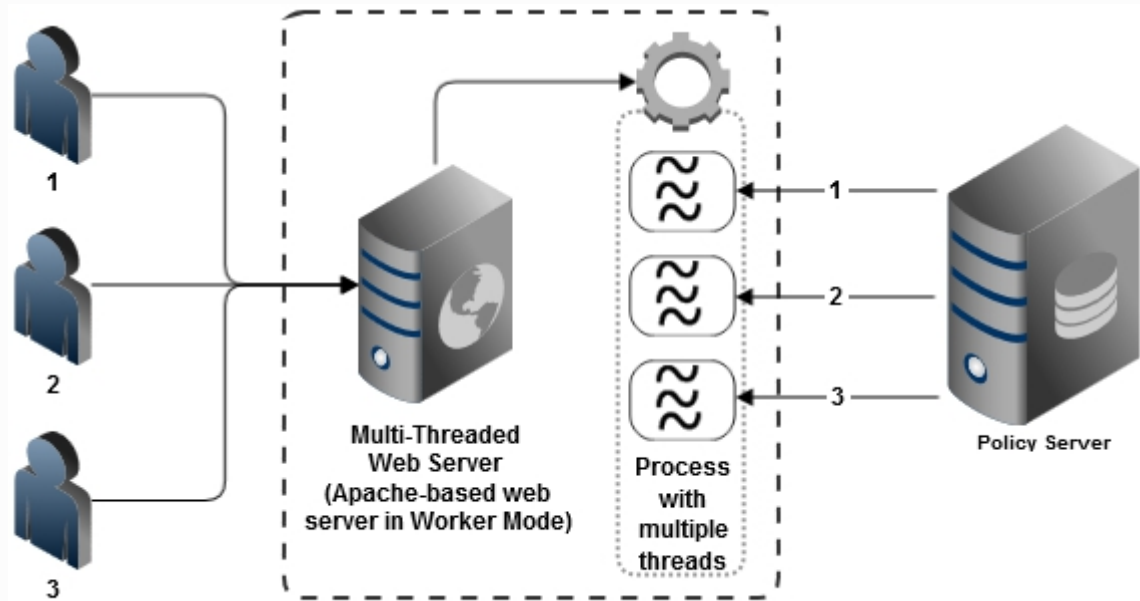


线程利用CPU资源
完成计算任务



MultiTasking v.s. MultiThreading

四、线程应用场景



本讲小结

- 什么是线程
- 线程结构
- 线程v.s.进程
- 线程典型应用场景

