

# 并发编程导论

- 基础篇
- 实战篇
- 案例篇
- 泛型篇

- 基础篇

# 并发与并行

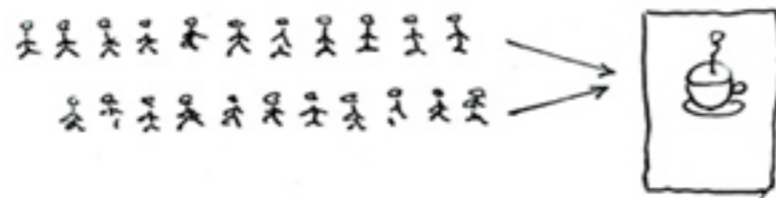
## Concurrent and Parallel Programming

05 Apr 2013

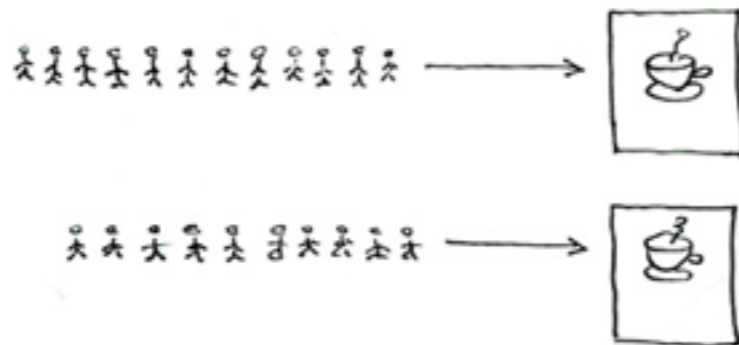
What's the difference between concurrency and parallelism?

Explain it to a five year old.

Concurrent = Two Queues One Coffee Machine



Parallel = Two Queues Two Coffee Machines



© Joe Armstrong 2013

**Concurrent** = Two queues and one coffee machine.

**Parallel** = Two queues and two coffee machines.

- 并发：逻辑上同时发生
- 并行：物理上同时发生
- 并行一定是并发的
- 并发比并行更好

# 并发编程的好处

- 资源利用率提升
- 更精简的程序设计
- 更高的响应性能

# 并发编程的问题

- 设计更复杂
- 上下文切换
- 消耗资源

# 并发编程基础知识

- 硬件基础
- 软件基础

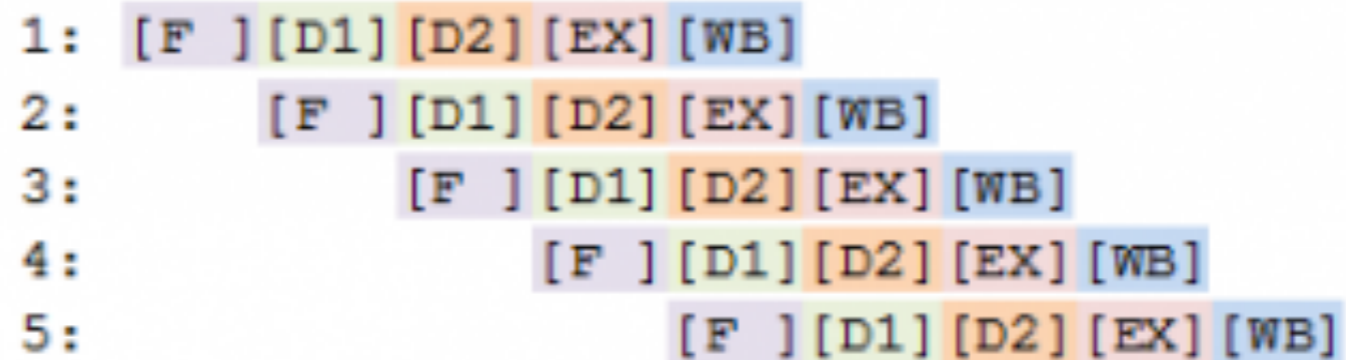
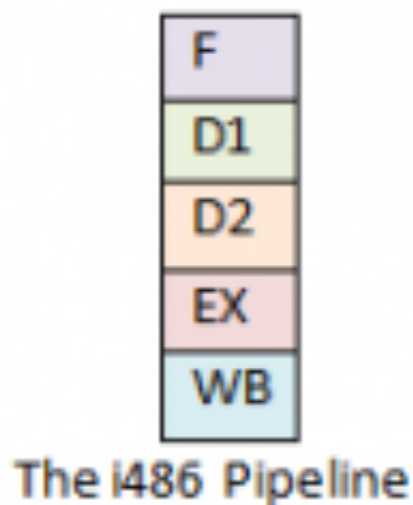
# 硬件基础

- CPU如何执行指令
- CPU如何访问数据
- CPU内存模型（何为内存模型？）



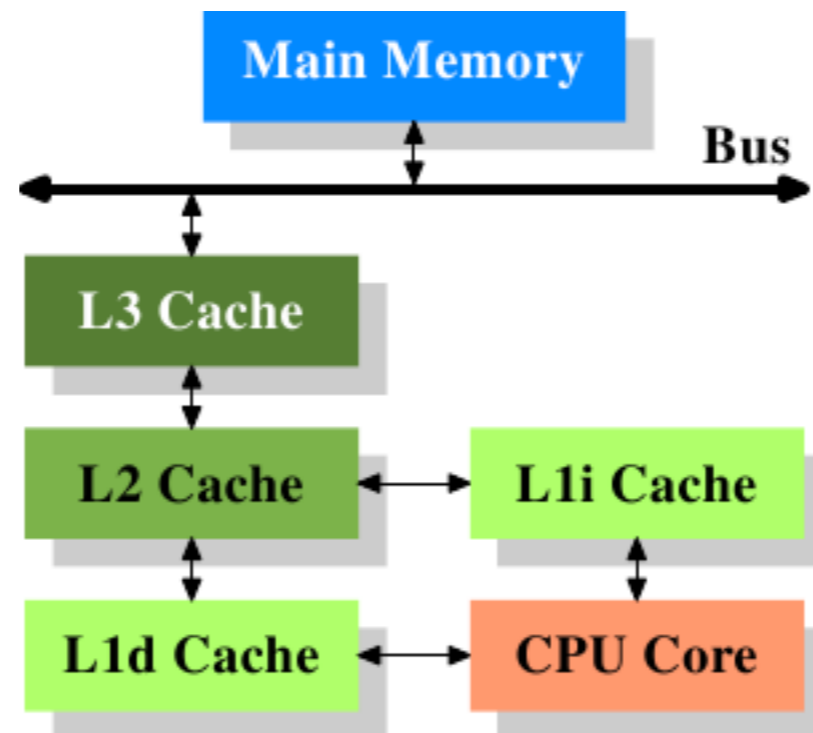
# CPU如何执行指令

- 处理器流水线基础（X86示意图，非某种CPU）
- 五级流水线为例。分别是：取指(Fetch)，译码(D1, main decode)，转址(D2, translate)，执行(EX, execute)，写回(WB)
- 乱序执行



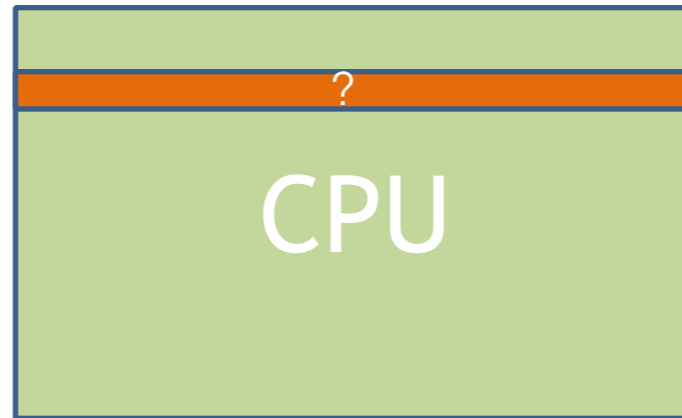
# CPU如何读取数据

- CPU的多级缓存架构
- 缓存行

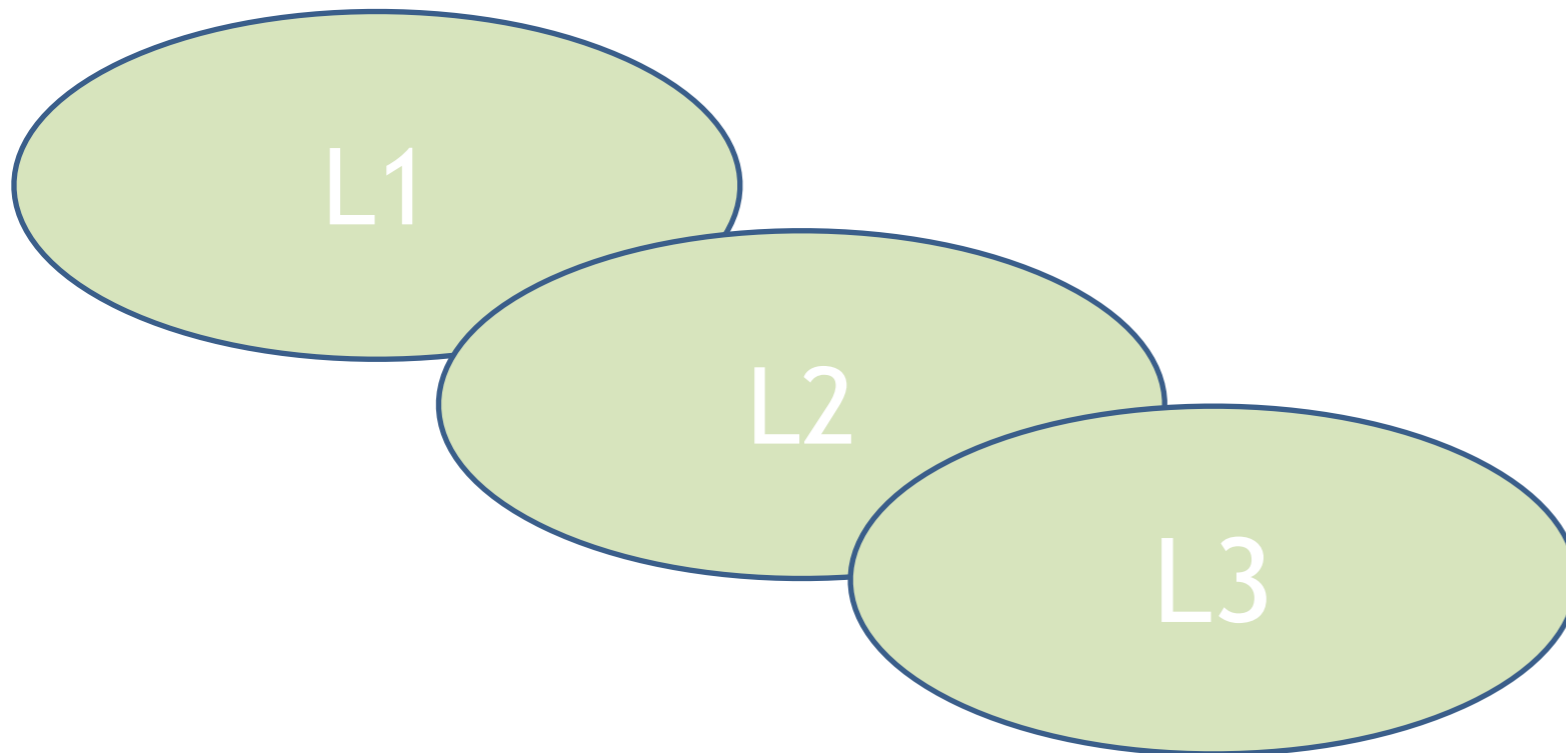


# 附录A-缓存寻址

cpu和缓存，它们的结构是  
怎么样的呢？平时他们是  
怎么样为我们服务的呢？



这条cache line如  
何找到它在缓存  
中是否存在，并  
且是存在哪个位  
置上呢？



# 附录A-缓存寻址

- 全相联
- 组相连
- 直接相连

先看看缓存的结构吧：CPU是分为多个连续的“组”的，每个组里边又有多个缓存行(cache line)，

组0							
组1							
组2							
.....							
组62							
组63							
64B	64B	64B	64B	64B	64B	64B	64B

因为是8路，8-ways这个是CPU厂商定好的

一个L1 Cache是32KB

一个cache line是64B

$$64 * 64 * 8$$

所以，L1里共有64个组

因为是8路，所以，一个组里有8条缓存行



# CPU内存模型

- 一套规则：当这个（或其他）处理器执行指令时，处理器能够看见什么值
- 内存模型分类：intel是强一致内存模型



# CPU写数据

- 回写 VS 直写
- 存储缓存 (Store Bufferes)
- 无效队列



# 软件基础 (Linux)

- 进程
- 内核线程
- 轻量级进程
- 线程
- 协程

# JAVA中的并发编程