



第2章 进程与线程

硬件基础运行环境



第2章 进程与线程

2.1 处理器概述

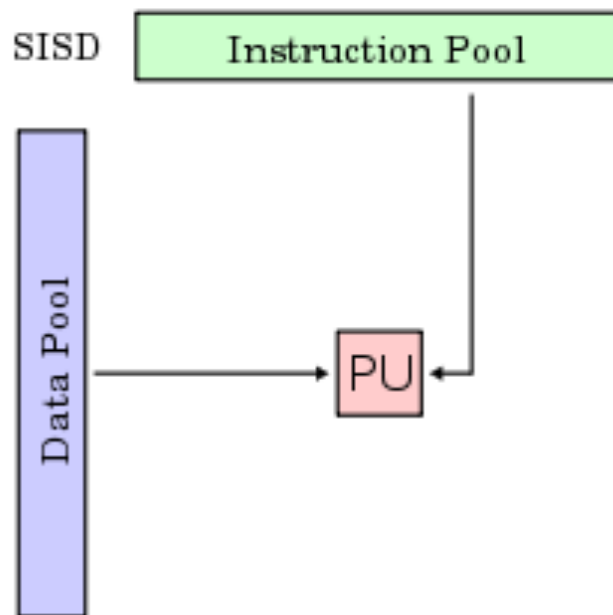


2.1.1 单处理器和多处理器系统

- 计算机系统的核心是中央处理器
 - 中央处理器的任务：
 - 取指→译码→取操作数→执行指令
 - 单处理器系统：一个计算机系统只包括一个运算处理器。
 - 多处理器系统：一个计算机系统有多个运算处理器。

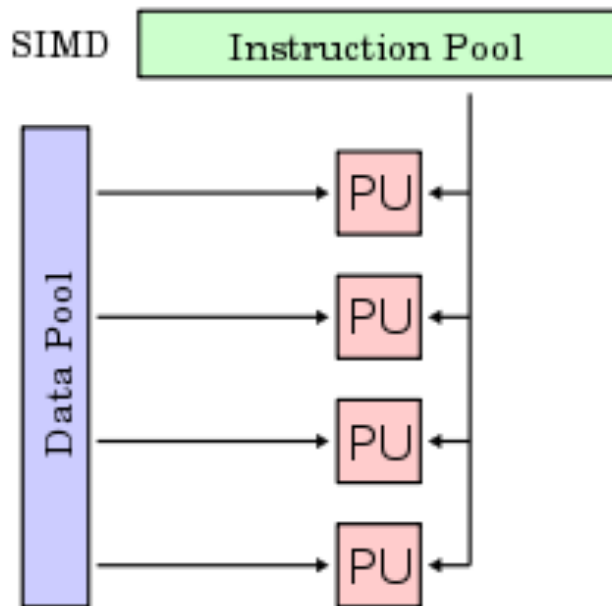
计算机系统结构分类

- 单指令流单数据流(SISD): 一个处理器在一个存储器中的数据上执行单条指令流



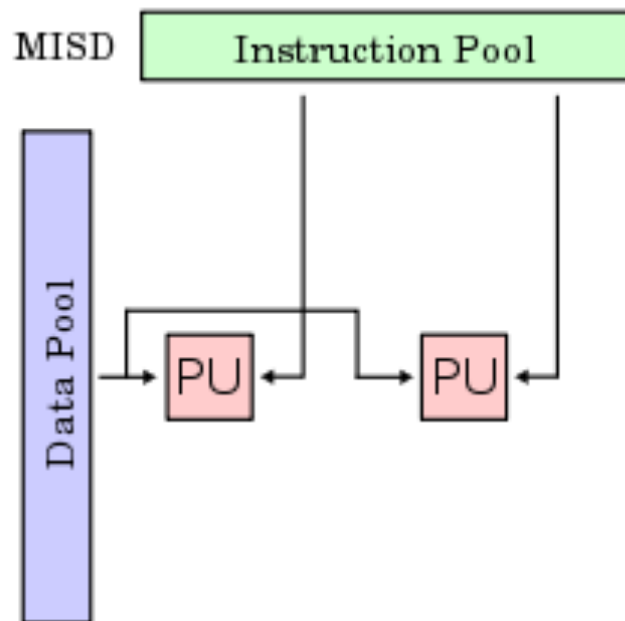
计算机系统结构分类

- 单指令流多数据流(SIMD): 单条指令流控制多个处理单元同时执行, 每个处理单元包括处理器和相关的数据存储, 一条指令控制了不同的处理器对不同的数据进行操作。向量机和阵列机是这类计算机系统的代表



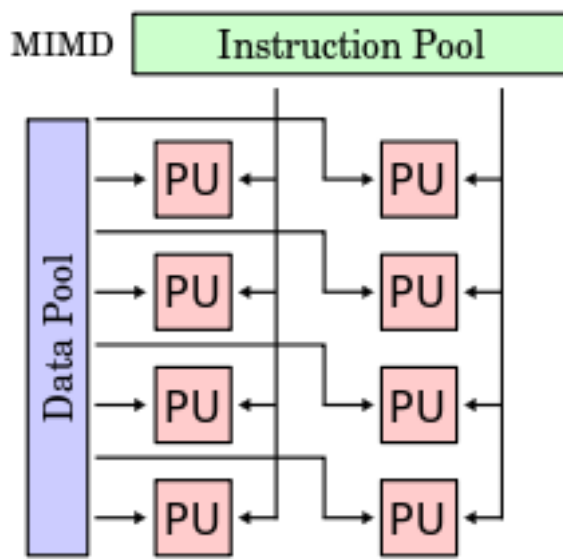
计算机系统结构分类

- 多指令流单数据流(MISD): 一个数据流被传送给一组处理器, 通过处理器上不同指令操作最终得到处理结果



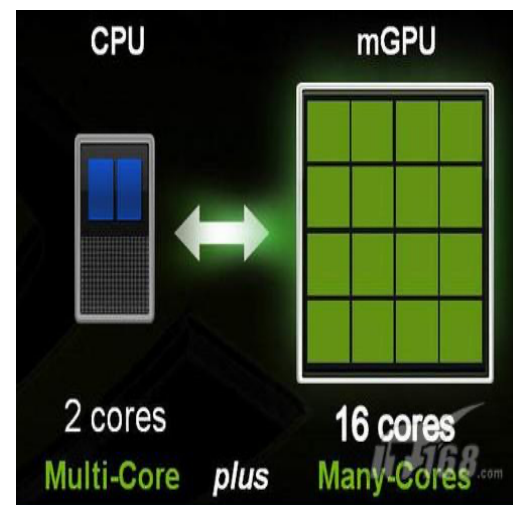
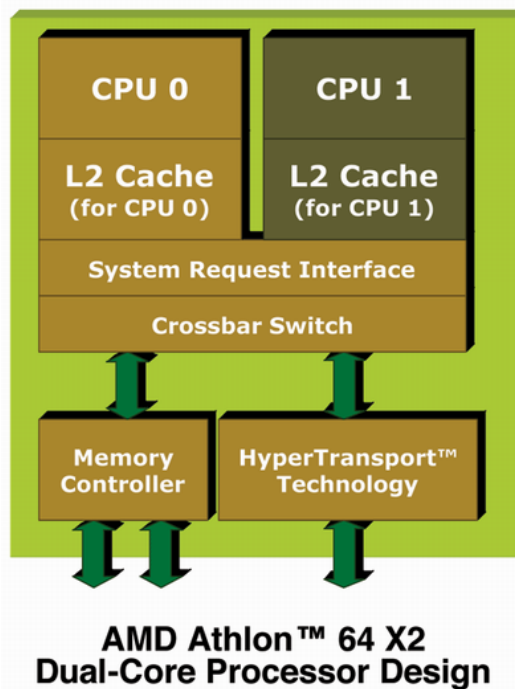
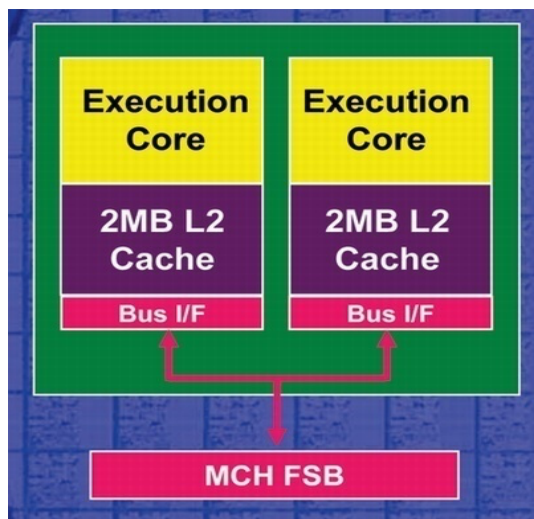
计算机系统结构分类

- 多指令流多数据流(MIMD): 多个处理器对各自不同的数据集同时执行不同的指令流。可以把MIMD系统划分为共享内存紧密耦合系统和内存分布松散耦合系统两类（共享内存型还可再分为MSP(Master/Slave)和SMP(Symmetric)）



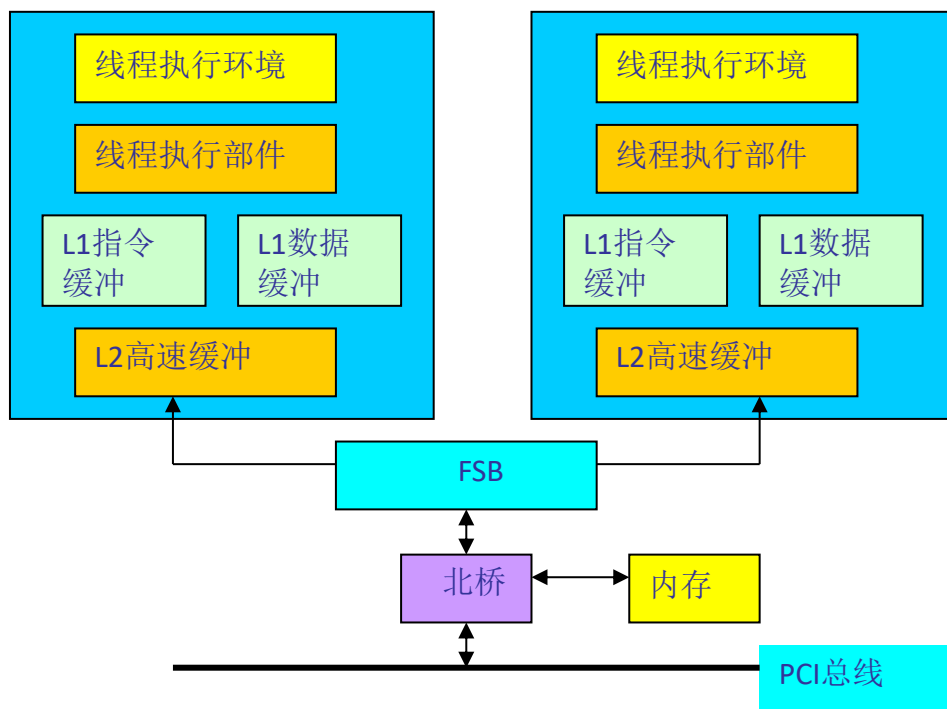
多处理器和多核处理器

- 多处理器指在一个体系结构上放置多个(单核)CPU芯片，而多核则指在同一块CPU芯片上放置多个核（core），即执行单元。
- 多CPU和多核的区别是后者更加紧凑，成本更低、功耗更小。



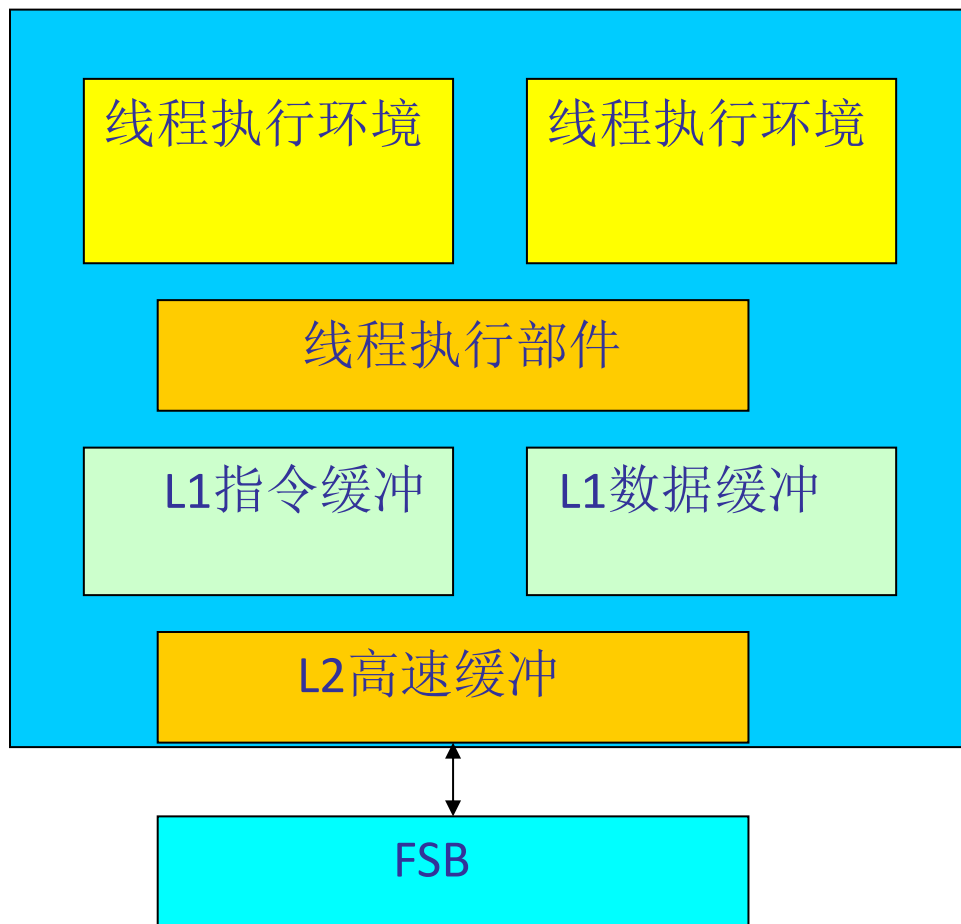
多处理器和多核处理器

■ (1)多处理器结构



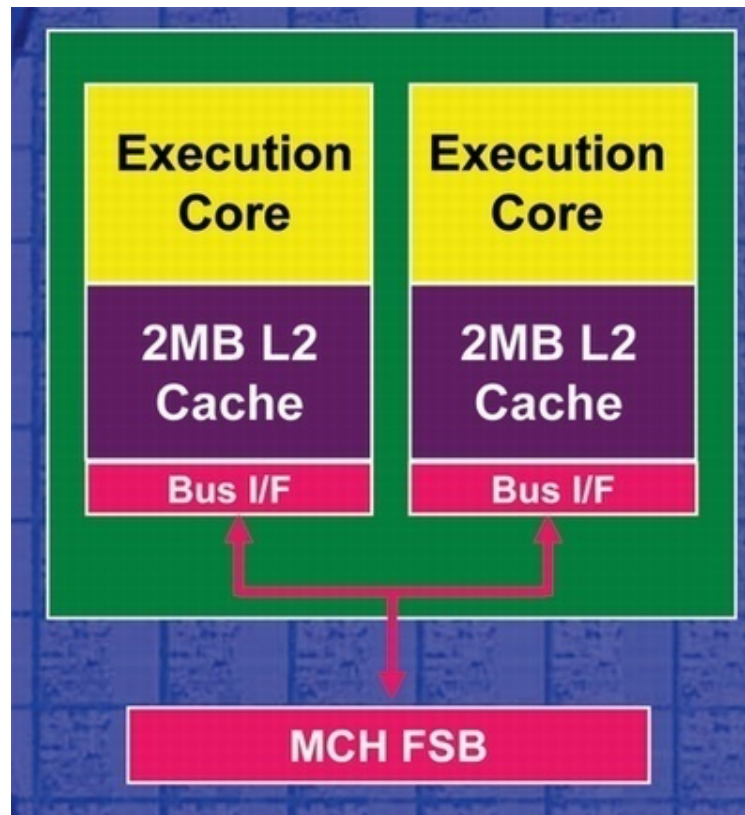
多处理器和多核处理器

■ (2)超线程结构



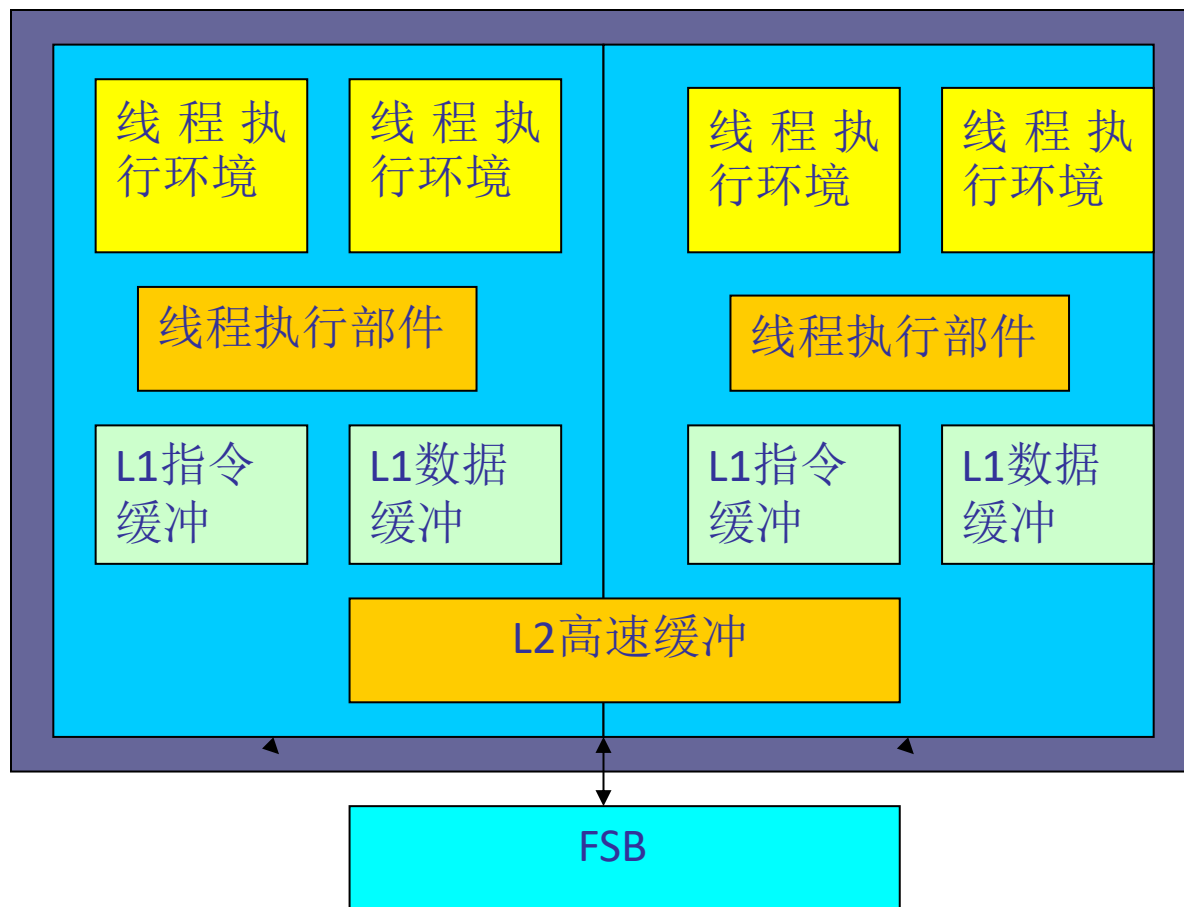
多处理器和多核处理器

■ (3)多核结构



多处理器和多核处理器

■ (4)多核超线程结构





多处理器和多核处理器

- 操作系统与多核处理器的关系
 - 处理器通信支持
 - 进程/线程数据共享支持
 - 存储器层次及管理
 - 程序并行执行模型支持
 - 同步支持
 - 调度及优化
 - 能耗管理





2.1.2寄存器(1)

- 计算机系统的处理器包括一组寄存器，其个数根据机型的不同而不同，它们构成了一级存储，比主存容量小，但访问速度快。
- 这组寄存器所存储的信息，与程序的执行有很大关系，构成了**处理器现场**。

寄存器(2)

- 按照功能分类：x86结构为例
 - 通用寄存器-- EAX, EBX, ECX和EDX
 - 指针及变址寄存器--ESP, EBP, ESI及EDI
 - 段选择符寄存器--CS、DS、SS、ES、FS、GS
 - 指令指针寄存器和标志寄存器--EIP、EFLAGS
 - 控制寄存器--CR0, CR1, CR2和CR3
 - 外部设备使用的寄存器



2.1.3 特权指令与非特权指令(1)

机器指令的集合称指令系统

① 数据处理类指令

- 执行算术和逻辑运算

② 转移类指令

- 改变指令的执行序列

③ 数据传送类指令

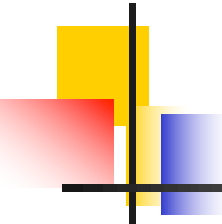
- 处理器的寄存器间、寄存器与主存单元间、主存单元间数据交换

④ 移位与字符串指令

⑤ I/O类指令

特权指令与非特权指令(2)

- 从资源管理和控制程序执行的角度出发，必须把指令系统中的指令分作两部分
 - 特权指令：只能提供给操作系统的核心程序使用的指令
 - 非特权指令：其他普通功能指令



特权指令做了一些什么？

- 设置定时器
- 读取时钟
- 清除内存
- 发起陷入指令
- 关中断
- 修改设备状态信息
- 用户与内核态切换
- 访问I/O设备

中央处理器如何判定特权指令是否能够执行呢？

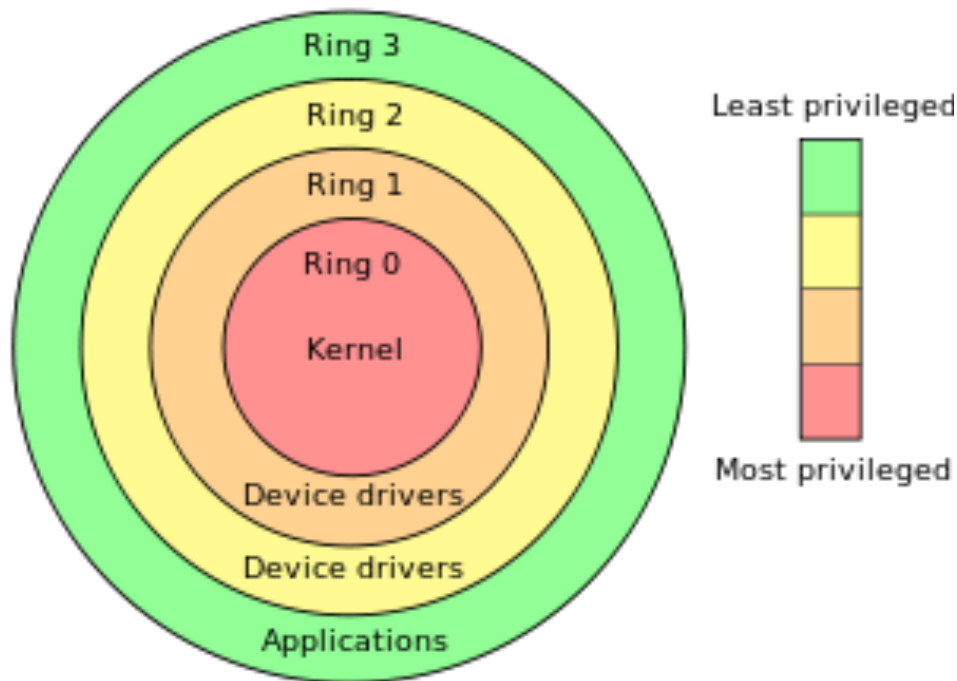
2.1.4 处理器状态

- 处理器状态标志和设置处理器成不同状态：
 - 管理状态（特权状态、系统模式、特态或管态）
 - 用户状态（目标状态、用户模式、常态或目态）
- 处理器处于管理状态时
 - 程序可以执行全部指令，使用所有资源，具有改变处理器状态的能力；
- 处理器处于用户状态时
 - 程序只能执行非特权指令
- 操作系统内核工作在管理态，用户程序工作在用户态
 - 特权指令只允许OS内核执行
- CPU通常用受保护的寄存器来区分管理态和用户态



OS保护

- Intel Pentium的处理器状态有四种，支持4个保护级别，0级权限最高，3级权限最低



- 内核态运行在Ring0
- 用户态运行在Ring3





内存保护

- OS必须做到不同进程间的内存隔离
- OS内核占用的内存不能被用户态的进程破坏
- 硬件支持的内存保护
 - 基址与限址寄存器
 - 页表寄存器/页属性/页保护
 - 段描述符/段属性
 - 大小、基址、特权级、代码/数据
 - 通过特权级检查，实现内存的隔离与保护



处理器状态的切换

应用程序和内核程序相互隔离保护了，怎么实现相互联系呢？

- 引起处理器状态切换的原因（用户态→核心态）：
 - 程序请求操作系统服务，执行系统调用
 - 程序运行产生中断或者异常事件，程序被中断，转向中断处理程序或异常处理程序
- 状态切换步骤
 - 保存中断处理器现场
 - 根据中断号设置程序计数器
 - 交换PSW，转向中断处理程序

2.1.5 程序状态字寄存器(1)

- 计算机如何知道当前处于何种工作状态？这时能否执行特权指令？
 - 程序状态字：**PSW**（**Program Status Word**），用于区别不同的处理器工作状态
 - 控制指令执行顺序
 - 保留和指示与程序有关的系统状态
 - 实现程序状态的保护和恢复
- 程序和处理器的配合：
 - 每个程序都有一个/组与其执行相关的**PSW**
 - 每个处理器都设置一个/组**PSW**寄存器
 - 程序占有处理器执行，它的**PSW**将占有**PSW**寄存器



程序状态字寄存器(2)

■ PSW寄存器包括以下内容：

■ 程序基本状态：

- 程序计数器；
- 条件码；
- 处理器状态位。

■ 中断码。保存程序执行时当前发生的中断事件。

■ 中断屏蔽位。指明程序执行中发生中断事件时，是否响应出现的中断事件。



2.1.6 x86总体架构

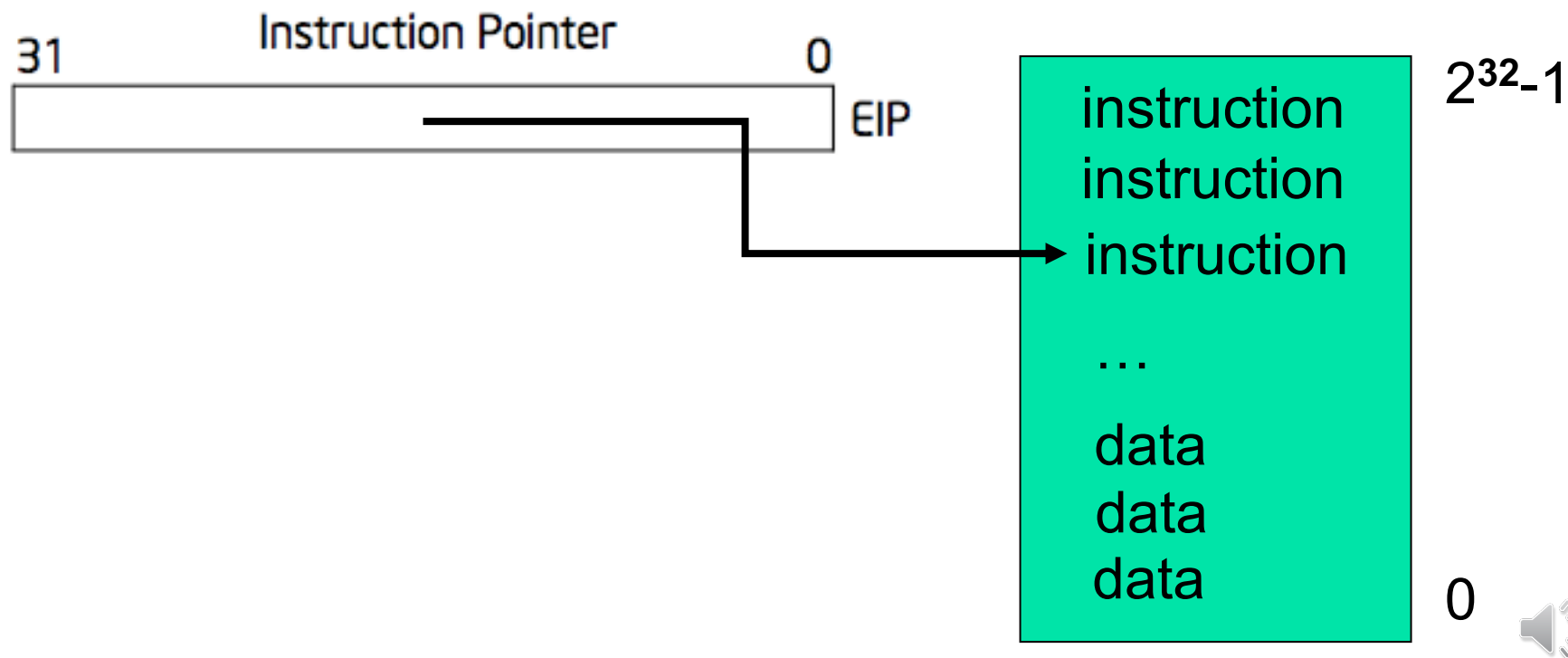
- 通用寄存器
- 8, 16, and 32 bit versions
- 堆栈平衡目的: %EBP, %ESP

General-Purpose Registers							
31	16	15	8	7	0	16-bit	32-bit
	AH		AL			AX	EAX
	BH		BL			BX	EBX
	CH		CL			CX	ECX
	DH		DL			DX	EDX
	BP						EBP
	SI						ESI
	DI						EDI
	SP						ESP



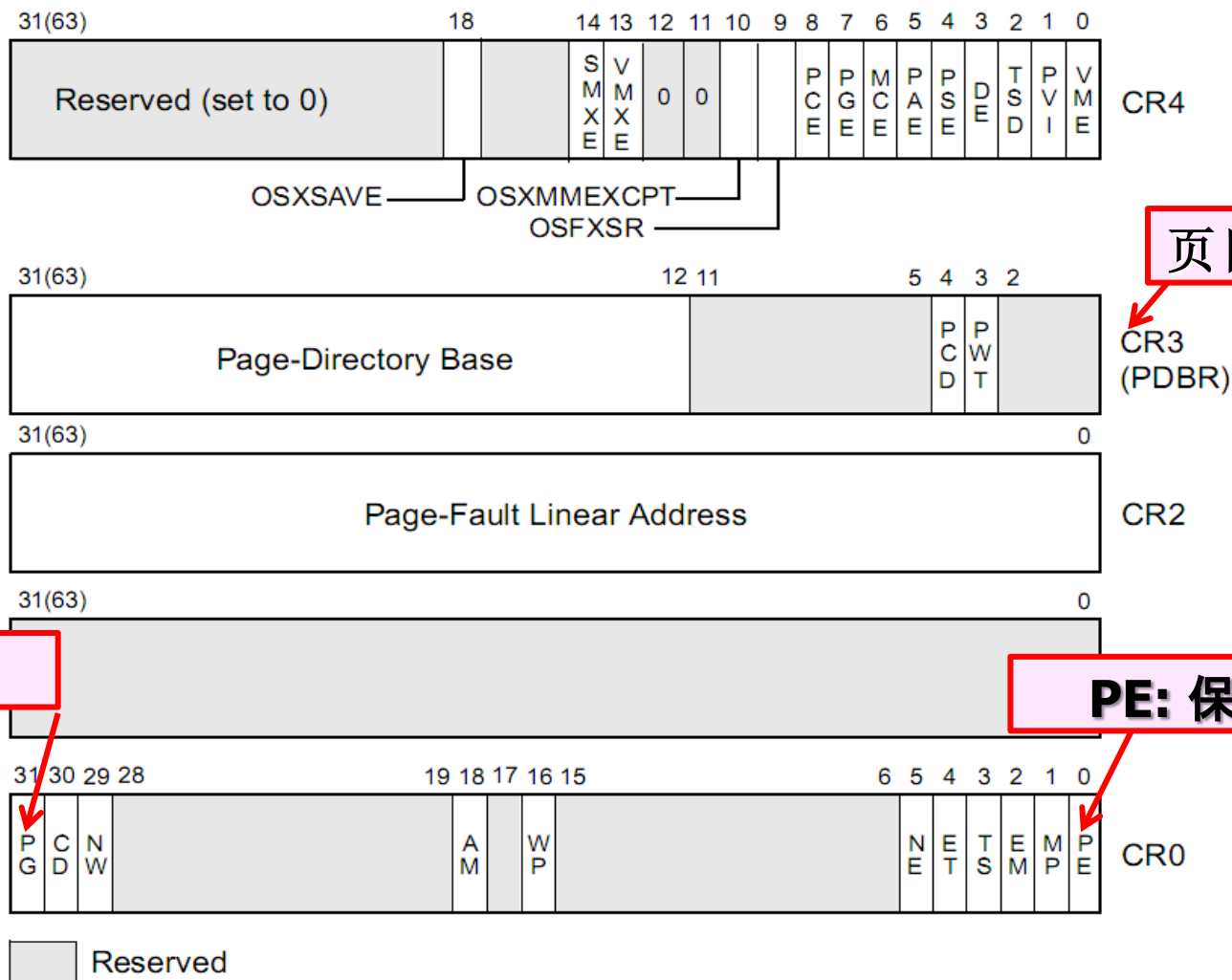
2.1.6 x86总体架构

- **EIP** 执行指令在内存中的地址
- 指令长度不是固定的
- EIP通常可以被 **CALL, RET, JMP**等控制

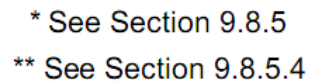


2.1.6 x86总体架构

■ 控制寄存器



- 实模式
- 保护模式
- **SMM**模式





2.1.6 x86总体架构

- 8086: 16-bits 处理器

- 可寻址空间: 64KB

- 地址总线 20-bits

- 最大寻址空间: 1MB

- 16位如何与20位匹配

- 引入段寄存器，分两次读进地址：段基址+偏移
 - 实模式寻址： $\text{physical addr} = 16 * \text{segment} + \text{offset}$
 - CS: code segment, for EIP
 - SS: stack segment, for SP and BP
 - DS: data segment for load/store via other registers
 - ES: another data segment, destination for string ops
 - *e.g. CS=f000 IP=fff0 => ADDR: ffff0*





2.1.6 x86总体架构

- 80386: 32-bit data and bus addresses

- 保护模式

- 向前兼容性

- 8086是16位的处理器+20位的地址总线
 - 为了兼容.....

- 从实模式启动，切换到保护模式
 - 一直沿用至今





2.1.6 x86总体架构

- 与内存管理相关的寄存器
 - GDTR (全局描述符寄存器)
 - 段属性
 - IDTR (中断描述符寄存器)
 - 中断响应程序的地址
 - 保护级别
 - TR (任务寄存器)
 - 与进程相关
 - 用户进程切换时的状态保存地址
 - TSS



2.1.6 x86总体架构

■ 与程序状态相关的寄存器

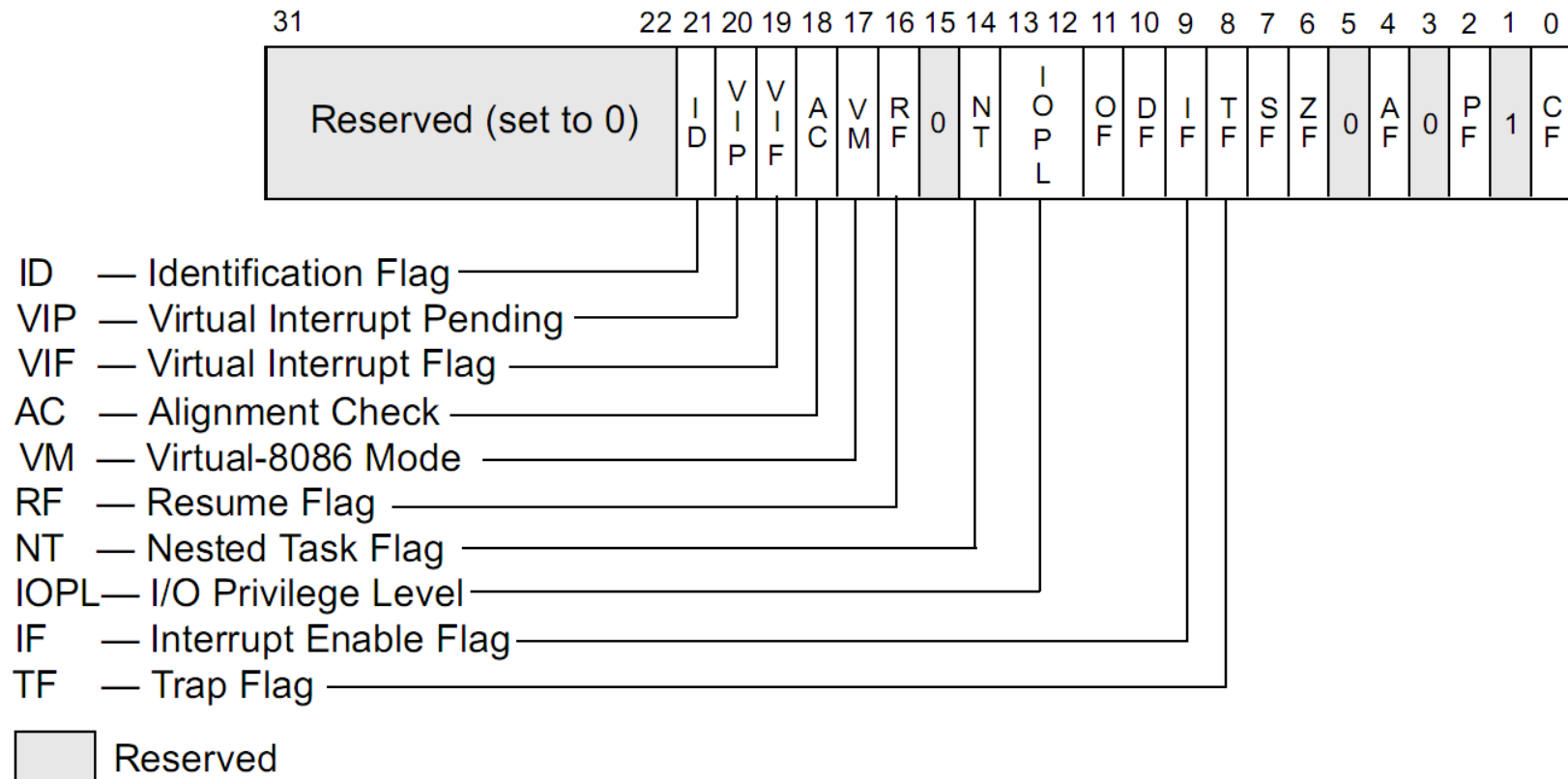
■ Intel Pentium中，PSW由**标志寄存器EFLAGS**和**指令指针寄存器EIP**组成，均为32位。

■ EFLAGS的低16位称**FLAGS**，标志可划分为三组：

- 状态标志：CF, PF, AF, ZF, OF, SF
- 控制标志：VM, TF, IF(中断允许) 等
- 系统标志：IOPL（控制I/O访问）

2.1.6 x86总体架构

■ System Flags in the EFLAGS





讨论问题

1. 在支持多道任务处理时，单核处理器、多核处理器、多重处理器从微观上看是串行还是并行，为什么？
2. 当应用程序和内核程序分别受到攻击，可能存在哪些风险，各有什么不同？