

计算机系统结构

第一课： 系统结构预基础

王韬

wangtao@pku.edu.cn

<http://ceca.pku.edu.cn/wangtao>

2018

第一课： 系统结构预基础

- 计算机发展简史
- 计算机硬件概述
- 系统结构与操作系统

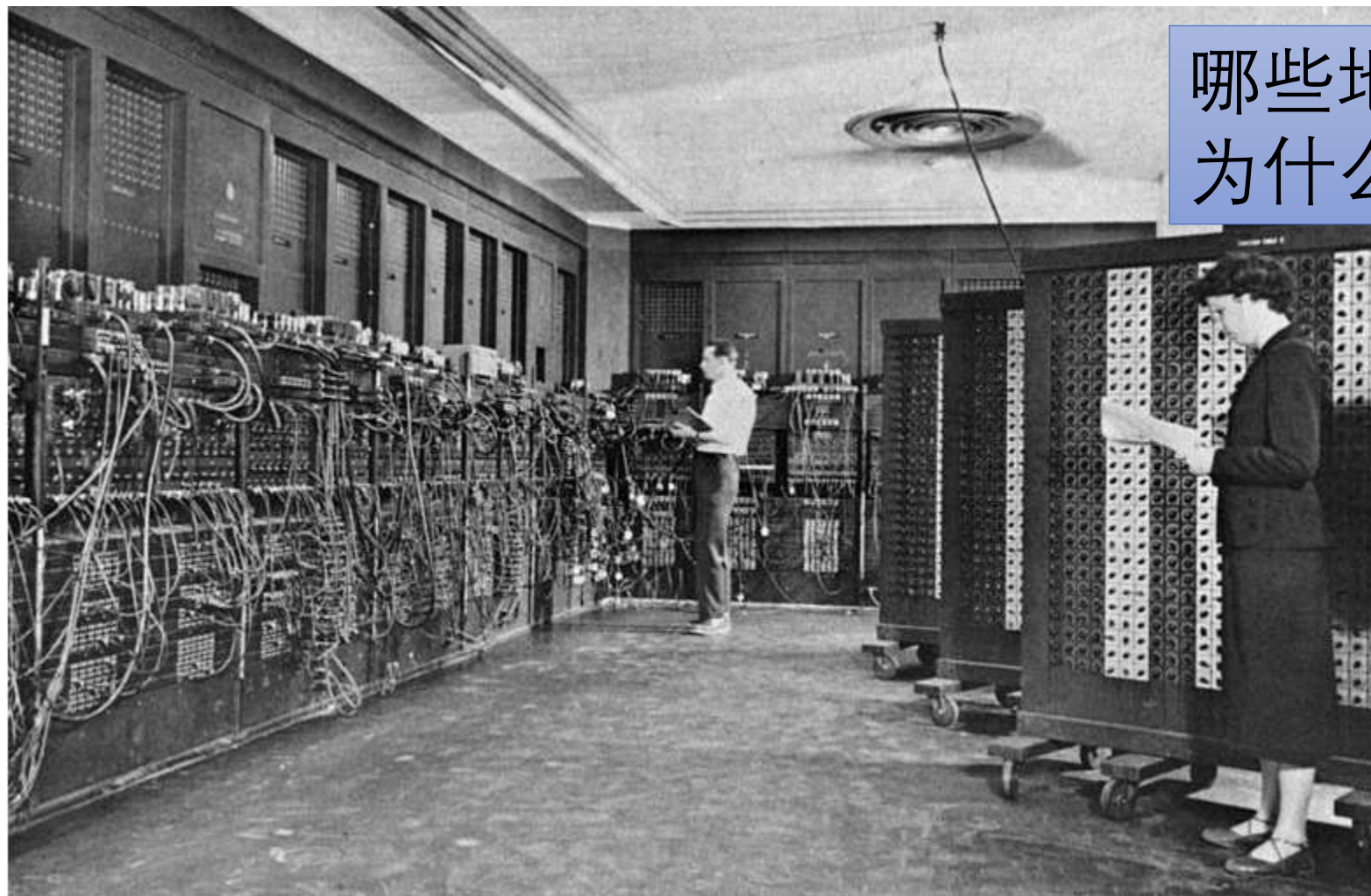
第一课： 系统结构预基础

- 计算机发展简史
- 计算机硬件概述
- 系统结构与操作系统

四代计算机系统

- 第一代：电子管计算机（1945-55）
- 第二代：晶体管计算机（1955-65）
- 第三代：集成电路计算机（1965-1980）
- 第四代：个人计算机（1980-现在）

第一代：电子管计算机



哪些地方应该被改进？
为什么？

Glen Beck (background) and Betty Snyder (foreground) program ENIAC in BRL building 328. (U.S. Army photo)

第二代：晶体管计算机 1/2



• IBM 7090

哪些地方应该被改进？
为什么？

第二代：晶体管计算机 2/2

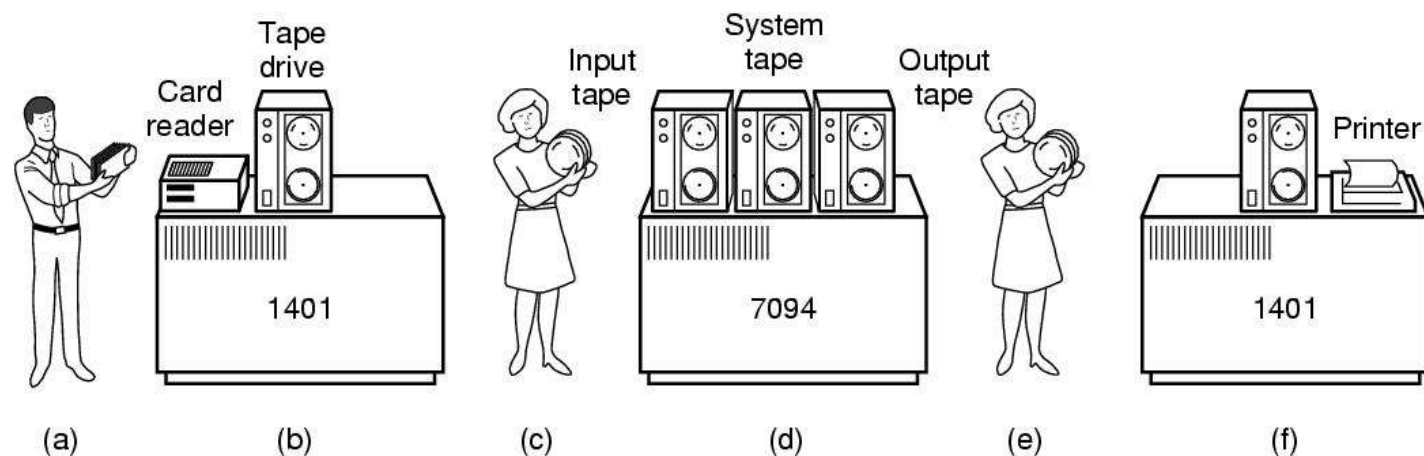
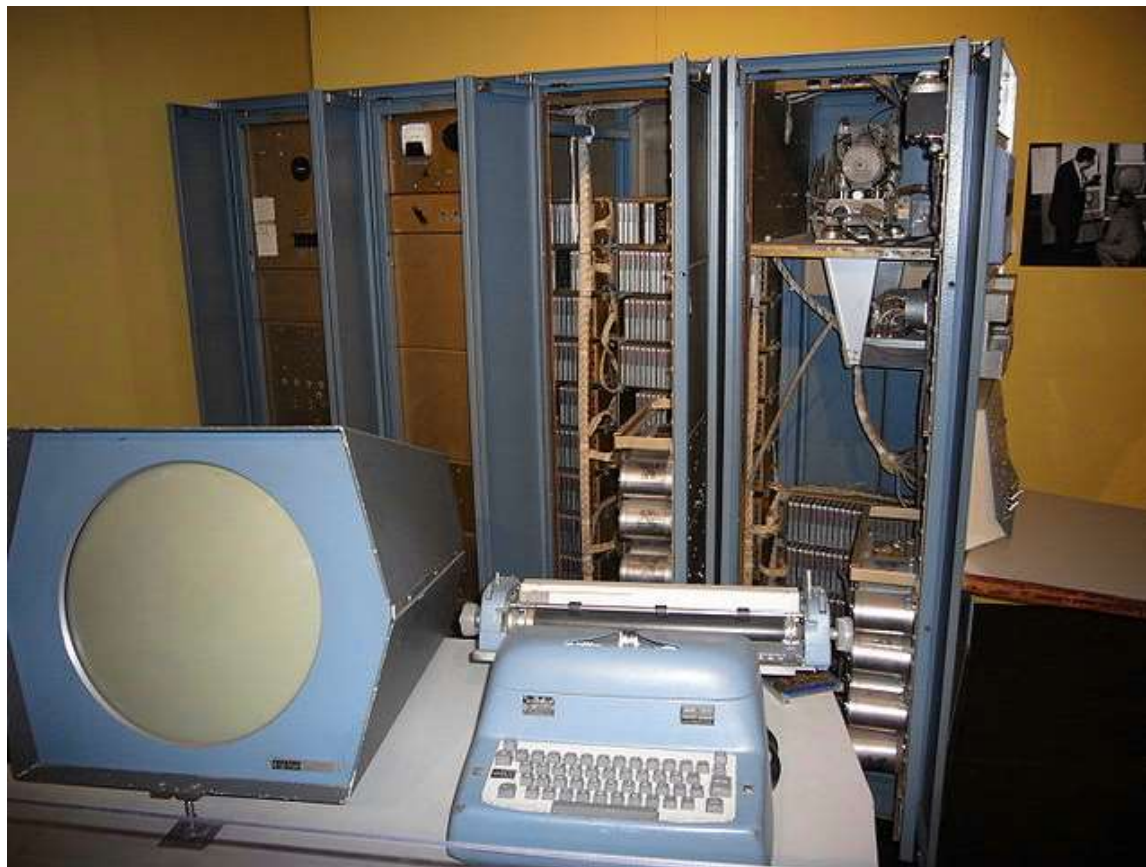


Figure 1-3. An early batch system.

- (a) Programmers bring cards to 1401
- (b) 1401 reads batch of jobs onto tape
- (c) Operator carries input tape to 7094
- (d) 7094 does computing.
- (e) Operator carries output tape to 1401
- (f) 1401 prints output

第三代：集成电路计算机 1/2



IBM System/360

SPOOLing

Time-sharing

MULTICS

.....

Mini-computer

PDP-1, 7, 11,...

Ken Thompson

UNIX

- The PDP-1 (Programmed Data Processor-1, DEC) was first produced in 1960. It was also the original hardware for playing history's first game on a minicomputer, Steve Russell's Spacewar!. – Source: Wikipedia

第三代：集成电路计算机 2/2

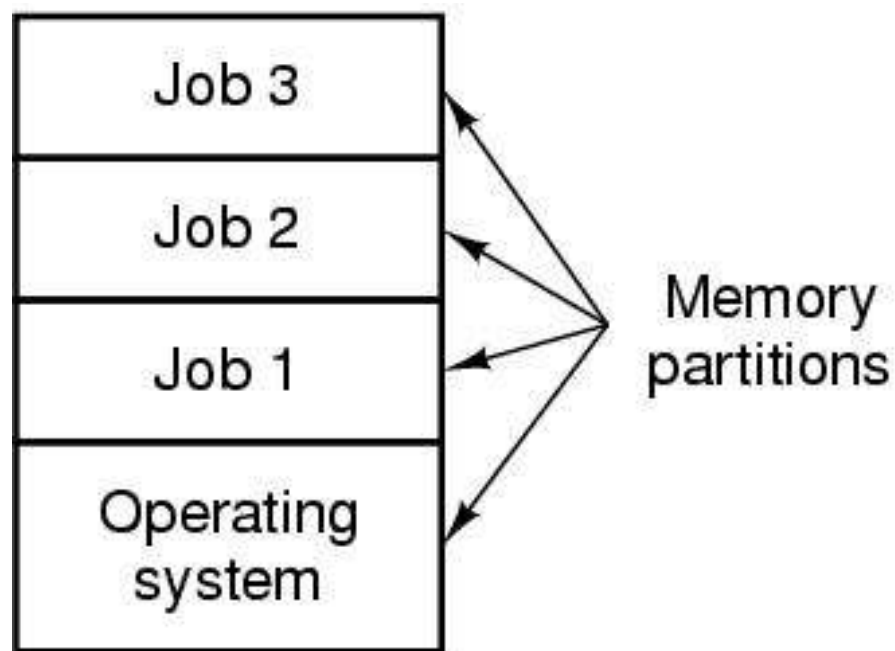


Figure 1-5. A multiprogramming system with three jobs in memory.

第四代：个人计算机 1/2

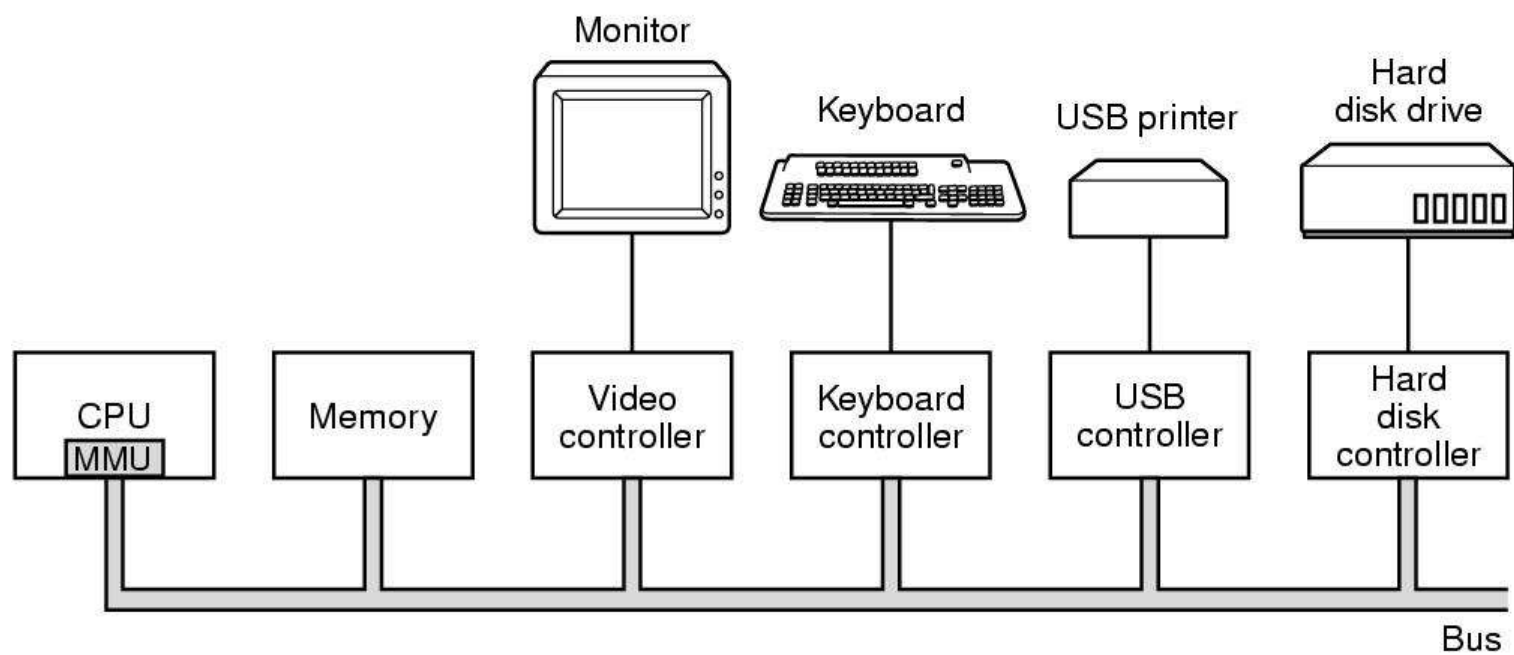


Figure 1-6. Some of the components of a simple personal computer.

第四代： 个人计算机 2/2

- Intel 8080, 8086, 8088, 80286, 80386, 80486, Pentium, Core, ...
- Motorola 6502, 68000, ...
- ARM, ...
- IBM PC, Apple II, smart phone, ...
- CP/M, DOS, Windows, Unix/Linux, Mac OS, iOS, Android, ...

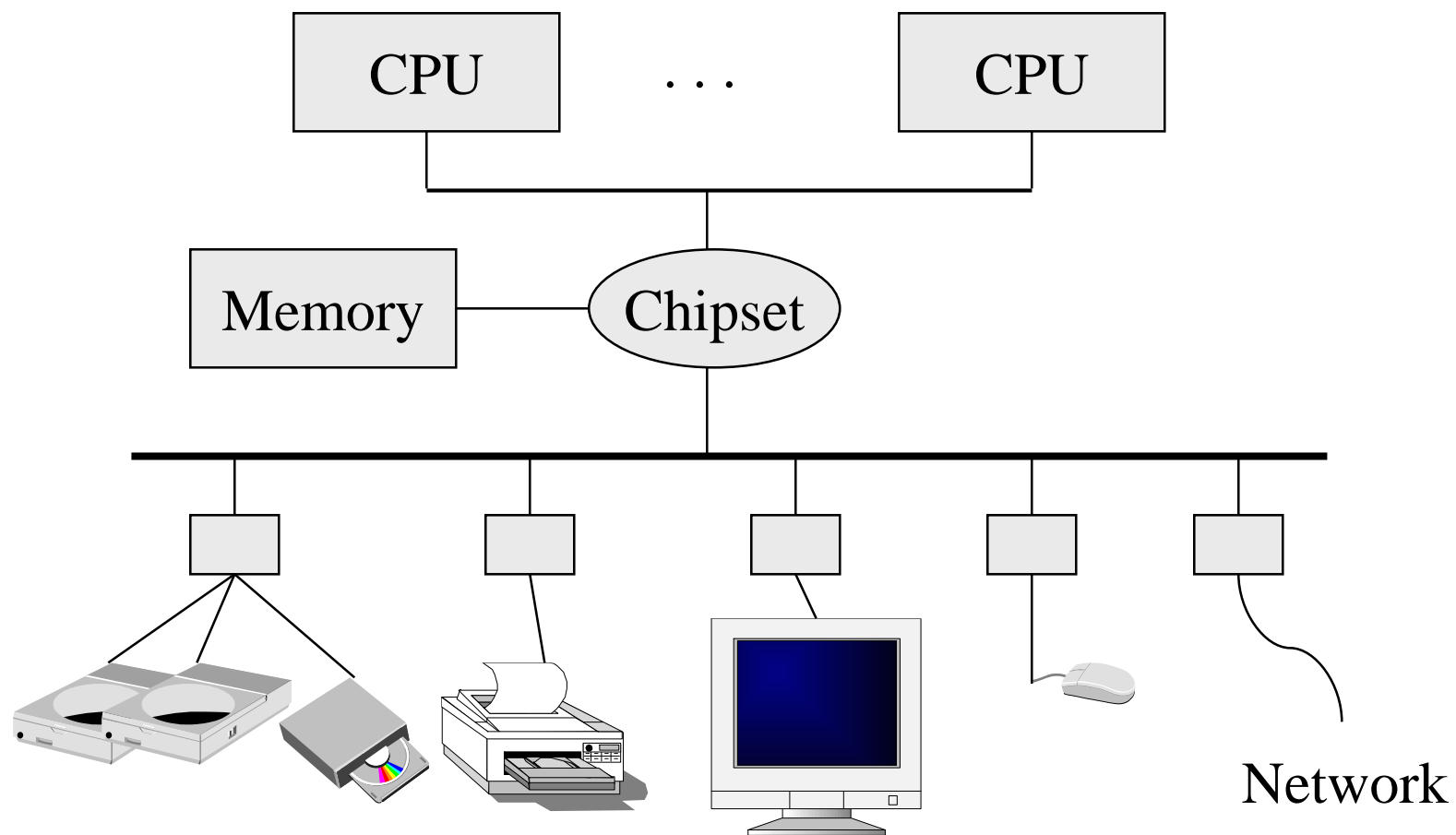
历史比较

	大型机	小型机	微机/移动计算机
系统价格/工作者年薪	10:1 – 100:1	10:1 – 1:1	1:10-1:100
主要考虑	系统利用率	总体造价	生产率
设计目标	系统容量	系统功能	易用性

第一课： 系统结构预基础

- 计算机发展简史
- 计算机硬件概述
- 系统结构与操作系统

一个典型的计算机系统



冯·诺依曼体系结构

- 冯·诺依曼体系结构是什么？

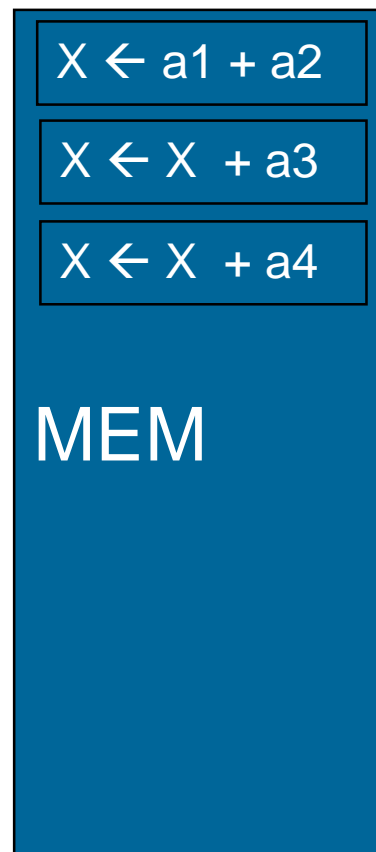


EDSAC, May 1949

$$X \leftarrow \sum_{i=1}^N a_i$$

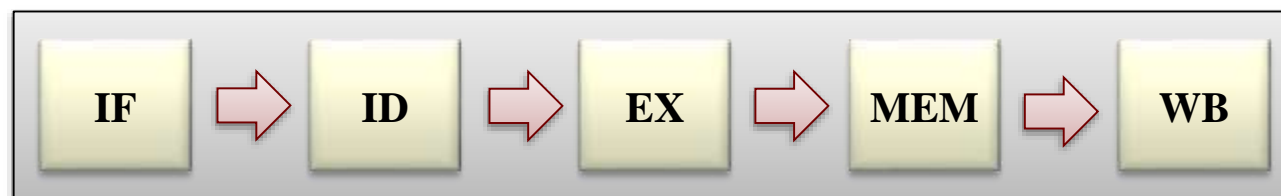
重要概念

- 指令
- Program counter



冯·诺依曼体系结构处理器的内部

- 指令集体系结构 (ISA)
- 通用寄存器
- 程序计数器 (PC)
- 程序状态字 (PSW)
- 流水线级



存储层次

为何要这么复杂的层次结构？

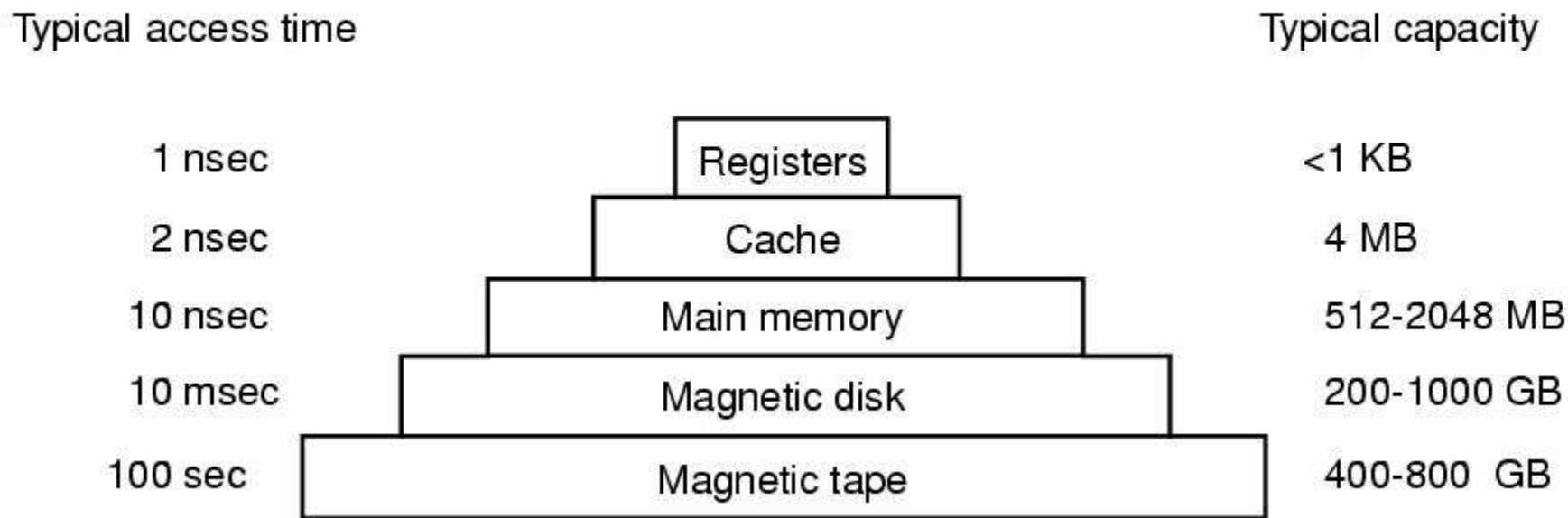


Figure 1-9. A typical memory hierarchy. The numbers are very rough approximations.

多核芯片

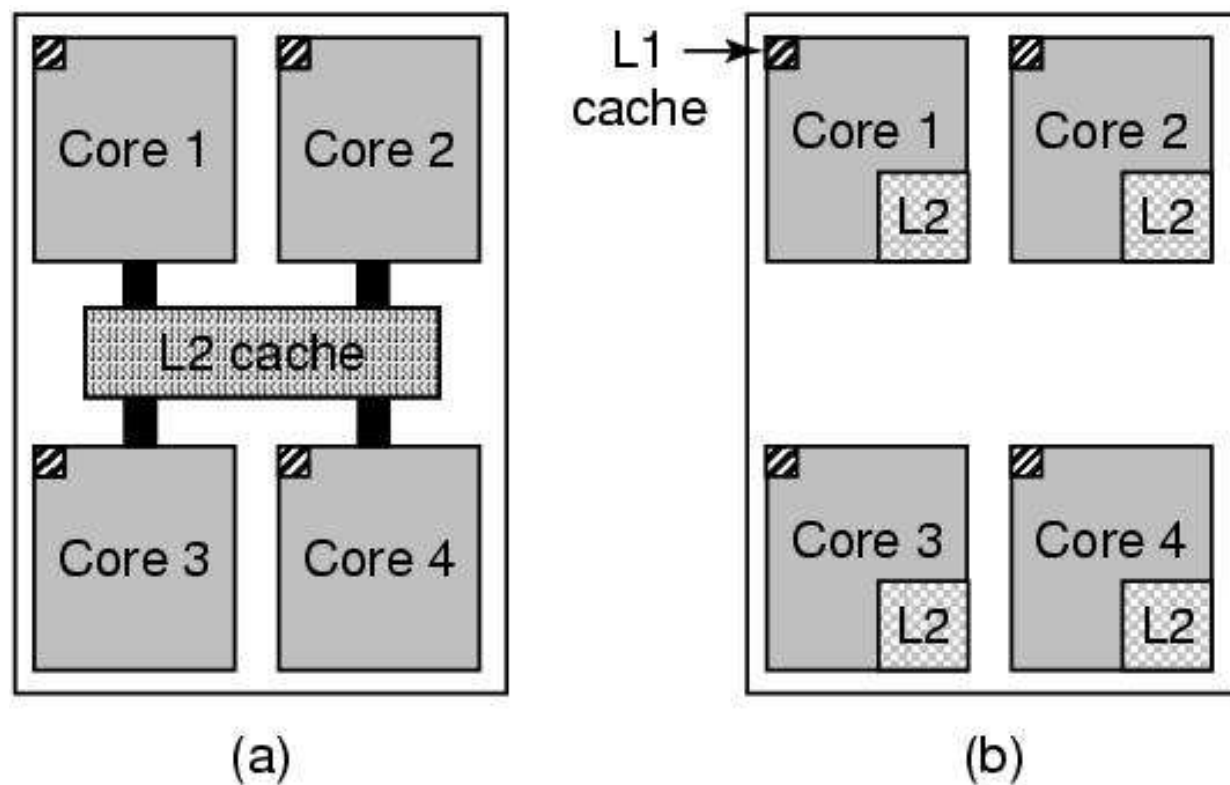


Figure 1-8. (a) A quad-core chip with a shared L2 cache.
(b) A quad-core chip with separate L2 caches.

磁盘

- 柱面, 磁头, 扇区 (CHS)
- 磁道, ...

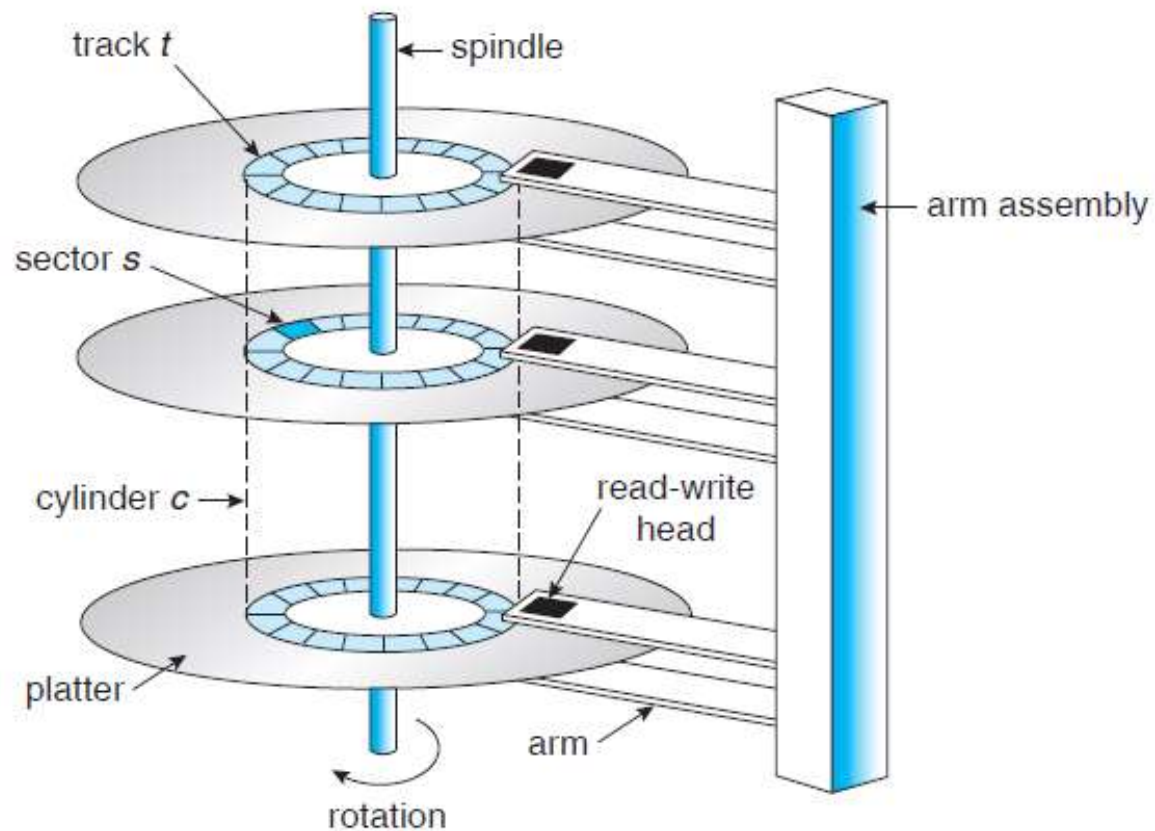


Figure 11.1 Moving-head disk mechanism.

输入输出设备

- 寄存器
 - 状态
 - 控制
 - 数据输入
 - 数据输出
- 轮询 v.s. 中断
- 可编程I/O (PIO) v.s. 直接内存访问 (DMA)

总线

- 为什么会有不同的总线?
- 当前有哪些流行的总线?

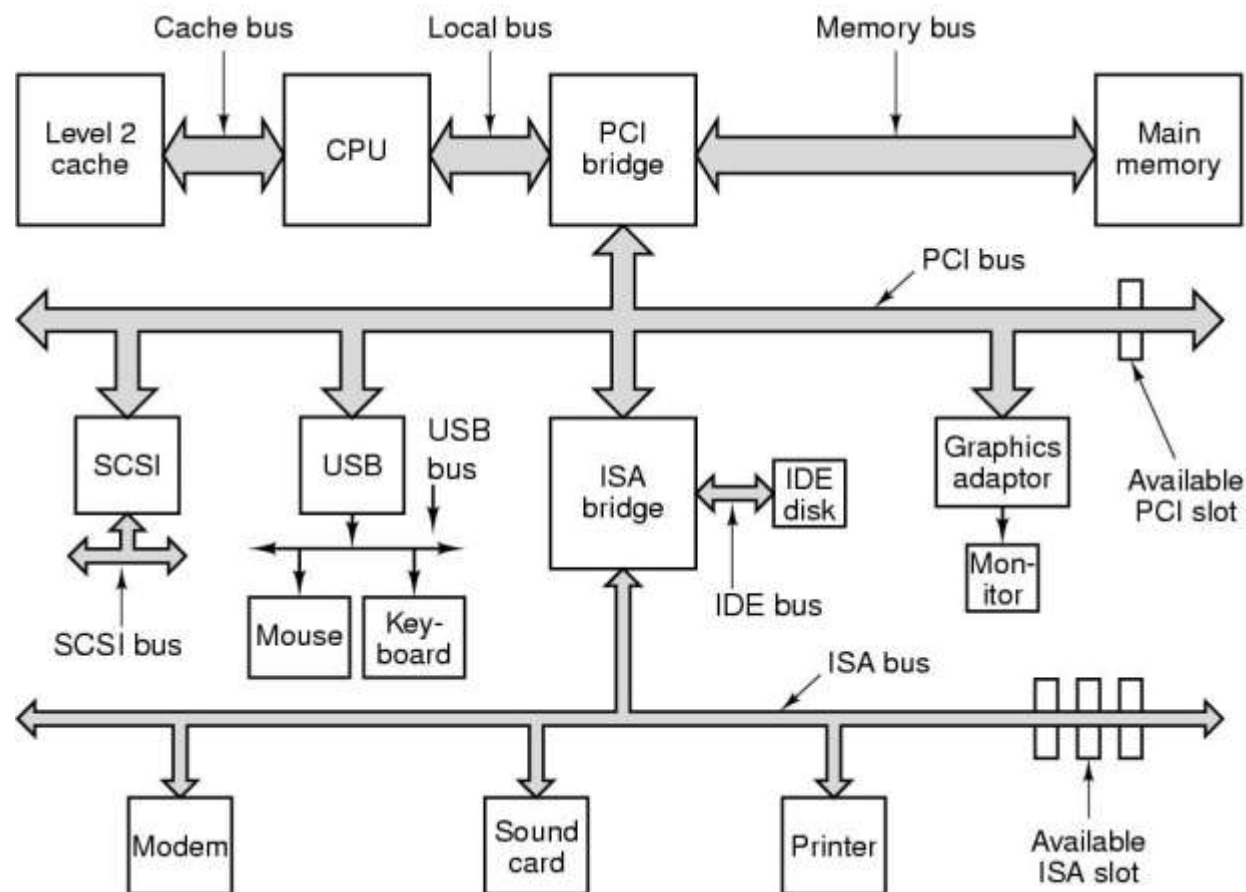
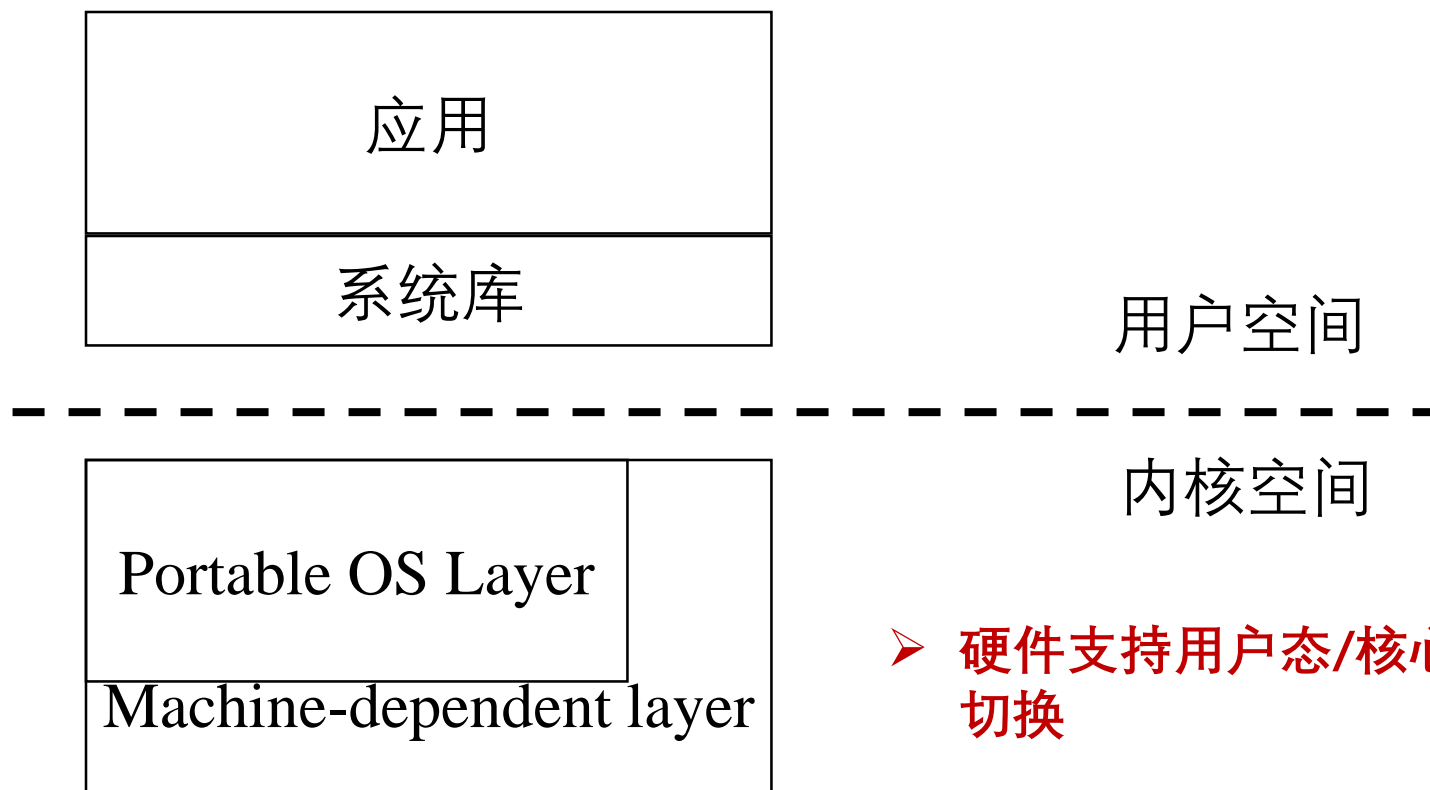


Figure 1-12. The structure of a large Pentium system

第一课： 系统结构预基础

- 计算机发展简史
- 计算机硬件概述
- 系统结构与操作系统

Unix/Linux操作系统一览



➤ 硬件支持用户态/核心态的切换

操作系统所需要的硬件支持

- 特权指令
- 操作系统保护（核心/用户模式）
- 内存保护
- 事件：异常与中断
- 时钟
- 输入输出控制
- ...

特权指令

- 任何一个CPU都拥有的指令子集，只能被操作系统调用
- 只有操作系统可以：
 - 直接访问I/O设备（磁盘、打印机等）
 - 安全性、公平性
 - 进行内存管理
 - 页表指针、页面保护、TLB管理
 - 对特权控制寄存器进行操作
 - 进入核心态、中断级别
 - Halt指令（为什么？）

操作系统保护

- 如何能够知道何时可以执行特权指令？
 - 一个简单的办法：只有运行操作系统代码的时候才可以
- 但是怎么知道正在运行操作系统代码？
 - 体系结构至少要支持两种模式：核心态和用户态
 - 当前所处的状态，由一个受保护的寄存器位来表示
 - 用户程序运行在用户态
 - 操作系统运行在核心态
- 特权指令只能在核心态执行
 - 当特权指令待运行时，处理器检查模式位
 - “普通指令”不能对模式位进行修改
- 问题：如何从用户态到核心态？

内存保护

- 为什么需要?
 - 操作系统必须能够防止程序相互影响
 - 操作系统本身也必须防止外界程序的侵入
- 内存管理硬件（MMU）提供内存保护机制
- 对MMU进行操作，需要使用特权指令

事件

- 事件，是指在控制流中“非自然”的改变
- 两种不同的事件：中断和异常
- 异常是在执行指令过程中引发的
- 中断是由外界事件引发的
 - 例如，设备完成了I/O操作等
- 异常和中断都可以有两种产生形式
 - 非预期产生
 - 故意产生

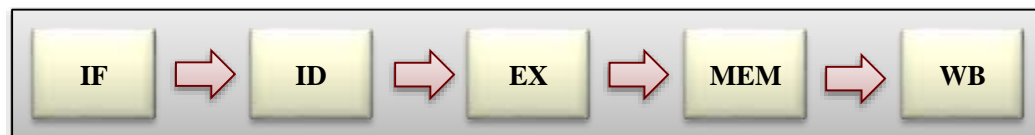
	非预期	故意
异常	错误	系统调用
中断	中断	软中断

错误

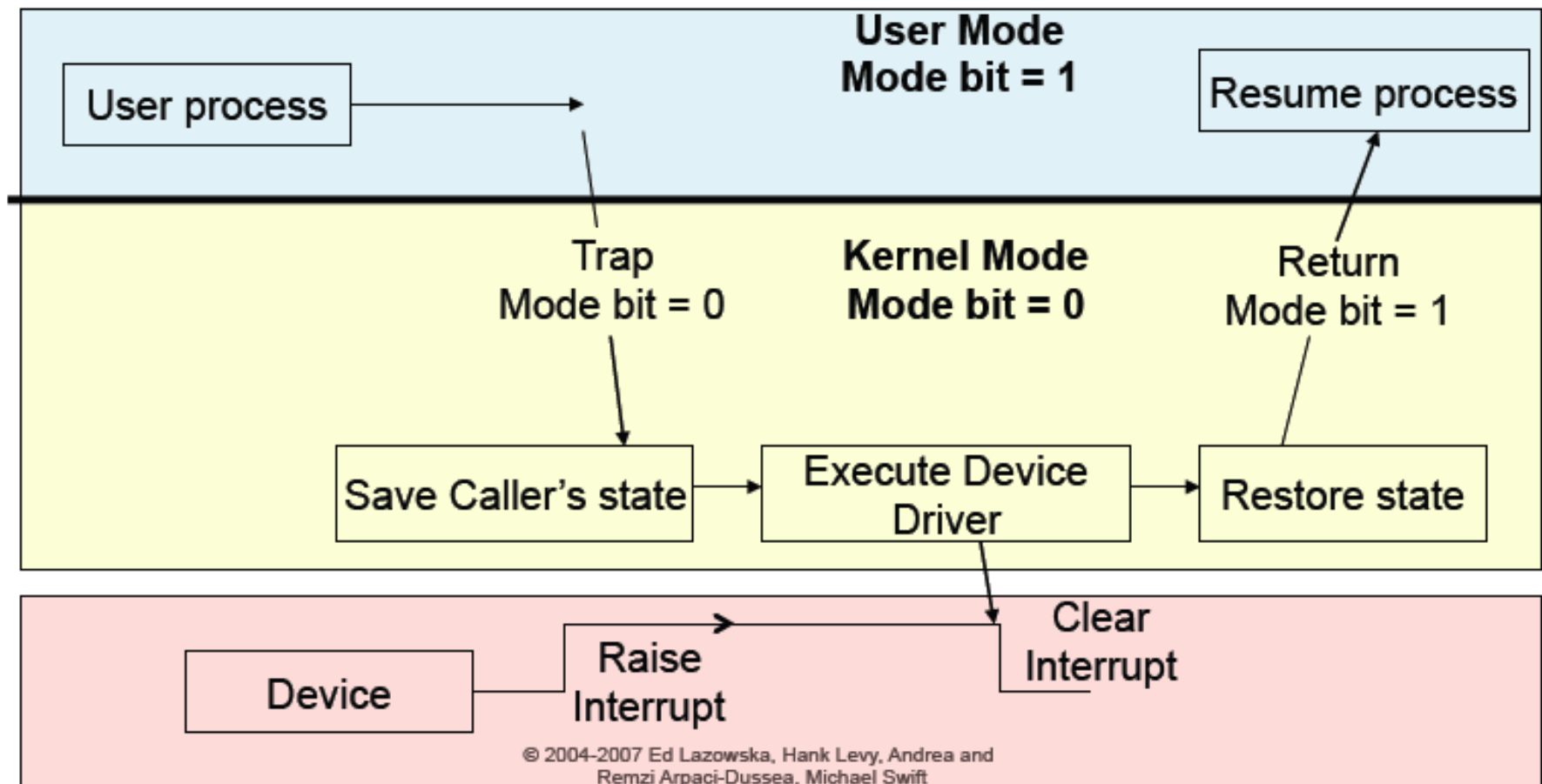
- 硬件检测出错误，并报告异常状态
 - 例如：页面错误、非对齐访问、除零
- 发生异常时，硬件“出错”
 - 必须保存系统状态（PC、寄存器、模式等），以便于出错的进程恢复执行
 - 操作系统会进行处理
 - 可恢复错误
 - 不可恢复错误

中断

- 中断引发异步事件
 - I/O硬件中断
 - 软件和硬件时钟
- 两种不同的中断
 - 精确中断：CPU只在指令之间来处理中断（之前指令执行完毕、下一条指令无效）
 - 非精确中断：CPU立刻开始中断（之前指令未必能执行完毕）
 - 操作系统设计者喜欢精确中断，CPU设计者喜欢非精确中断



中断过程示意



时钟

- 时钟对于操作系统至关重要
- 通过时钟，操作系统能够定时重新控制系统
 - 时钟会被设置成每隔一段时间就发一次中断
 - 设置时钟要通过特权指令
- 时钟可以用来处理死循环程序
 - 当操作系统发现某些程序错误地过度浪费CPU资源时，可以强制收回控制权
- 还可以被用来实现一些与时间相关的系统调用（例如`sleep()`）

I/O控制

- I/O涉及的问题
 - I/O的发起
 - I/O的结束
- I/O的发起
 - 特殊指令
- I/O的结束
 - 同步I/O: 特殊指令
 - 异步I/O: 中断

异步I/O的结束

- 中断是异步I/O的基础
 - 操作系统发起I/O
 - 设备与系统其它部分异步（并行）工作
 - 设备完成后，向处理器发出一个中断信号
 - 操作系统维护一个中断向量表，记录了处理不同事件的不同处理例程入口地址
 - CPU根据中断号来查找入口地址，切换到处理例程

I/O示例

1. 以太网控制器接到一个网络包，将它写入内存
2. 以太网控制器发出一个中断
3. CPU中断当前的操作，切换到核心态，保存系统状态
4. CPU从中断向量表中找到中断处理例程的入口，跳转到此入口
(以太网控制器驱动程序)
5. 以太网控制器驱动程序处理网络包
6. 工作完成后，将系统恢复到保存的系统状态

总结

- 计算机发展简史
- 计算机硬件概述
- 系统结构与操作系统
- 下一课：系统结构的基本概念