

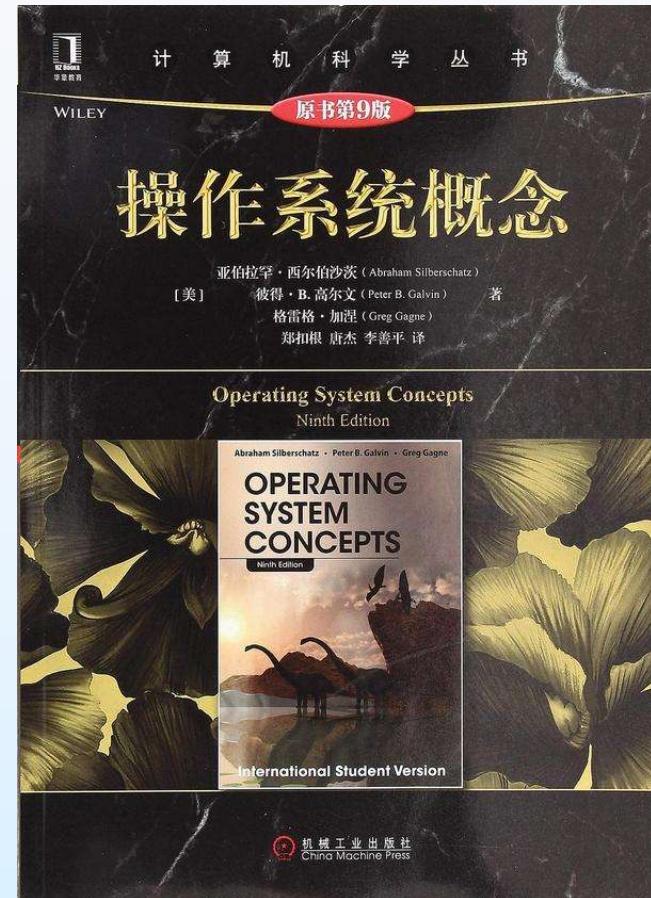
# 操作系统





# 一、课程情况

- 课时： 34
- 学分： 2
- 课程性质： 学位课
- 教材： 《操作系统概念》 第9版
  - ▶ 机械工业出版社
  - ▶ Abraham Silberschats etc.





# 教学目标

- 掌握计算机操作系统的**基本原理及组成结构**
- 掌握计算操作系统的**基本概念**和相关的新概念、名词及术语
- 理解操作系统各**组成部分之间的关系**、**总体结构**和**综合工作原理**
- 了解计算机操作系统的**发展特点**、**设计技巧**和**方法**
- 培养学生分析、修改和设计操作系统的能力
- 对常用计算机操作系统会进行**基本的操作使用**





# 先修课程

- 汇编语言程序设计
- C语言或Java
- 微型计算机技术
- 数据结构
- 面向对象及可视化程序设计



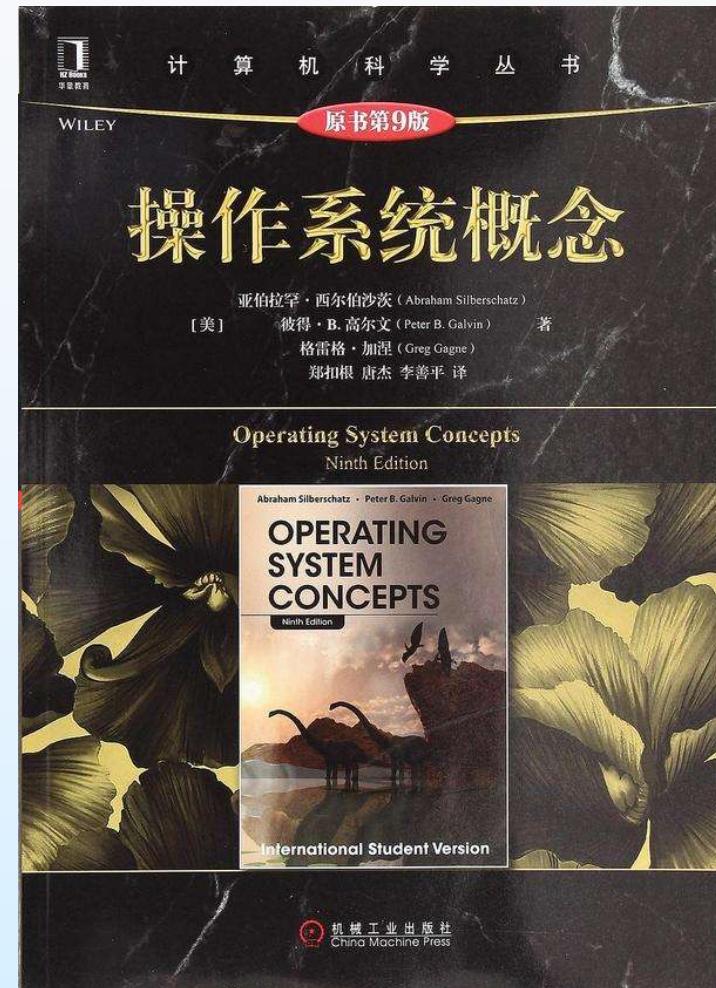
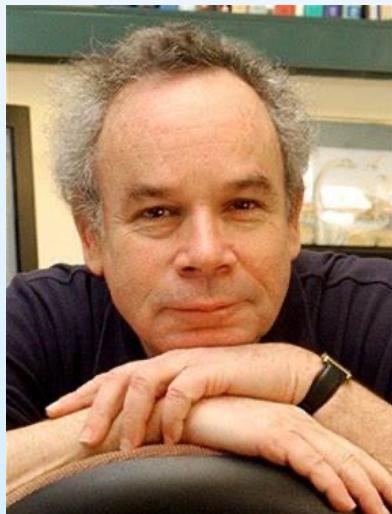
# 教科书

## ■ 操作系统概念

- 美国大学常用教材
- 操作系统经典教材

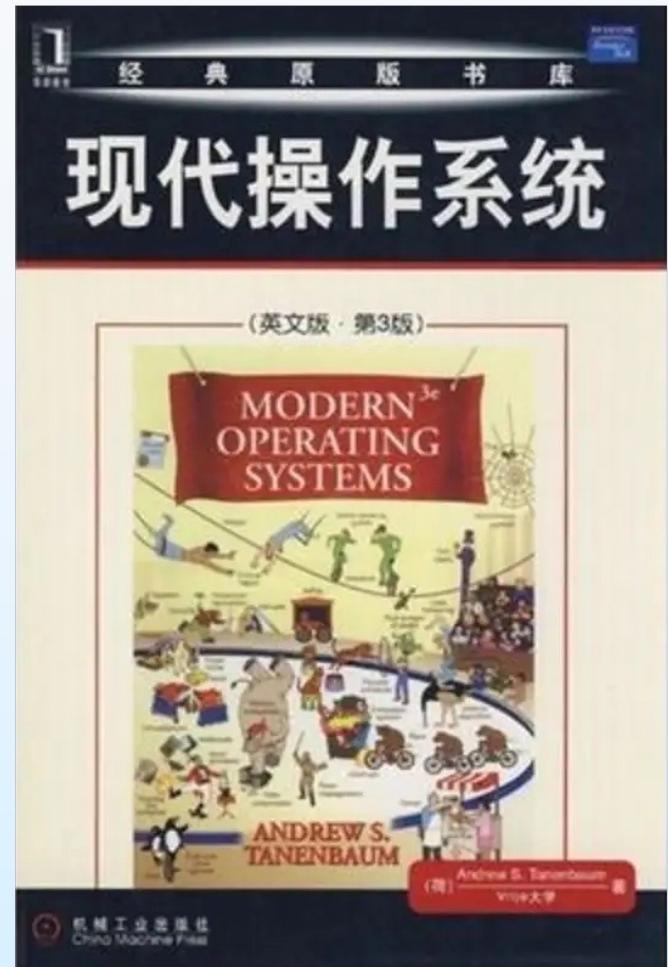
## ■ Abraham Silberschatz

- 耶鲁大学教授



## 二、参考书目

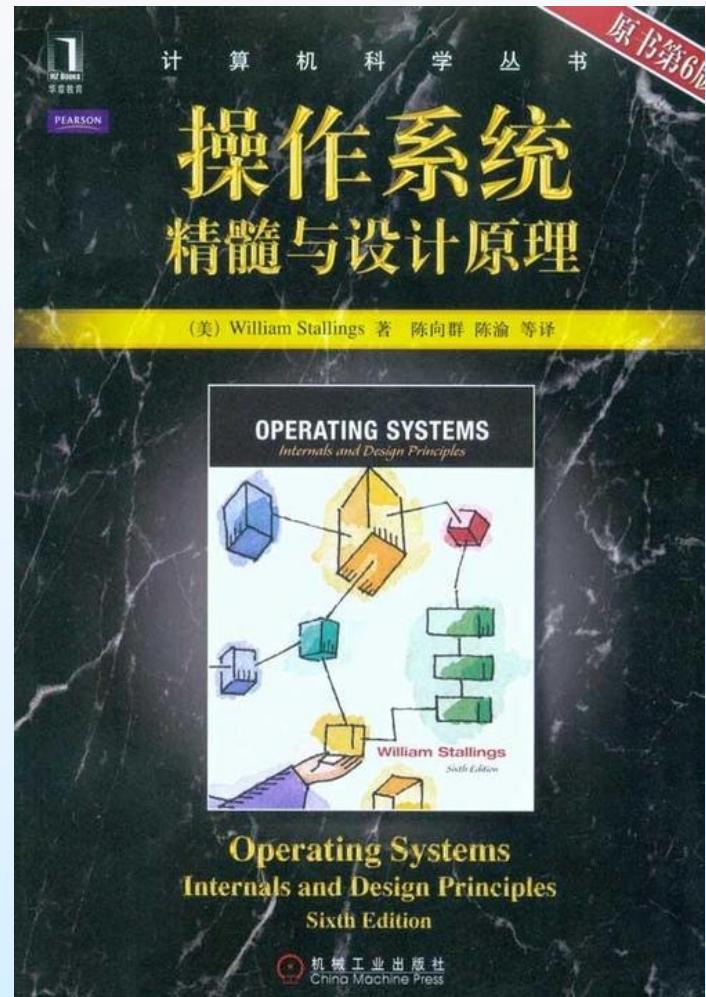
- 现代操作系统 Andrew S. Tanenbaum  
机械工业出版社
- Tanenbaum:
  - 三个操作系统的设计师或联合设计师
  - 长期设计开发操作系统的经验
  - 《现代操作系统（英文版第3版）》  
-操作系统领域的经典之作





## 二、参考书目

- 操作系统精髓与设计原理  
William Stallings 机械工业出版社
- 以当代最流行的操作系统——windows、Unix和Linux为例，全面清楚地展现了当代操作系统的本质和特点
- William Stallings: 世界知名的计算机学者和畅销教材作者，已经撰写了17部著作，出版了40多本书籍





### 三、课程内容

- Part one Overview (导论)
- Part two Process Management (进程管理)
- Part three Memory Management (内存管理)
- Part Four Storage Management (存储管理)

# 第一章 导论

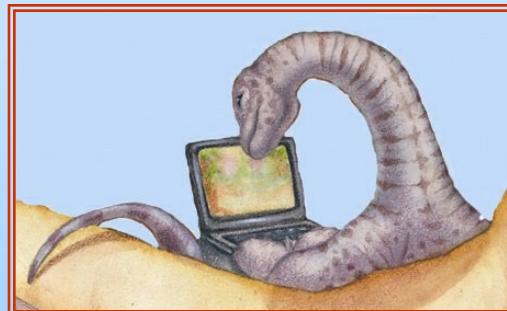




# 内容

- 1、什么是操作系统？
- 2、多道程序设计和分时技术
- 3、操作系统类型
- 4、操作系统操作和功能

# 1、什么是操作系统？



操作系统的  
目的  
操作系统定义  
计算机硬件结构  
操作系统启动

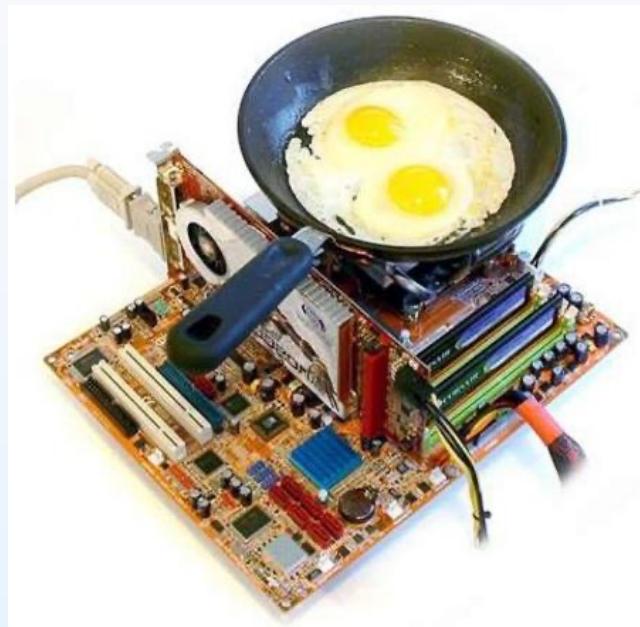


- 操作系统，Operating System，简称OS。
- 如果没有操作系统，计算机能做什么？
  - 我们将无法在计算机上运行任何程序



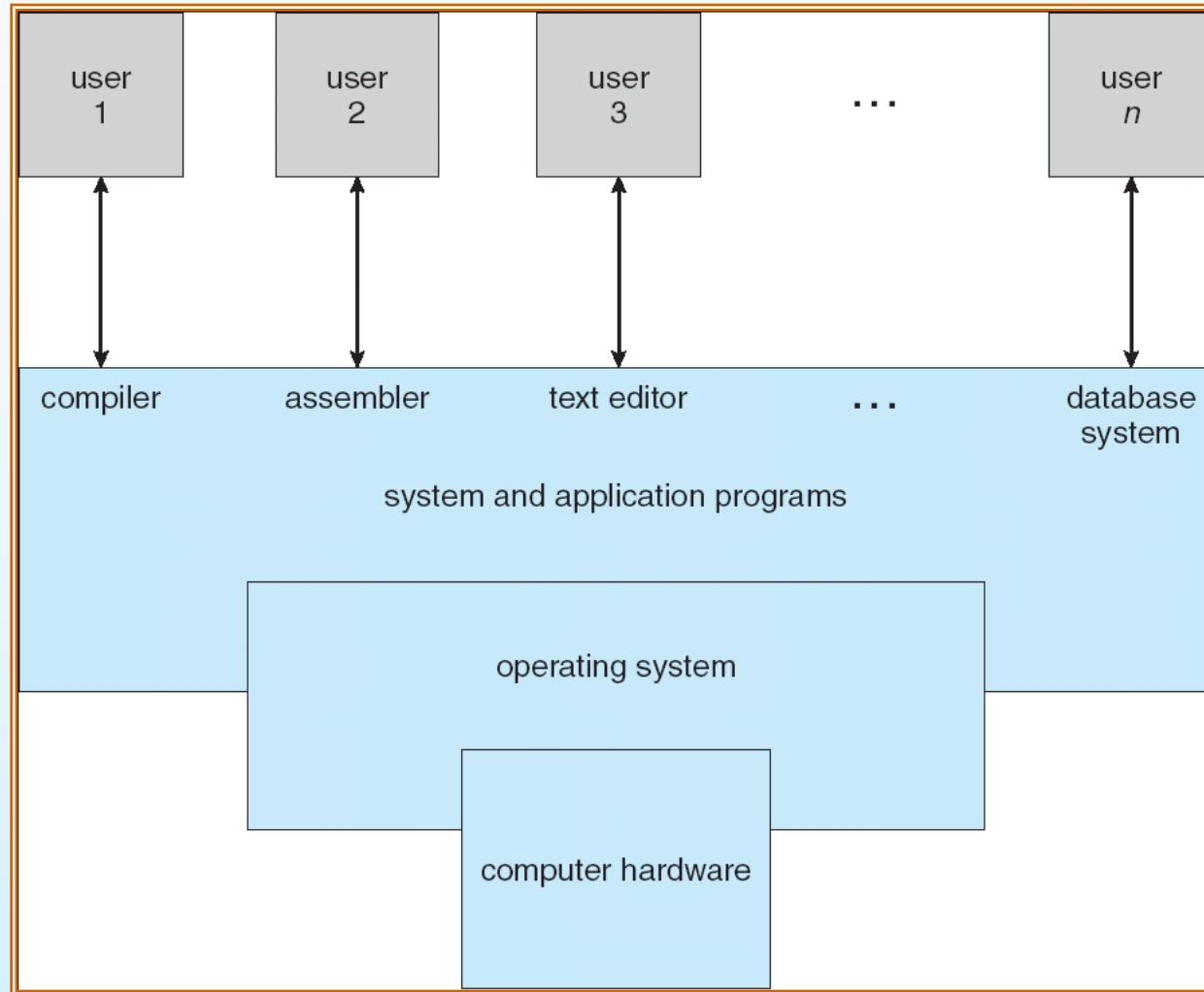


# 如果没有操作系统.....





# 计算机系统的4个部分



操作系统作为最底层的软件，是应用程序运行的基本支撑环境，不可或缺。





# 操作系统设计目标

- 管理计算机硬件的程序，在计算机用户和计算机硬件之间充当中介。
- 操作系统设计目标：
  - 运行用户程序——核心目标
  - 更方便地解决用户问题，使计算机系统方便地使用——面向用户
  - 以一种高效方式使用计算机硬件——面向系统
- 不同用户、不同系统、不同阶段有不同的侧重点
  - 早期：高效
  - 目前：方便





# 用户视角

- 不同的用户，不同的视角
- PC用户希望操作系统
  - 方便
  - 易用
  - 高性能
  - 不关心资源利用率
- 主机用户：满足所有用户-资源利用率
- 工作站用户：性能和资源利用率折中
- 手持设备用户：方便、电池续航时间
- 有些计算机无用户界面-家电的电脑控制器





# 系统视角

## ■ OS 是资源分配器

- 管理所有资源
- 面对冲突的资源请求，决定如何分配资源，以便系统能有效而公平地运行

## ■ OS 是控制程序

- 管理用户程序的运行以防止计算机资源的错误使用或使用不当
- 由于在系统层面对于操作系统的要求基本一致，这就使得绝大多数的操作系统具有共性，都包括CPU管理、内存管理、文件管理、设备管理等。





# 操作系统定义

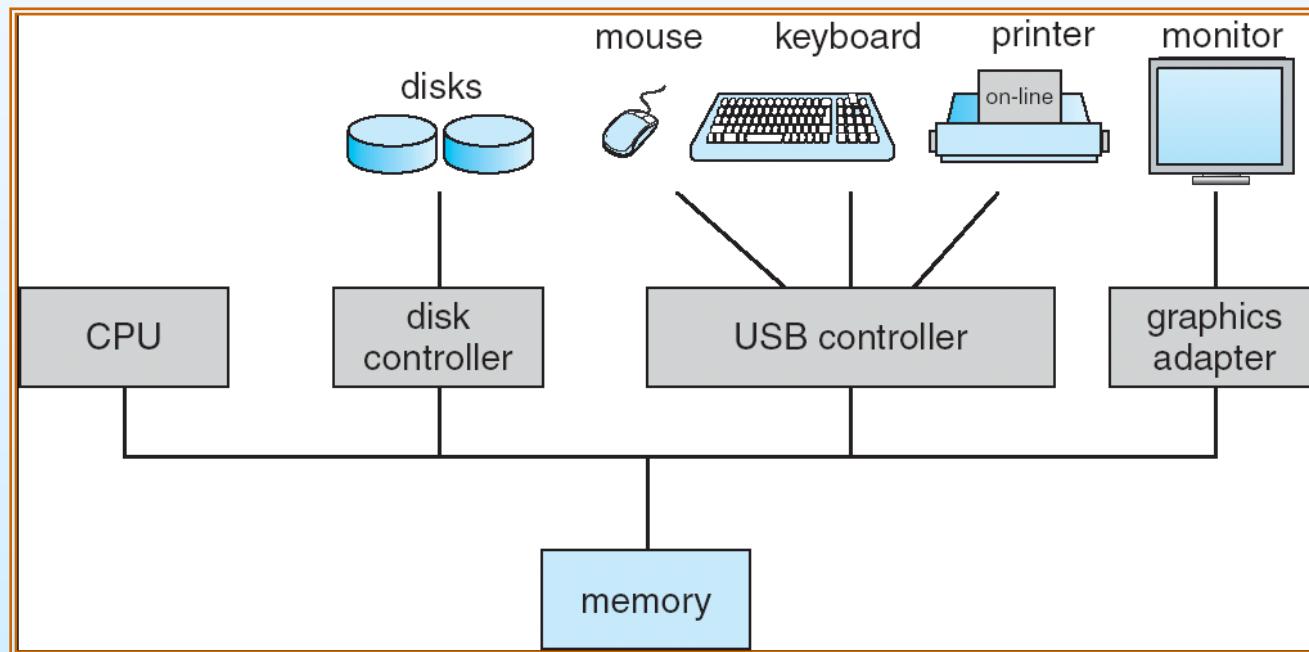
- 没有一个可广泛接受的操作系统定义
- 观点一：“当你预定一个操作系统时零售商所装的东西” 就是操作系统（简单观点）（极大化）
  - 包括的特性随系统不同而变化很大
  - 1998年微软和欧盟关于垄断的纠纷
- 内核才是操作系统。“一直运行在计算机上的程序” 称为内核 (**kernel**)。其他程序则为系统程序和应用程序。（极小化）





# 现代计算机系统

- 一个或多个 CPU 和内存
  - 若干通过总线相连的设备控制器及其设备
  - 总线
- 
- CPU 和设备控制器可 **并行** 工作，并竞争内存

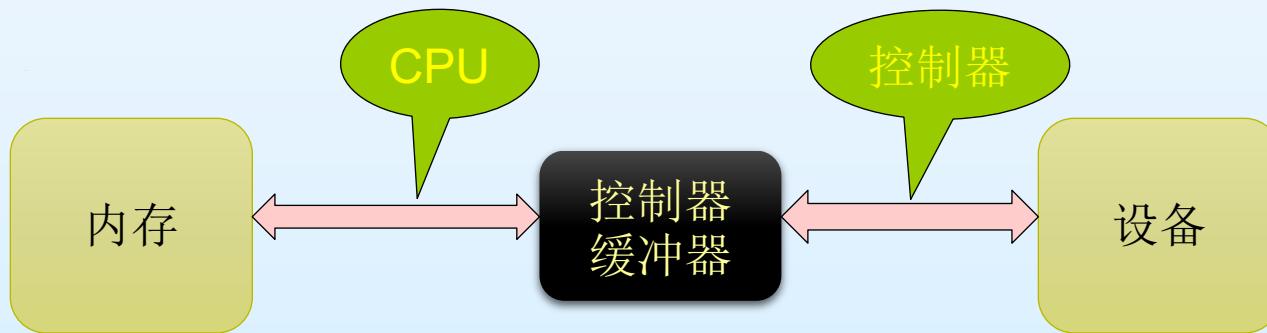


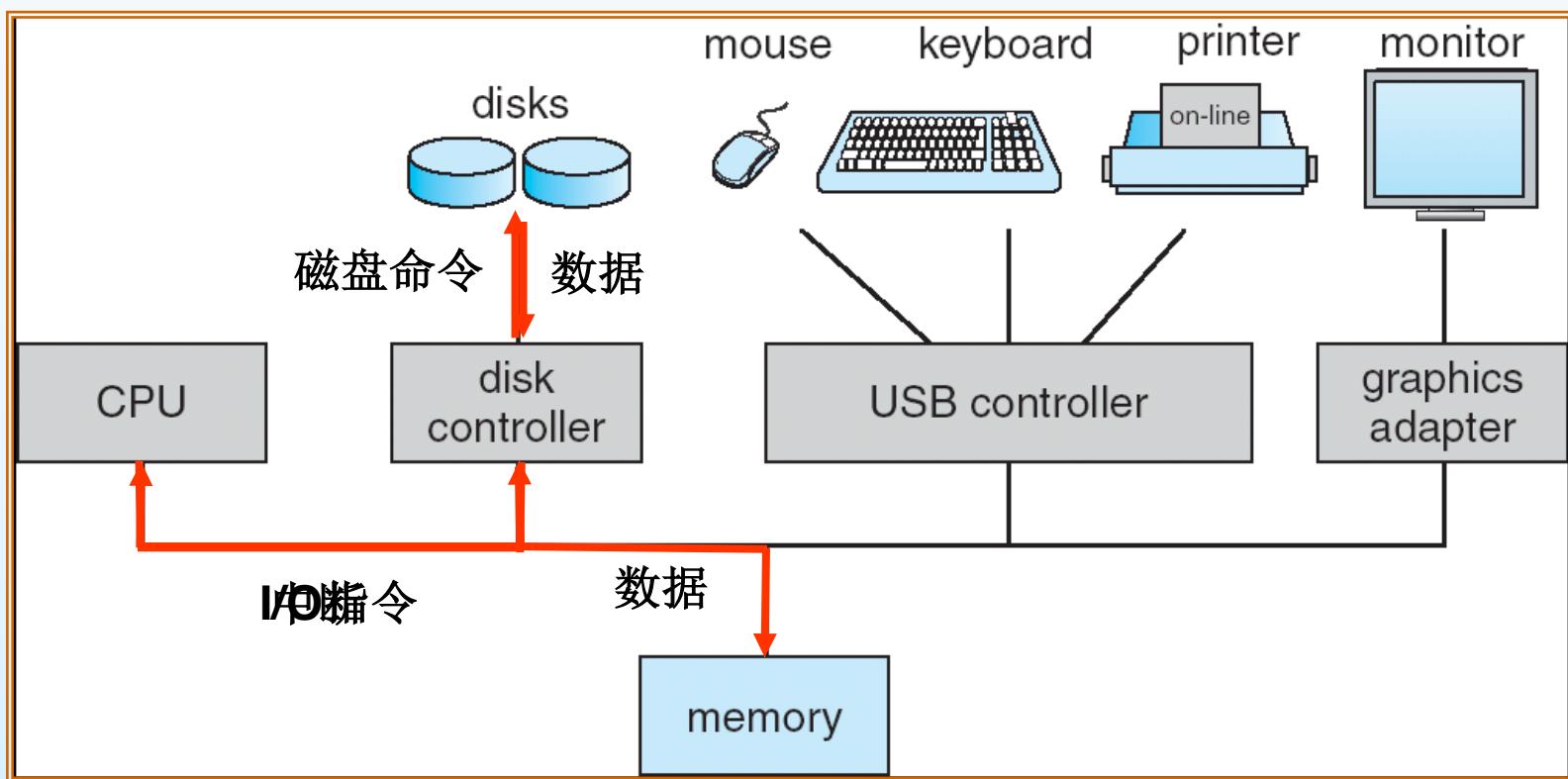


# 计算机系统操作

## ■ CPU通过设备控制器控制设备运行的过程

- 每个设备控制器有一个本地缓冲
- CPU 在**内存和本地缓冲**之间传输数据
- I/O控制器从**设备到本地缓冲**之间传输数据
- 协作：控制器通过调用中断通知**CPU**完成操作

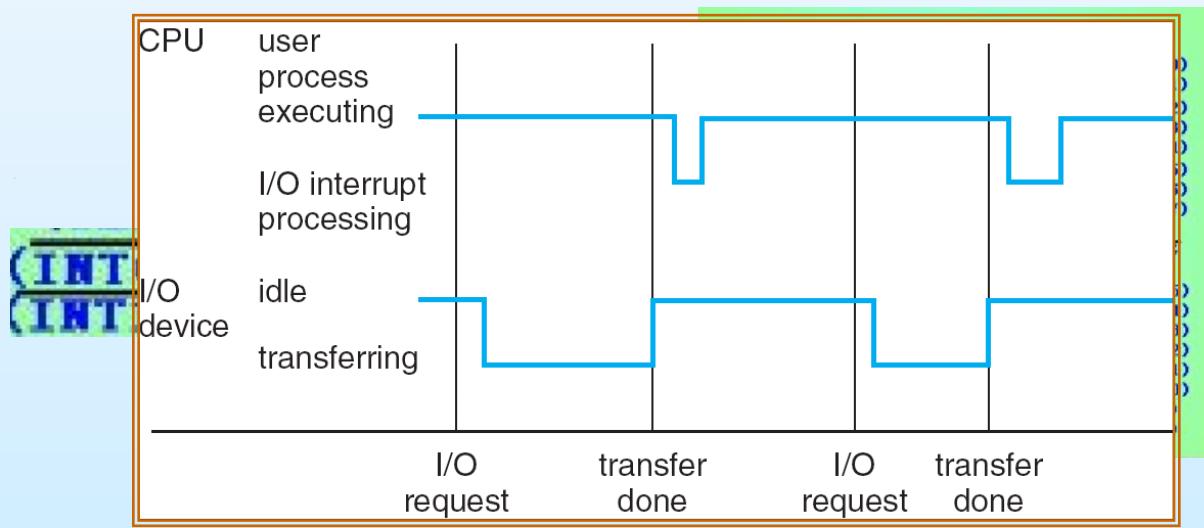






# 中断

- 中断：指当出现需要时，CPU暂时停止当前程序的执行，转而执行处理新情况的中断处理程序。当执行完该中断处理程序后，则重新从刚才停下的位置继续当前进程的运行。
- 中断号：外部设备进行I/O操作时产生的中断信号，发送给CPU
- 中断向量：中断服务程序的入口地址
- 中断服务程序：执行中断处理的代码
- 操作系统是中断驱动





# 系统启动





# 系统启动

## ■ 引导程序 (Bootstrap program)

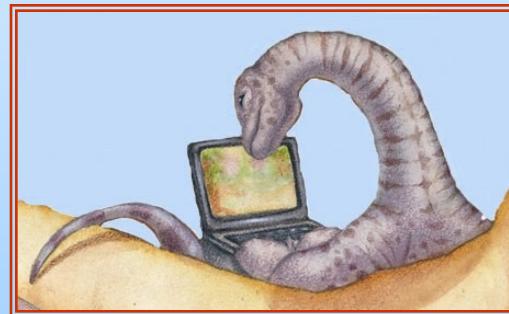
- 在打开电源或重启时被装载
- 通常位于 ROM 或EPROM中, 称为 **Firmware** (固件)
- 初始化所有硬件
- 负责装入操作系统内核并开始运行



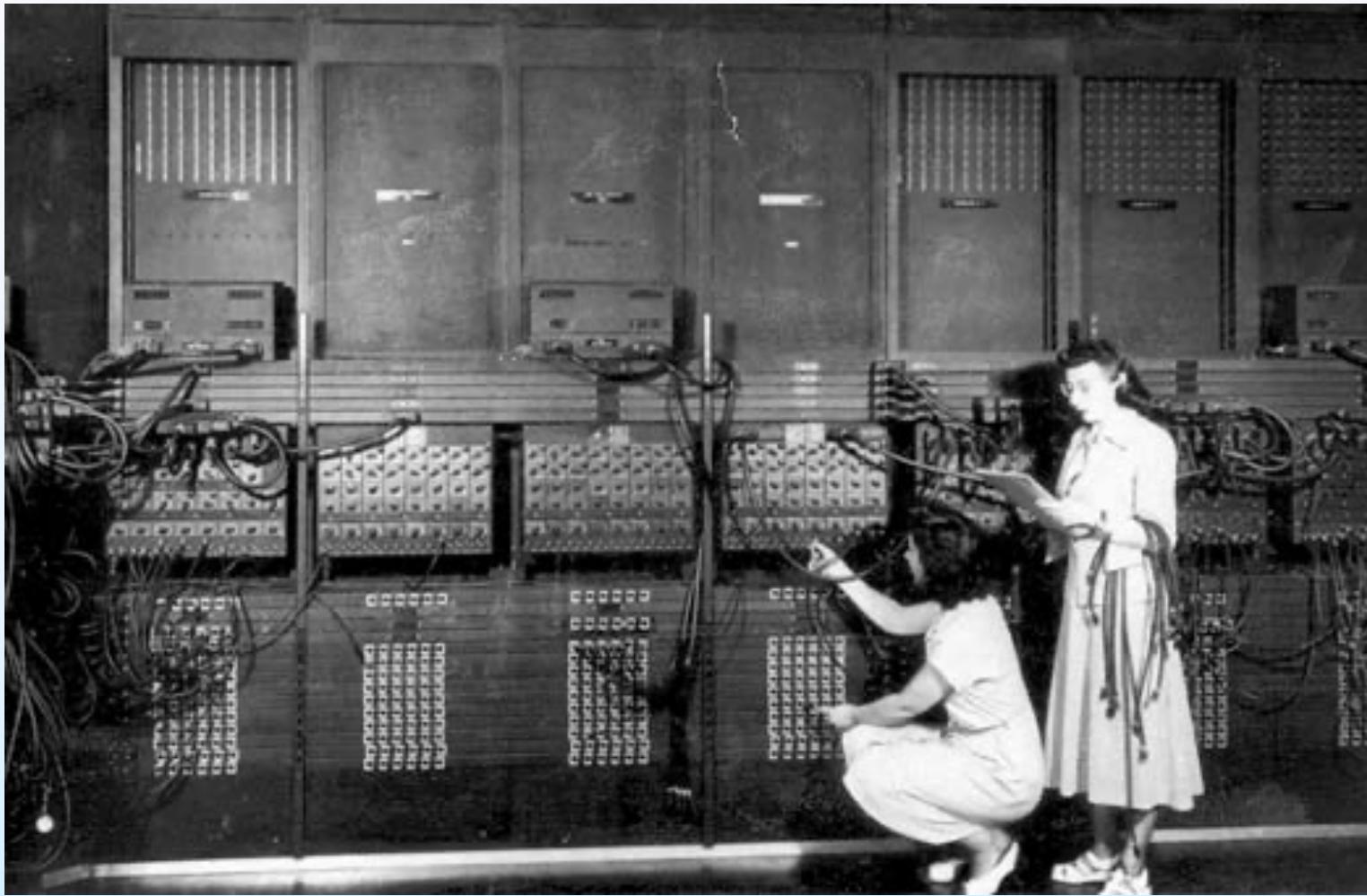
## 2、多道程序设计和分时技术



简单批处理系统  
多道程序系统  
分时系统



# 无操作系统计算机





# 简单批处理系统

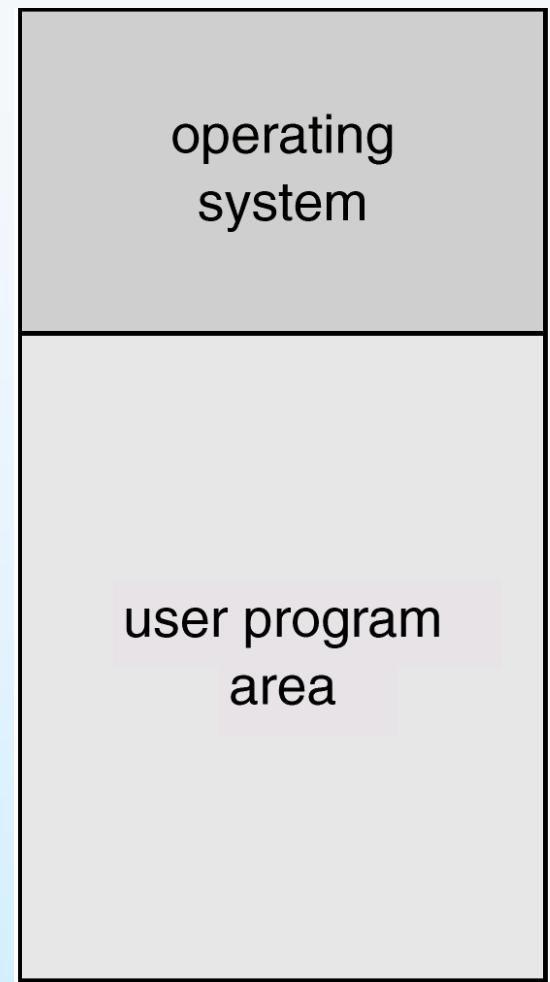
- 最早的操作系统，是操作系统的雏形
- 实现了自动化作业调度：自动从一个运行完的作业转换到运行下一个作业，在无需人工干预的条件下运行作业
- 程序的一次运行称为**作业**
- **批处理**：一批无需人机交互的作业批量运行
- 那么如何使得作业无需人工干预？需要操作系统的介入。
- 简单批处理系统的核心是一个**常驻监控程序（monitor）**
  - 控制作业传输
  - 调度作业运行
  - 自行从一个运行完的作业转换到下一个作业
- 单道程序运行

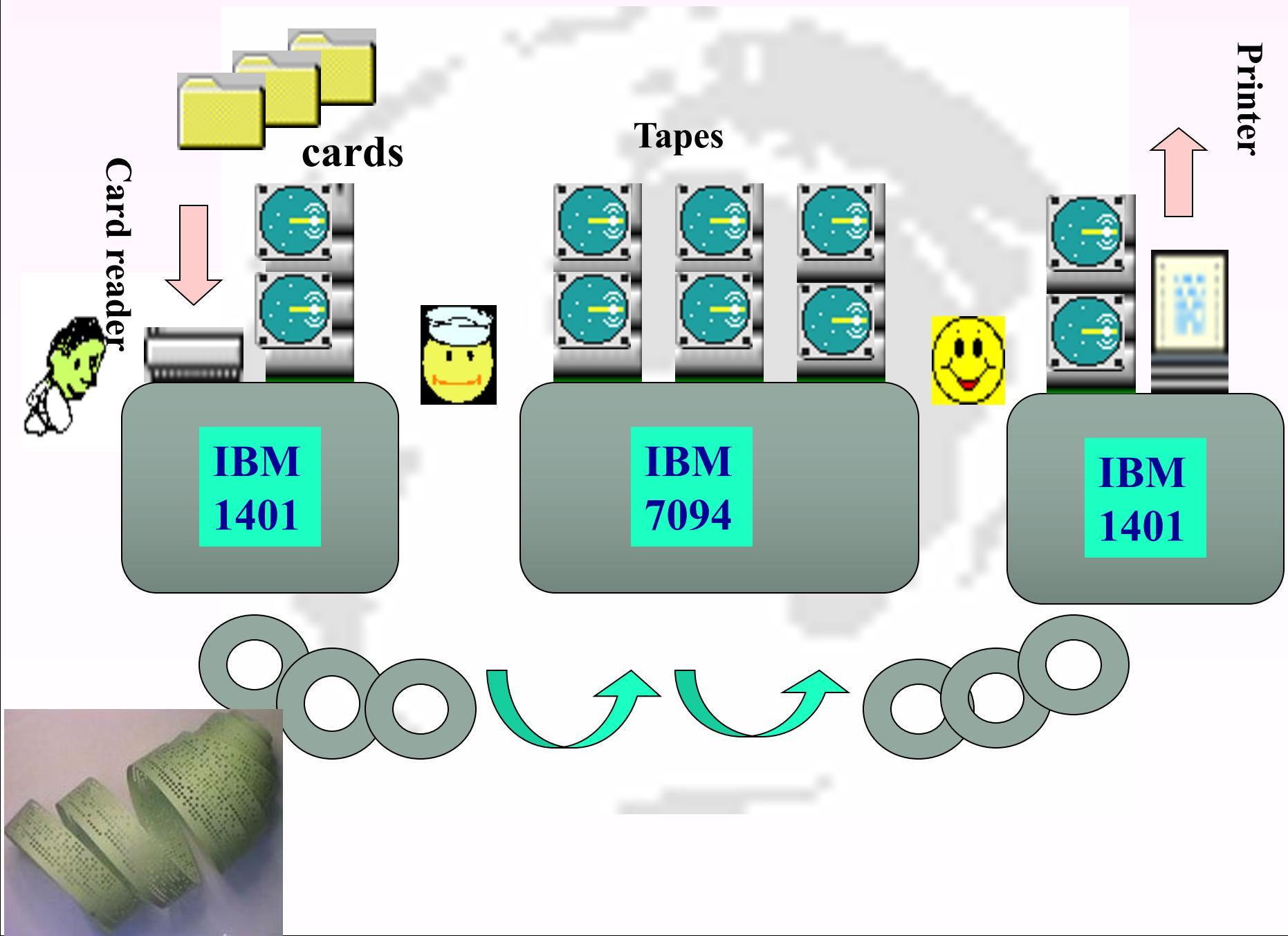




# 简单批处理内存映像

- 在用户程序区任何时候只能装入一道作业运行
- 当其运行完后，由Monitor装入下一道作业运行
- 所以简单批处理系统采用的也称为单道程序设计



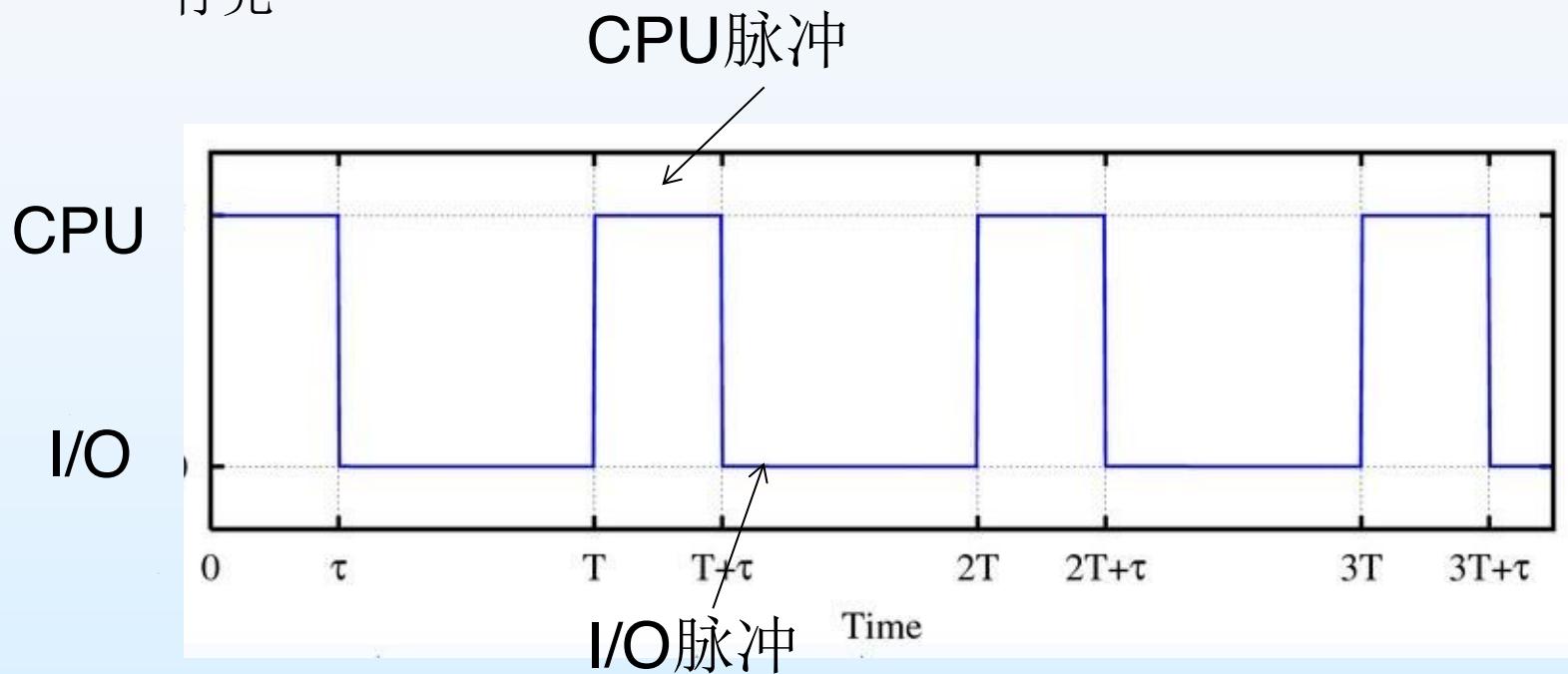




# 简单批处理系统

## ■ 作业运行过程

- CPU和I/O交替运行，经过无数次交替运行后作业才能运行完





- 下面有关简单批处理系统的描述，错误的是（ ）。
- A. 简单批处理系统的核心是一个常驻监控程序（Monitor），具有控制作业传输，调度作业运行，自动从一个运行完的作业转换到下一个作业等功能。
  - B. 简单批处理系统任何时候内存中只有一个作业。
  - C. 简单批处理系统考虑到了CPU和I/O设备的并行性。
  - D. 简单批处理系统采用单道程序设计技术。





# 多道程序系统

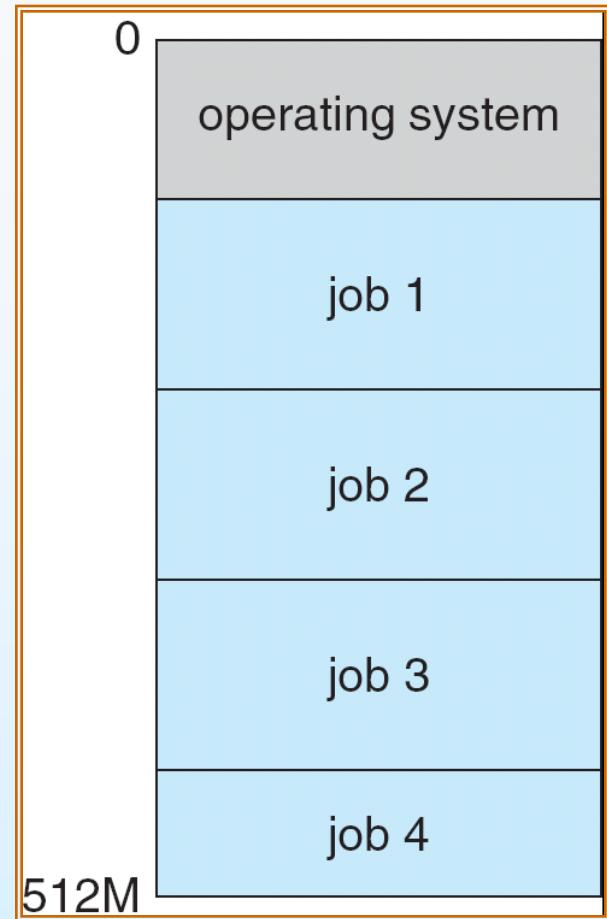
- 单个用户通常不能总是使得 CPU 和 I/O 设备在所有时间内都忙碌
- 多道程序设计：通过组织作业（代码或数据）使CPU总有一个作业可执行
- CPU 和 I/O 设备在管理程序控制下互相穿插运行，使得CPU和 I/O 操作可以并行
- 目的：使得CPU和设备在所有时间内尽可能忙碌，从而提高CPU和设备的利用率，充分发挥计算机系统各个部件的并行性





# 多道程序系统

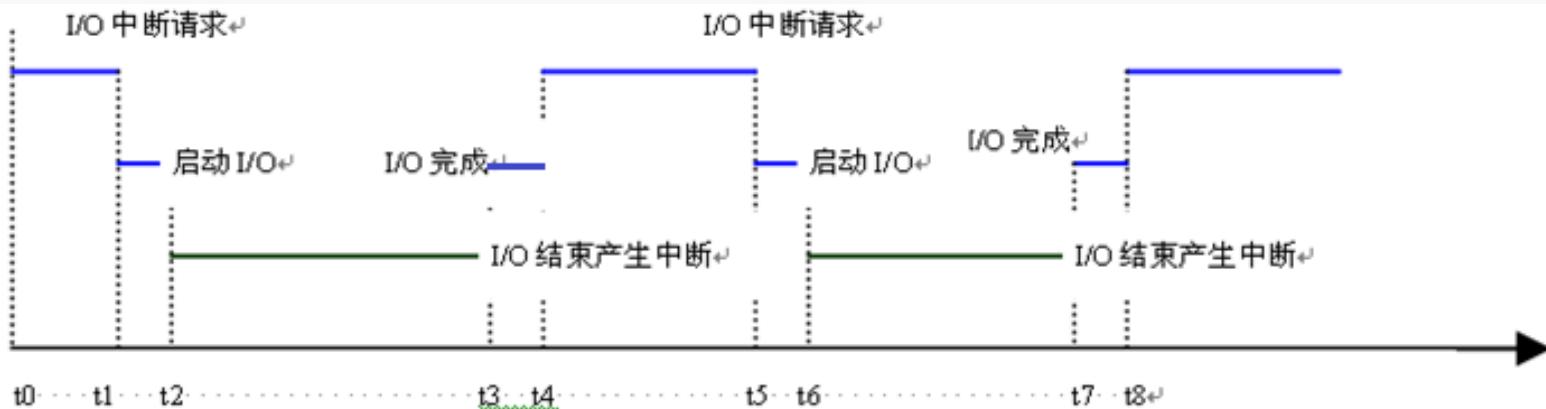
- 图中四个作业交替并发
- 需要重新设计操作系统的CPU管理、内存管理等模块
- 引入作业调度
  - ▶ 内存中保存系统中所有作业的子集
  - ▶ 通过作业调度(**job scheduling**)选中一个作业并运行
  - ▶ 当该作业必须等待时(如等待I/O), OS切换到另一个作业
- 多道程序设计是现代操作系统的基础,绝大多数操作系统都采用多道程序设计
- 由不同控制器控制的设备也可以并行操作
- 由相同控制器控制的设备一般不可以并行操作



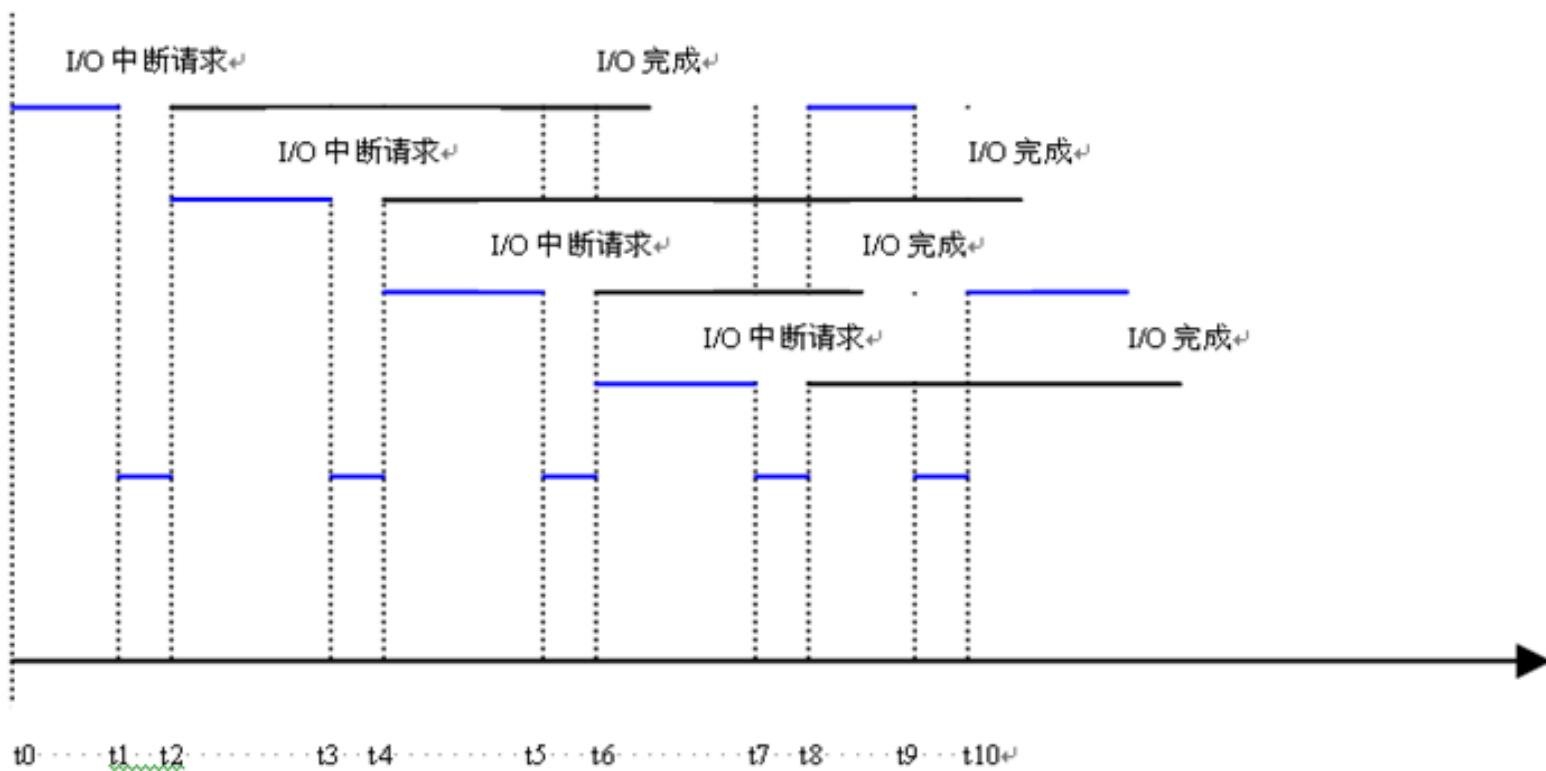


# 多道程序和单道程序运行例子

用户程序  
系统程序  
I/O 操作



用户程序 A  
用户程序 B  
用户程序 C  
用户程序 D  
系统程序





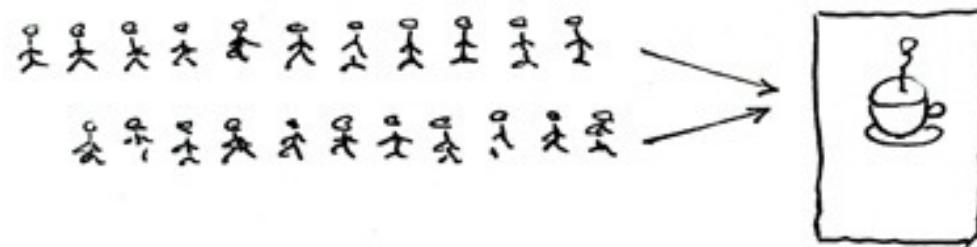
- 并行：两个或者多个作业在同一时刻运行
- 并发：两个或者多个作业在同一时间间隔内依次运行（几个作业在同一个CPU上运行，但是任意时刻点上只有一个作业在运行）
- 并发从宏观上类似于并行，从微观上类似于串行



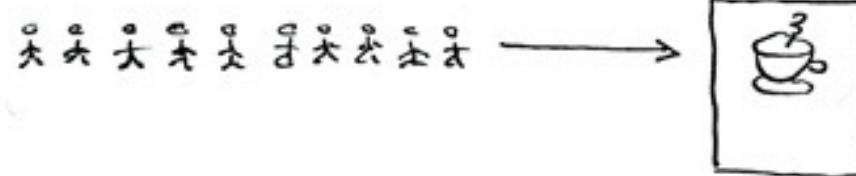
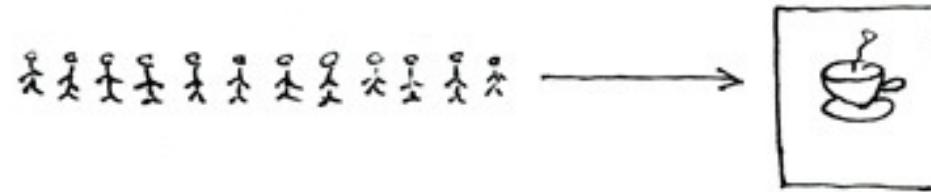


## ■ Erlang 之父 Joe Armstrong 用一张5岁小孩都能看懂的图解释了并发与并行的区别

Concurrent = Two Queues One Coffee Machine



Parallel = Two Queues Two Coffee Machines





■ 下面有关并行的描述，正确的是（ ）。

- A. 并行是指两个或多个作业在同一时间间隔内依次运行。  
也就是在一个时间段中，有几个作业在同一个**CPU**上运行，但任一个时刻点上只有一个作业在**CPU**上运行。
- B. 并行是指两个或者多个作业在同一时刻运行。
- C. 任意两个设备都能并行操作。
- D. 两个作业可以在一个**CPU**上并行运行。





# 分时系统

- 早期都是无需人机交互的批处理作业，随着主机系统的出现，出现了另外一种需要人机交互的作业，交互作业
- **交互作业：用户等待交互结果，用户希望响应时间  $< 1 \text{ second}$**
- 响应时间：从用户提交作业到作业第一次运行的时间，对用户而言是指到屏幕有显示的时间。
- 为了使得计算机适合交互作业，就有了分时技术，并应用到分时系统中
- 分时系统(或多任务)是多道程序设计的延伸





# 分时系统

- **时间片：**把一段**CPU**时间按照固定单位进行分割，每个分割得到的时间段称为一个**时间片**。
- **分时策略：**在一段时间内，**每个作业依次轮流运行一个时间片**，每个作业都获得较短的响应时间，从而满足交互作业的要求。
- 遵循分时策略的联机多用户交互式的操作系统就是**分时系统**。
- 分时系统
  - 一般采用**时间片轮转方式**使一台计算机为多个用户提供服务
  - 单位时间内，每个用户获得一个**时间片并运行**
  - 保证用户获得足够小的响应时间，并提供交互能力





# 第一个分时系统

## ■ CTSS(Compatible Time-Sharing System)

- 1962年开发
- 第一个分时系统
- MIT开发
- 运行在IBM 7094大型机
- 支持32个交互式用户

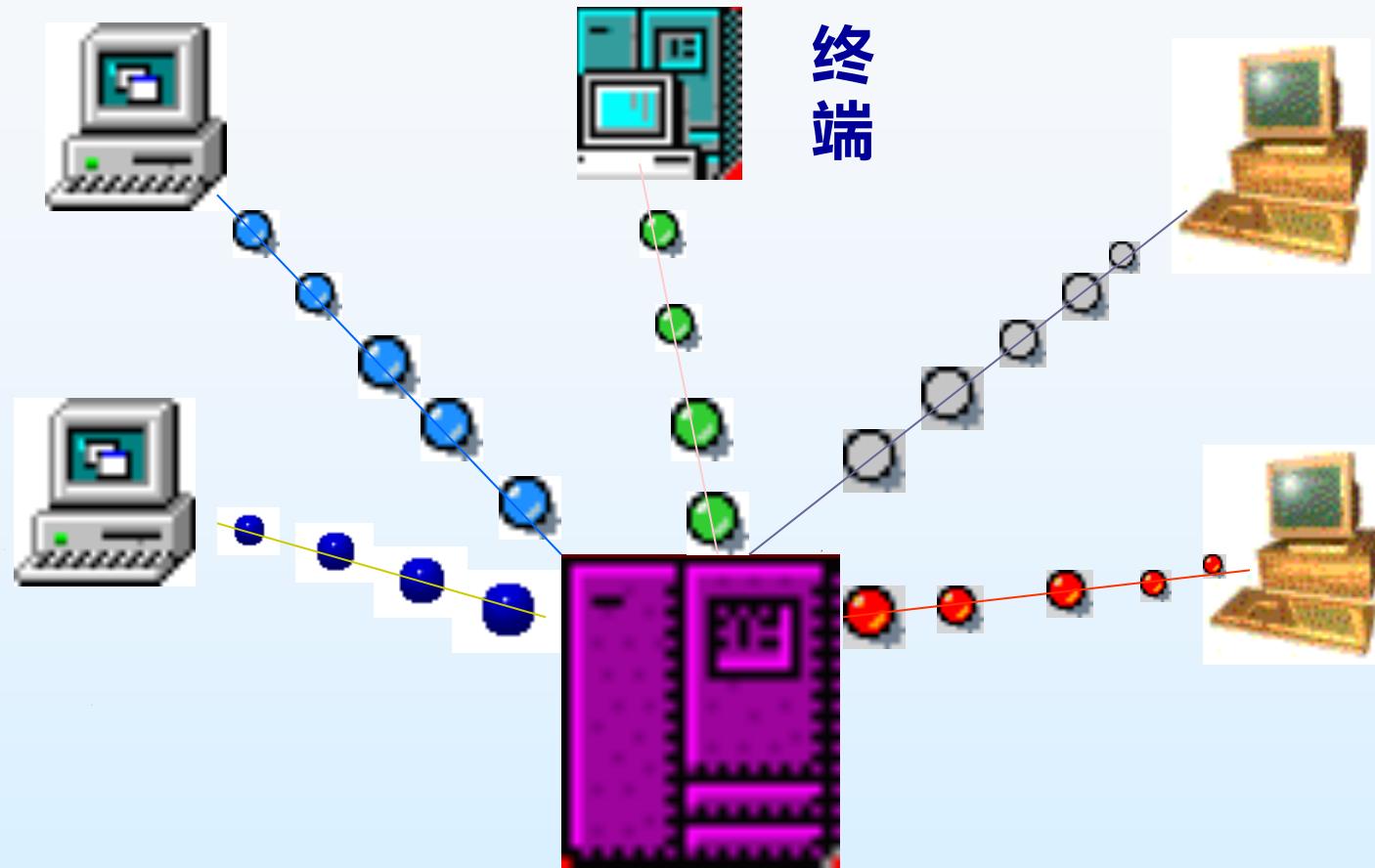


IBM 7094 350万美元





# 分时系统例子

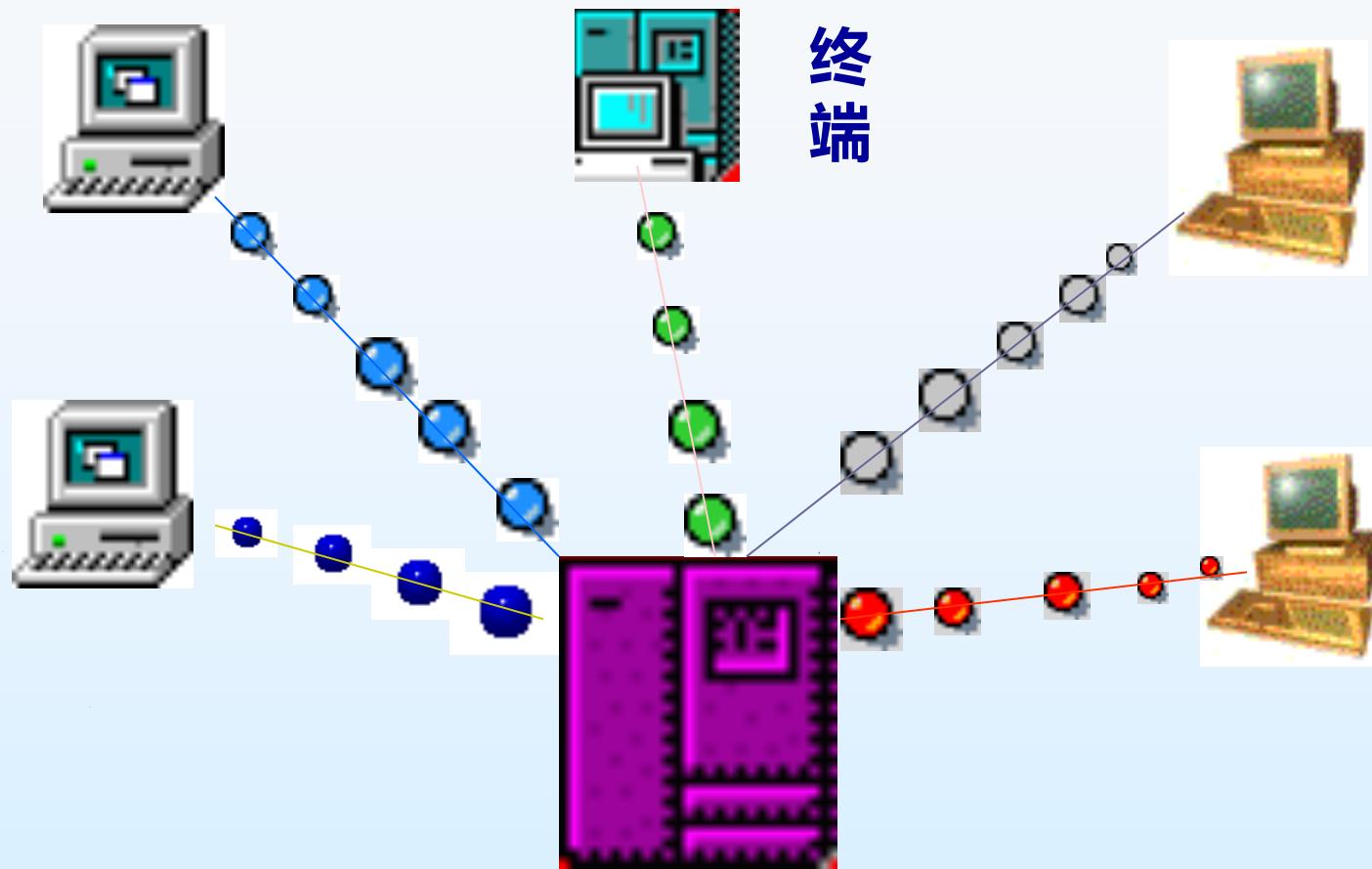


主机





# 分时系统例子



主机



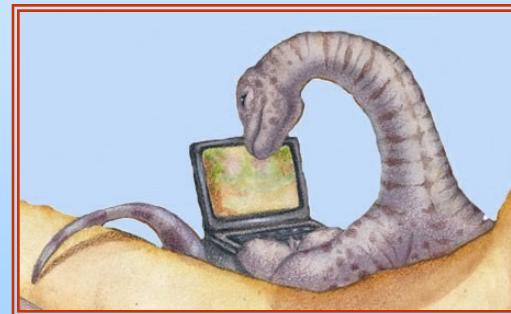


# 分时系统

- 假如分时系统中有**5**个作业，设定时间片大小为**2**毫秒，每个进程在**10**毫秒内各自运行了**2**毫秒。
- 那么，该系统平均响应时间是（ ）毫秒。



### 3、操作系统类型

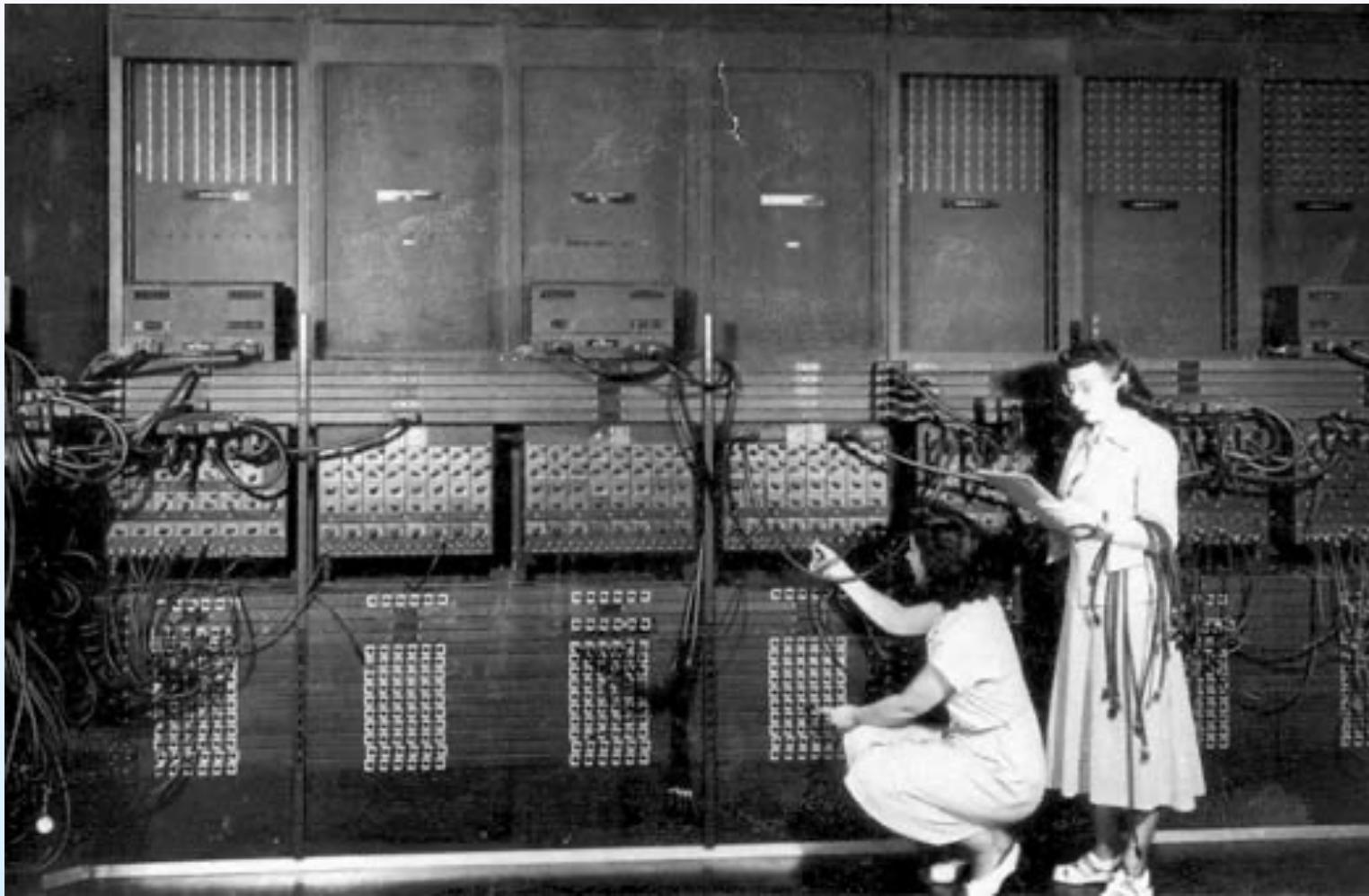




# 操作系统类型

- 无操作系统
- 大型机系统
- 桌面系统
- 嵌入式系统
- 手持系统
- 分布式系统
- 虚拟系统
- 多处理器系统

# 无操作系统计算机





# 大型机系统

- 简单批处理系统
- 多道程序系统
- 分时系统
  
- 大型机通常采用集中式体系架构，具有强大的计算能力和I/O能力
- 一般运行分时系统和批处理系统
- 追求的目标：系统效率
- IBM AIX IBM z/OS HP-Unix





# 桌面系统

- 个人计算机（PC） 20世纪70年代
- 每个人拥有一台计算机
- 设计目标：用户的方便性，响应速度
- 图形化界面（GUI）
- 多数技术沿用大型机技术，但更注重交互性
- Windows Mac OS Linux





# 手持（移动）系统

- 手持设备，如手机、平板
  - 资源局限性，如电池供电、屏幕和键盘偏小、采用触摸屏
  - 扩展功能，如GPS导航、移动支付、位置服务
- 
- 苹果iOS      谷歌Android





# 嵌入式系统

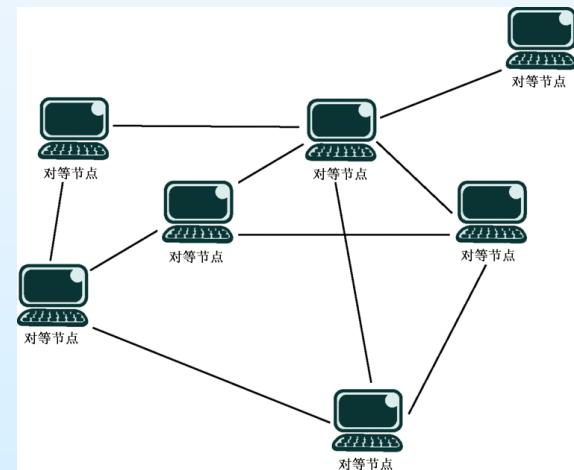
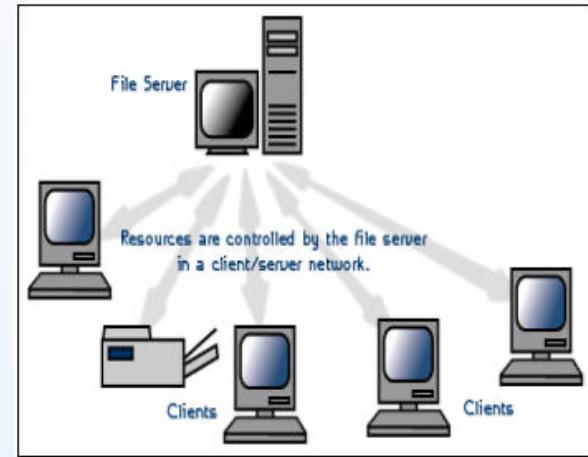
- 嵌入式系统（Embedded System）：完全嵌入受控器件内部，为特定应用而设计的专用计算机系统。
- 随着单片机的出现而出现
- 特点：内核较小、专用性强、系统精简、高实时性
- 应用场景：工业控制、交通管理、信息家电、家庭智能控制、POS网络、环境工程
- 目前大部分嵌入式系统都是在 Linux 内核上进行裁剪得到的





# 分布式系统

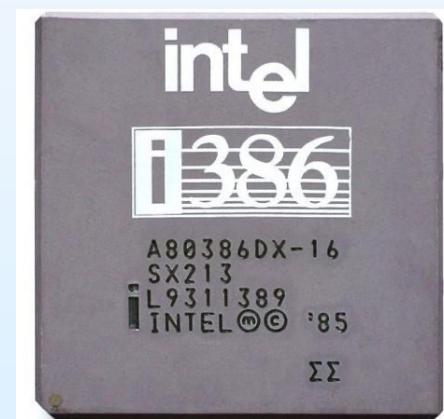
- 支持分布式处理的软件系统，又称松耦合系统
- 通过网络通讯：TCP/IP
- 客户机-服务器系统：客户端与服务器分离，每个客户端程序都可以向服务器端请求服务
- P2P系统：每一个节点既是服务器又是客户机，计算共享的P2P很难实现，主要是数据共享的P2P
- 分布式系统的部分实现





# 单处理器系统

- 传统计算机的操作系统一般是单核处理器系统
- 单核处理器系统
  - 一个CPU
  - 一个CPU一个Core





# 多处理器系统

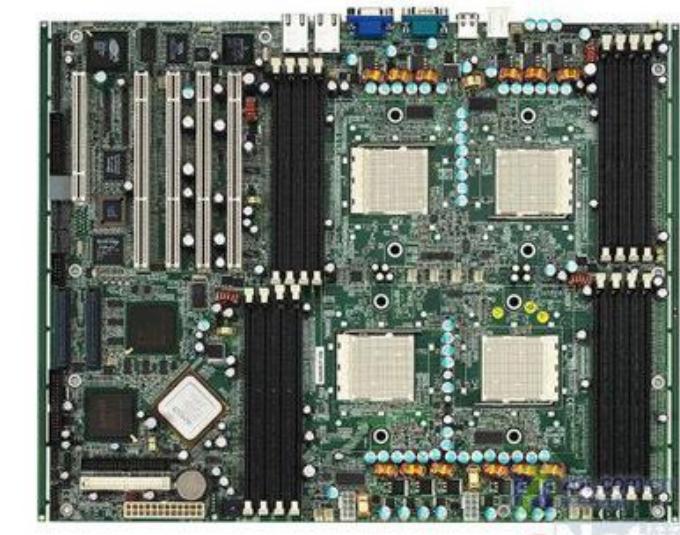
■ 多处理器系统：有多个紧密通信的处理器的系统

- 并行系统(Parallel System)
- 紧耦合系统(Tightly Coupled System)。

■ 多个处理器共享计算机总线、时间、内存和外设等。

■ 多处理器系统的优点：

- 增加吞吐量
- 节省资金
- 增加可靠性





# 多核处理器系统

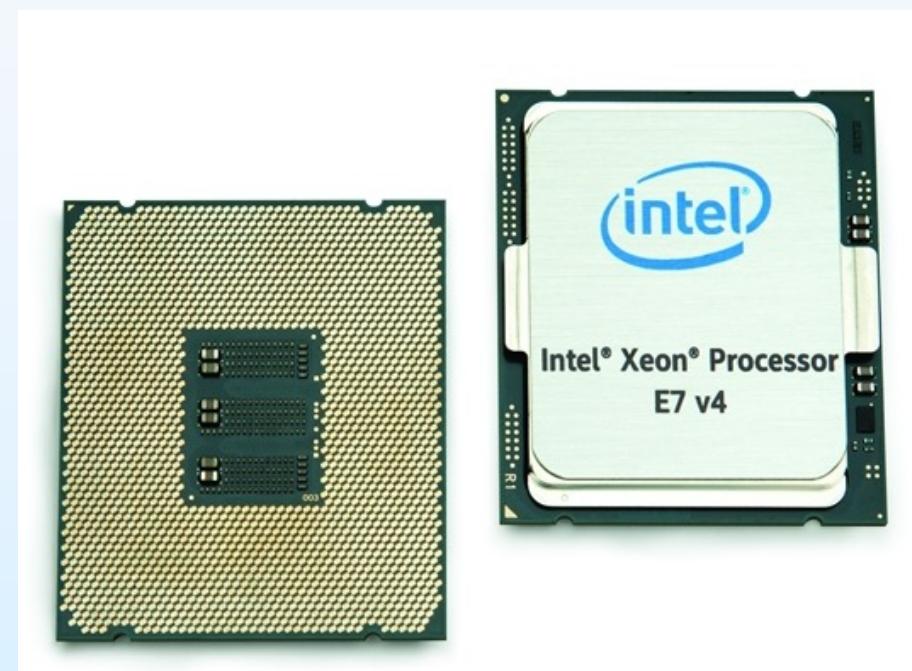
- 为了提高单核处理系统的计算能力，就有了多核处理器系统
- 多核处理器系统
  - 一个CPU
  - 一个CPU多个核，一个核即一个处理器
    - 2核
    - 4核
    - 8核
  - 每个核相当于一个处理器





# Intel Xeon E7

- E7 8890 V4
- 24内核
- 最高频率3.34GHZ
- 60MB Cache

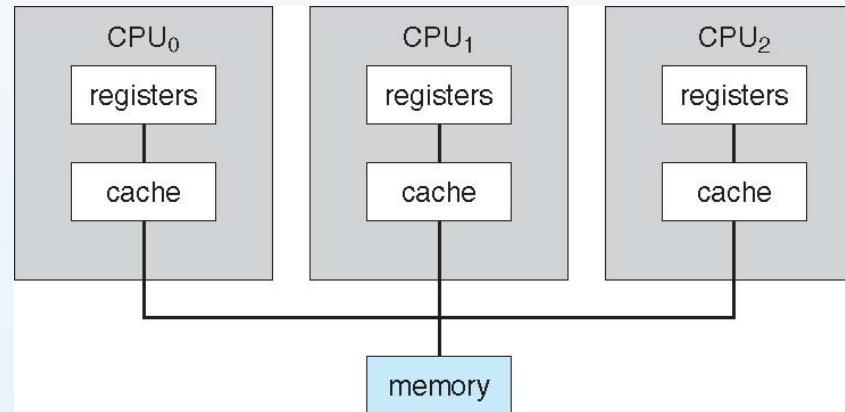




# 多核/多处理器系统分类

## ■ 对称多处理(Symmetric Multiprocessing, SMP)

- 每个处理器运行操作系统的相同副本
- 许多进程可以立即运行不会降低性能
- 多数现代操作系统支持SMP



## ■ 非对称多处理(Asymmetric Multiprocessing, ASMP)

- 每个处理器赋予一个特定任务；主处理器为从处理器调度和分配作业
- 一般服务于特定场景，如游戏、通讯等
- 不适合PC、服务器、主机
- 适合手机等要求功耗的设备，有图像处理和网络处理的专用核





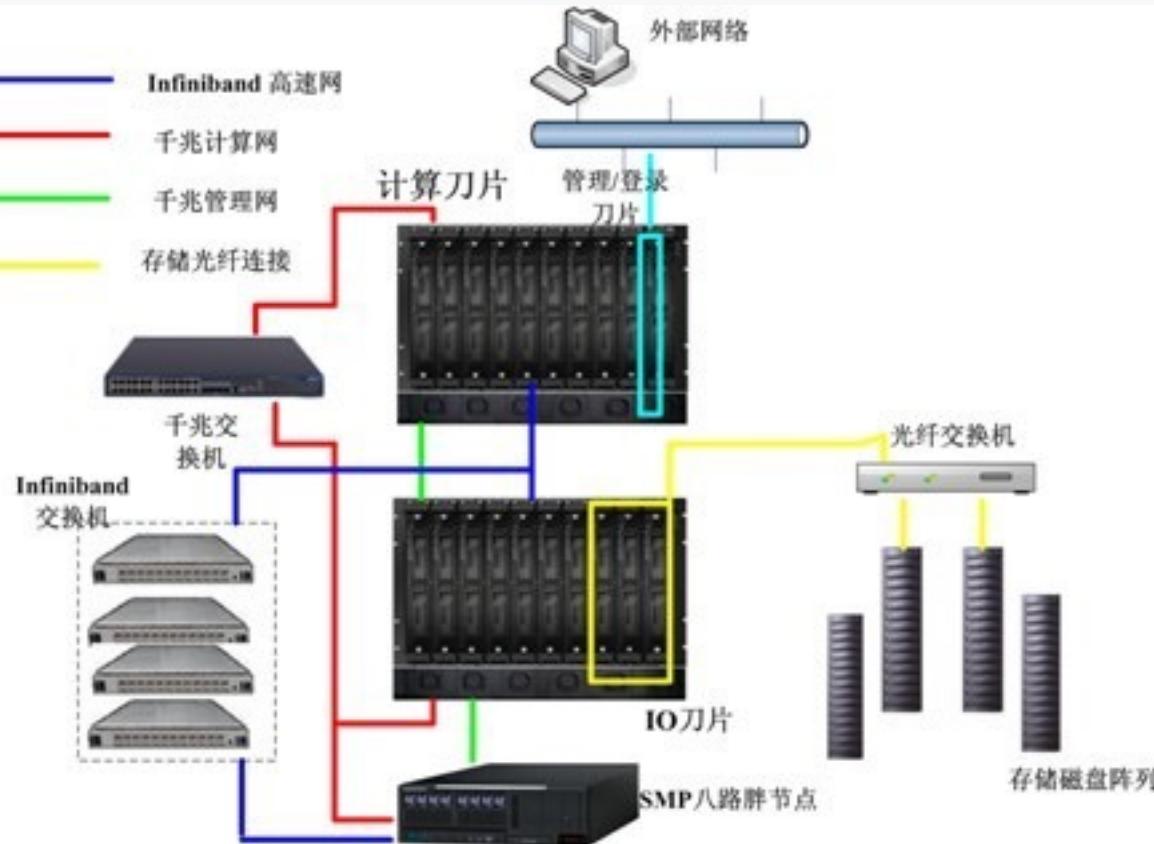
# 集群系统

- 多处理器系统不可能集成大量的处理器，因此单个计算机不能满足超级计算的需求
- 集群系统（**clustered system**）通过专用网络连接一群计算机，把这些计算机利用集群技术虚拟化为一台有超强计算能力的计算机给用户使用
- 由两个或多个独立的系统耦合起来，共享数据，松耦合，每个节点可为单处理器系统或多处理器系统
- 特点：高性能、低成本、高可扩展性、高可靠性
- 非对称集群(**Asymmetric Clustering**)：一台机器运行应用程序，而其他机器处于热备份模式
- 对称集群(**Symmetric Clustering**)：多个主机都运行应用程序





# 集群系统





■ 目前应用于大规模计算的系统一般采用（）。

- A. 多核系统
- B. P2P系统
- C. 紧耦合系统
- D. 集群系统



# 神威 太湖之光

- 2019全球超级计算机前十
- No.3 神威 太湖之光（Sunway TaihuLight）中国
- 处理器：10,649,600 个；峰值速度：125,436 TFlop/s
- 2016年6月，计算能力世界第一，世界上首台运算速度超过十亿亿次的超级计算机，40960块处理器，4核申威26010处理器





# 天河2号

- 2019全球超级计算机前十
- No.4 TH-2 天河二号（中国）
- 处理器：4,981,760个；峰值速度：33,862 TFlop/s





# IBM SUMMIT

- **2018年运算最快计算机**
- 美国橡树岭国家实验室
- 浮点运算速度峰值达到每秒20亿亿次(200PFlops)
- 4608台计算服务器



百家号/图报



## 4、操作系统操作和功能





# 操作系统操作和功能

- 操作系统核心功能（确保操作系统能运行的关键技术）：
  - 双模式
  - I/O和内存保护
  - 定时器
- 操作系统的主要功能：
  - 进程管理
  - 内存管理
  - 文件管理
  - I/O设备管理





# 双重模式操作

## ■ 程序运行中的问题

- 软件错误或特别请求产生异常或陷阱
  - ▶ 除数为零, 完成操作系统服务等
- 其他进程的问题
  - ▶ 死循环, 一个程序可能修改另一个程序
  - ▶ 申请资源过大
- 恶意程序试图篡改系统信息

## ■ 这些问题不处理, 会造成系统异常, 乃至崩溃

- 特别是与硬件相关的程序, 如果用户程序随意使用, 就会造成混乱

## ■ 解决方法: 双模式





# 双重模式操作

- 双重模式：允许OS保护自身和其他的系统部件
  - 用户模式(**User mode**)：所有的用户应用程序只能在用户模式下运行
  - 内核模式(**kernel mode**)：只能运行操作系统的程序
- 双模式需要CPU的支持，如果CPU有模式位（区分系统正在运行用户代码或内核代码），则可以在操作系统中实现双模式，目前主流的CPU都有模式位
- 作用：允许OS保护自身和其他的系统部件不受故障的影响
- 实现技术，引入特权指令
  - 特权指令是可能引起系统崩溃的指令，只能运行在内核模式
- 用户程序需要运行特权指令怎么办？
  - 解决方法：系统调用（**System Call**）

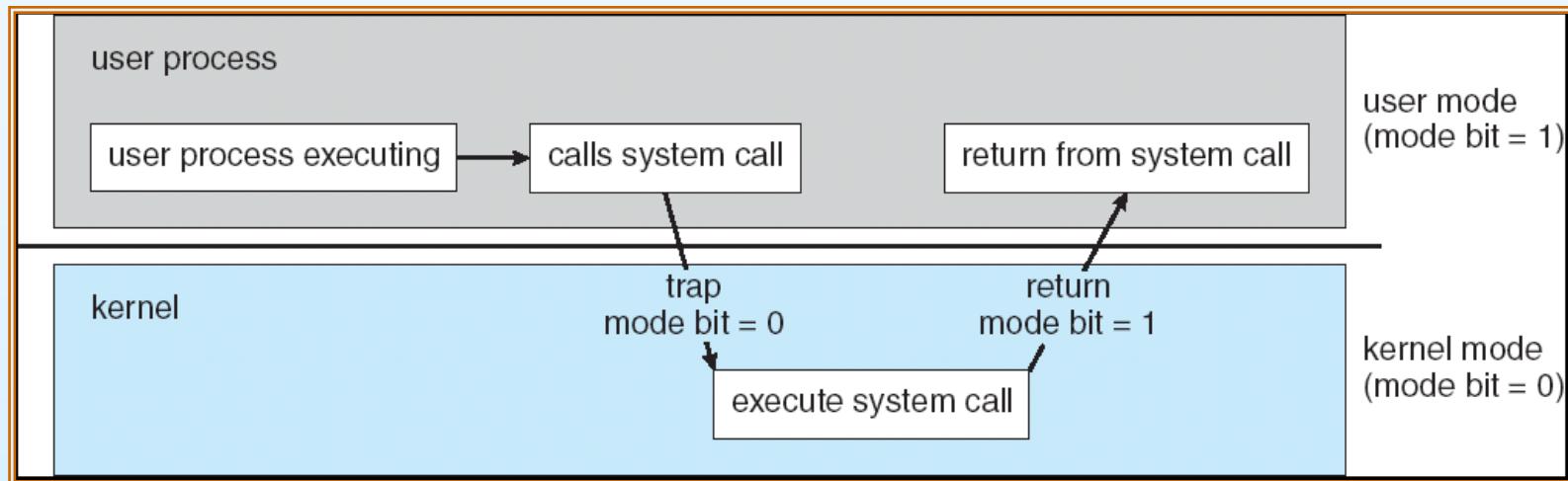




# 用户模式到内核模式的转换

## 系统调用 (System Call)

当运行在用户模式的用户程序需要进行系统调用时，则发生软件中断





# 系统调用

- 系统调用是操作系统向应用程序提供服务的接口。
- 应用程序调用**printf()**, 会触发系统调用**write()**
  - 哪个设备显示字符串
  - 字符串的内容
  - 例如屏幕显示“hello, world”
- 有了这些接口, 应用程序就可以完成各种各样的操作, 对操作系统进行间接的控制和管理
- 程序访问是通过高层次的**API**接口而不是直接进行系统调用
  - Win 32 API 用于 Windows
  - POSIX API 用于 POSIX-based system (包括 Unix, Linux, Mac OS X 各版本)
  - Java API 用于 JAVA 虚拟机 (JVM)





# I/O操作保护和内存保护

## ■ 防止用户程序非法执行I/O操作

- 操作系统定义**所有的I/O指令是特权指令**
- 用户程序不能直接访问I/O设备，而是通过系统调用进行I/O操作
- 利用双模式和设置所有的I/O指令是特权指令就可以实现I/O保护，防止用户非法I/O操作

## ■ 防止内存非法访问

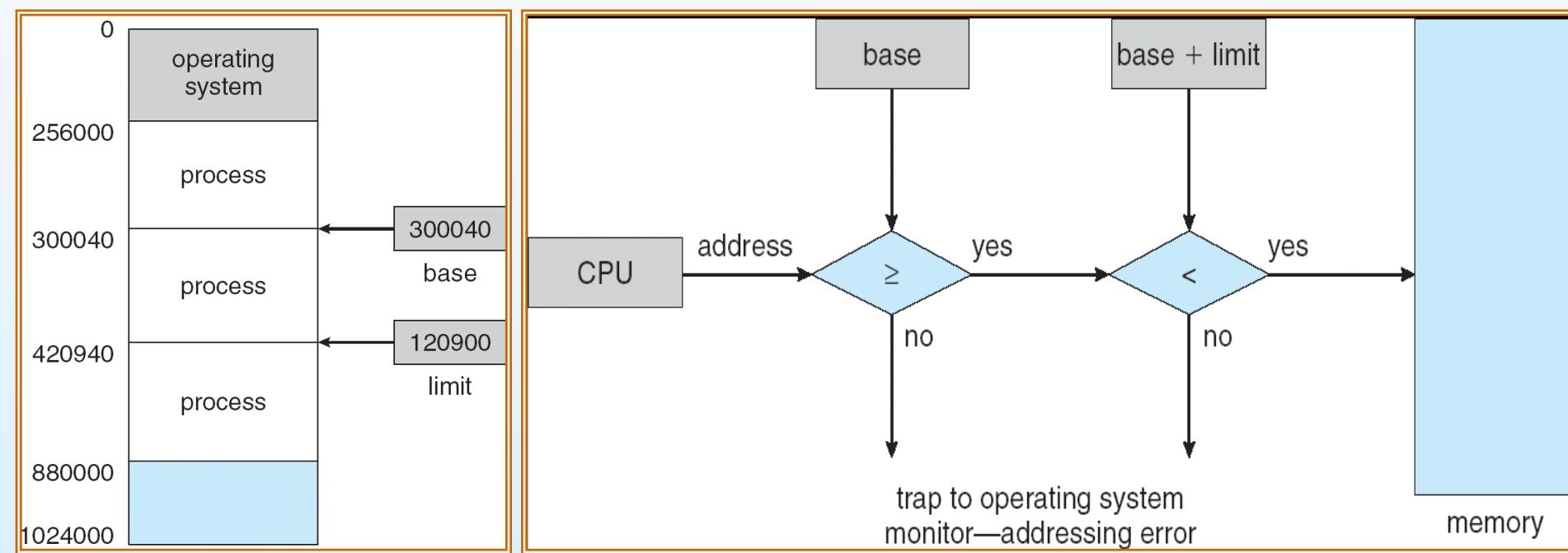
- 存储保护机制
- 硬件支持





# 内存保护

- 基址寄存器 (Base register)
- 限长寄存器 (Limited register)





# 定时器

- 如果用户程序死循环，或用户程序不调用系统调用，那么操作系统将无法获得**CPU**并对系统进行管理
- 所以操作系统必须能够随时获得**CPU**的控制权，并管理各种资源
- 解决方法：引入定时器
  - 系统在将**CPU**的控制权交给用户程序之前设置好定时器
  - 该定时器在一段时间后发生中断，**CPU**控制权返回操作系统
  - 定时器分为固定时间和可变时间定时器，一般利用时钟和计数器实现





■ 以下指令不属于特权指令的是（ ）

- A. I/O指令
- B. 定时器访问指令
- C. 文件访问指令
- D. 核心内存访问指令





# 操作系统的主要功能

- 进程管理
- 内存管理
- 文件管理
- I/O设备管理



# 进程管理

- 操作系统的核心目标
  - 运行程序，也就是如何管理CPU
- 现代操作系统引入进程的概念，一个运行中的程序称为进程
- CPU管理也称为进程管理
  - 创建和删除用户和系统进程
  - 暂停和恢复进程
  - 提供进程同步机制
  - 提供进程通讯机制
  - 提供死锁处理机制



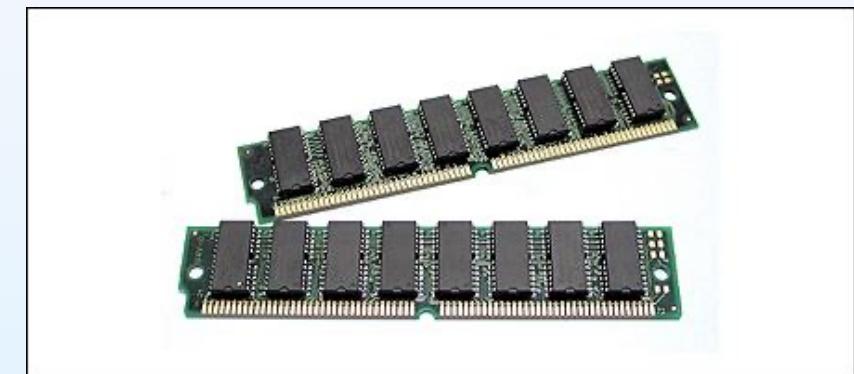
电脑百事网  
WWW.PCBAI.COM





# 内存管理

- 内存是程序运行必须的存储设备，因为**CPU**只能直接访问寄存器、高速缓存和内存这三类存储设备。
- 一个进程处理前和处理后的所有数据、执行的指令都在内存中。
- 内存管理的主要工作：
  - 内存分配
  - 内存回收
  - 地址转换
  - 共享和保护
- 内存管理的目的：
  - 提高内存利用率和访问速度，从而提高计算机运行效率





# 文件管理

- 现代操作系统一般以文件为单位，以目录为组织方式构建文件系统，并把文件系统存储在磁盘等二级存储设备上
- 文件管理的内容包括：
  - 文件系统
  - 文件逻辑结构
  - 文件物理结构
  - 目录
  - 文件检索方法
  - 文件操作
  - 空闲空间管理
  - 存储设备管理





# I/O设备管理

■ I/O设备管理负责管理种类繁多的各种I/O设备，解决计算机中信息的输入和输出问题。

■ 核心技术：

- 设备无关性（独立性），即操作系统把所有物理设备按照物理特性分门别类地抽象为逻辑设备。
- 应用程序不是针对物理设备编程，而是针对逻辑设备编程。
- 应用程序和数量、种类繁多的物理设备无关，主要针对抽象出来的若干逻辑设备编程即可。

■ 主要工作：

- 设备管理
- 设备驱动

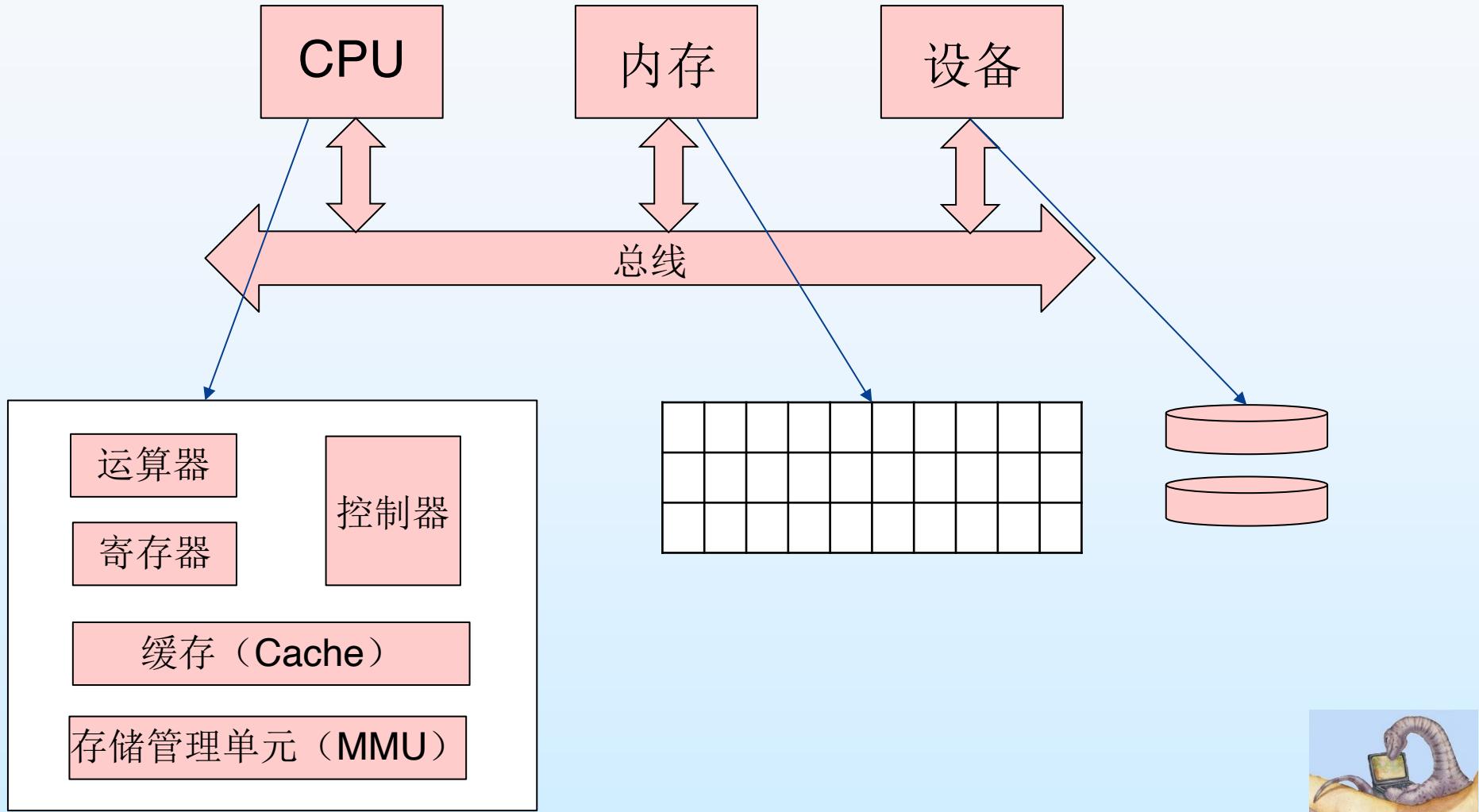




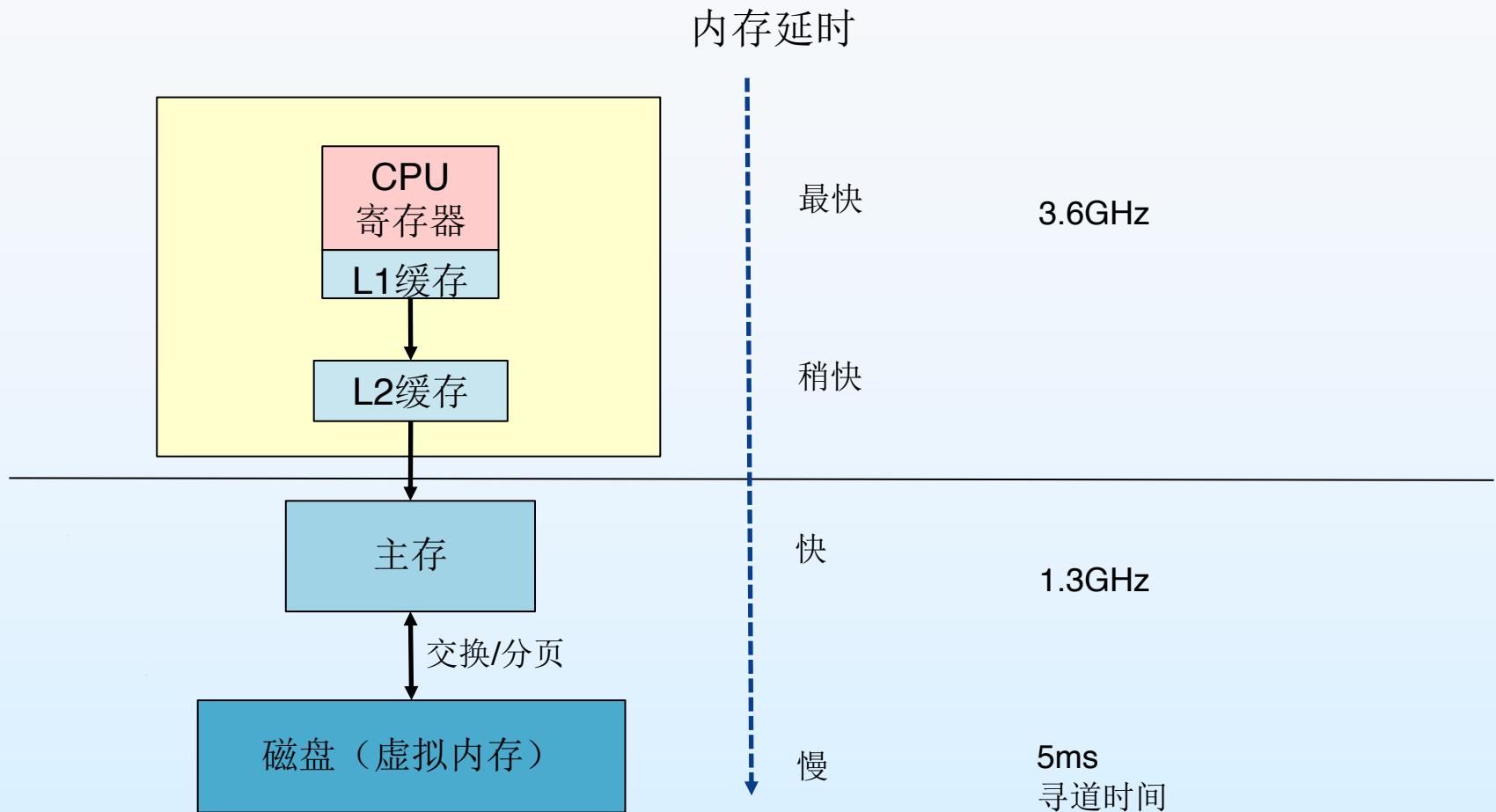
- 进程管理、内存管理、文件管理、I/O设备管理这四个核心部分中，和程序运行最密切相关的是（ ）。
- A. 进程管理
  - B. 内存管理
  - C. 文件管理
  - D. I/O设备管理

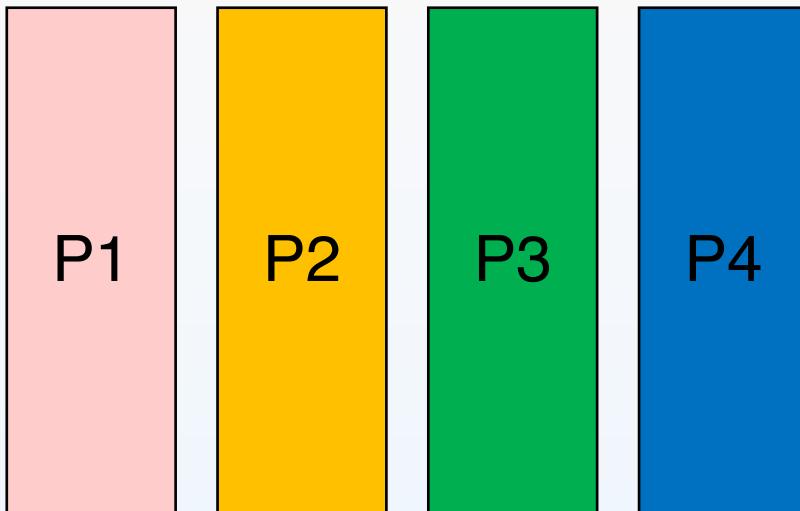


# 计算机基本硬件结构



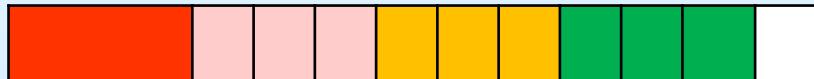
# 内存的层次结构



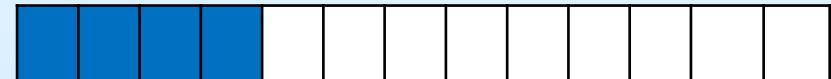


逻辑（虚拟）空间

物理空间



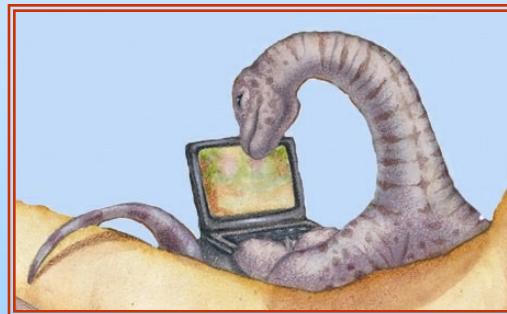
主存



磁盘（虚拟内存）



## 5、计算环境





# 传统计算

## ■ 传统计算

- 随着时间的推移而模糊
- 办公环境
  - ▶ 一些联网的PC，终端与提供批处理和分时的大型机或小型机相连
  - ▶ 现在门户网站(portal)允许网络和远程系统访问相同的资源
- 家庭网络
  - ▶ 以前是单一的系统，通过调制解调器相连
  - ▶ 现在使用防火墙相连





# 分布式计算

- 分离的、可能异构的系统集合
  - 通过网络通信： TCP/IP
    - ▶ Local Area Network (LAN)
    - ▶ Wide Area Network (WAN)
    - ▶ Metropolitan Area Network (MAN)
    - ▶ Personal Area Network (PAN)
- 网络操作系统（Network Operating System）





# 移动计算

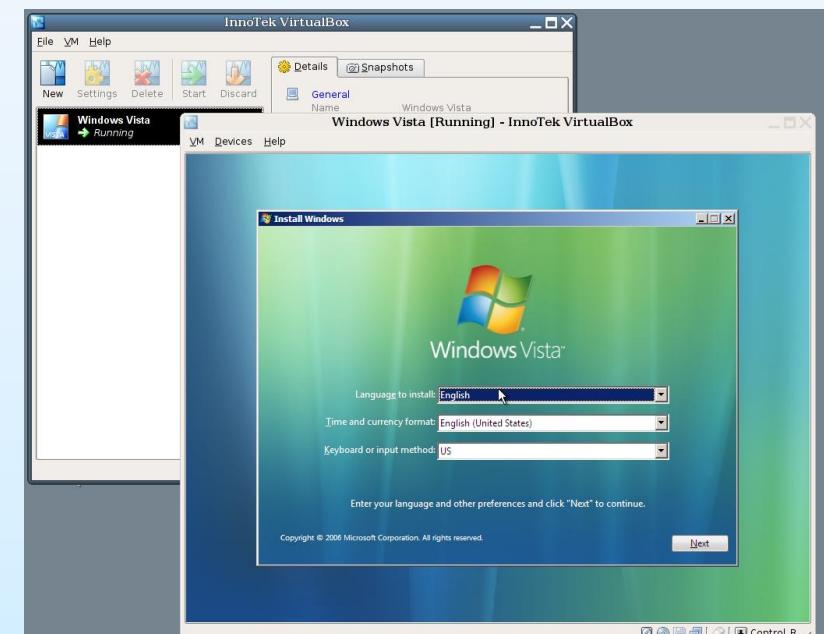
- 手持设备，如手机、平板和传统笔记本不同
- 扩展功能
  - GPS导航
  - 移动支付
  - 位置服务
- Apple iOS
- Google Android





# 虚拟计算

- 虚拟计算：建立在虚拟环境中的计算模式
- 模拟器：
  - 不同CPU模拟
  - 不同操作系统模拟
- 虚拟机
  - VMware下运行WinXP
  - 服务器虚拟化
  - 桌面虚拟化
  - 存储虚拟化
  - 网络虚拟化





# 云计算

## ■ 美国国家标准与技术研究院（NIST）定义：

- 云计算是一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池（资源包括网络，服务器，存储，应用软件，服务），这些资源能够被快速提供，只需投入很少的管理工作，或与服务供应商进行很少的交互。

## ■ 以虚拟计算为基础，是虚拟计算的逻辑扩展

- Amazon EC2：几千服务器，几百万虚拟机，PT级别存储
- 阿里云

## ■ 类别

- 公共云（Public cloud）
- 私有云（Private cloud）
- 混合云（Hybrid cloud）

## ■ 服务模式

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)



# 云计算架构

