OPERATING SYSTEMS

Organization & Architectures

Operating Systems

郑贵锋 博士
中山大学计算机学院
zhenggf@mail.sysu.edu.cn
https://gitee.com/code_sysu

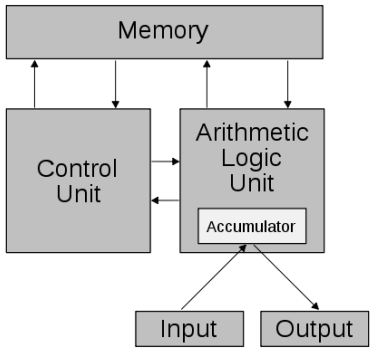
中山大學 SUN YAT-SEN UNIVERSITY

Computer System Organization & Architectures

- **大纲**
  - 计算机硬件结构
    - *冯·诺依曼结构*
    - 哈佛建筑
  - 计算机系统组织
    - 中断
    - 存储结构
    - 主存管理
    - I/O结构
  - 计算机系统结构与环境
    - 多道程序批处理系统
    - 分时系统
    - 实时系统
    - 个人/台式计算机
    - 多处理器系统
    - 集群系统
    - 网络系统与分布式系统
    - 基于Web的系统
    - 手持系统和移动系统

Computer Hardware Architectures

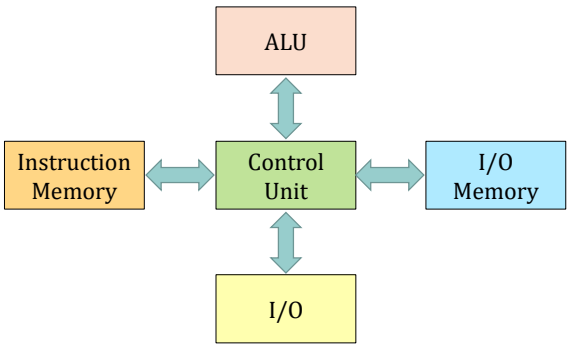
- **计算机硬件结构**
  - *冯·诺依曼结构*
    - 又名普林斯顿结构。
    - 例如，X86…



冯·诺依曼结构

Computer Hardware Architectures

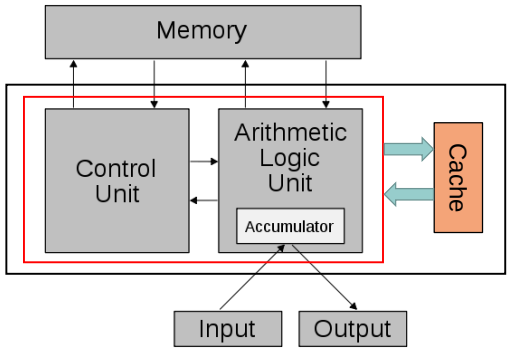
- **计算机硬件结构**
  - 哈佛结构
    - 例如，ARM9、MIC、大多数DSP…
    - 修改的哈佛架构…



哈佛结构

Computer Hardware Architectures

- 计算机硬件结构
  - 混合结构
    - 在CPU中引入缓存

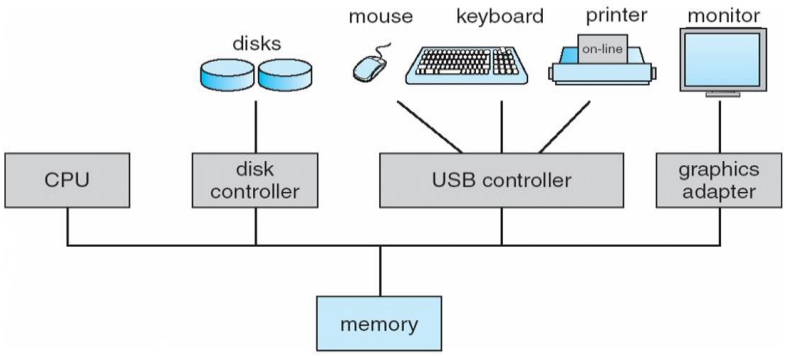


Computer Hardware Architectures

- 计算机硬件结构
  - 基本属性
    - 数据表示
    - 寻址方式
    - 寄存器
    - 指令系统
    - 内存系统
    - 中断控制器
    - 输入/输出控制器
    - 信息保护机制

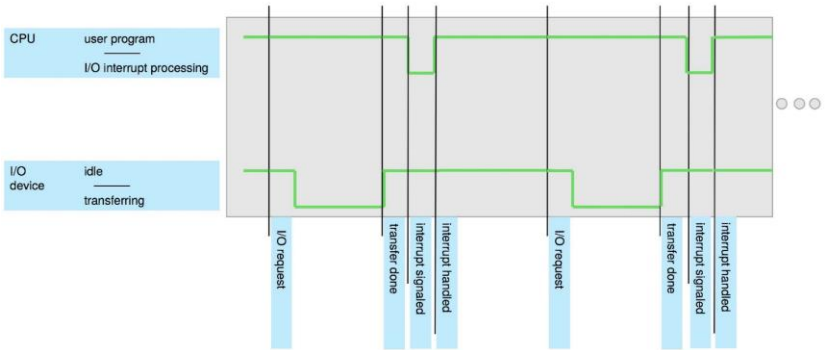
Computer System Organization

- **计算机系统组织**
  - 现代通用计算机系统由一个或多个CPU和多个通过公共总线连接的设备控制器组成，该总线提供组件和共享内存之间的访问。
  - 操作系统中，每个设备控制器都有一个设备驱动程序。
  - CPU和设备控制器可以并行执行，争夺内存周期。



Computer System Organization

- **中断**
  - 中断是操作系统和硬件交互的关键部分。
    - 硬件可通过向CPU发送信号（通常通过系统总线）随时触发中断。
    - CPU停止它正在做的事情，并立即将执行转移到中断服务例程（ISR）。
    - 完成后，CPU恢复中断的计算。

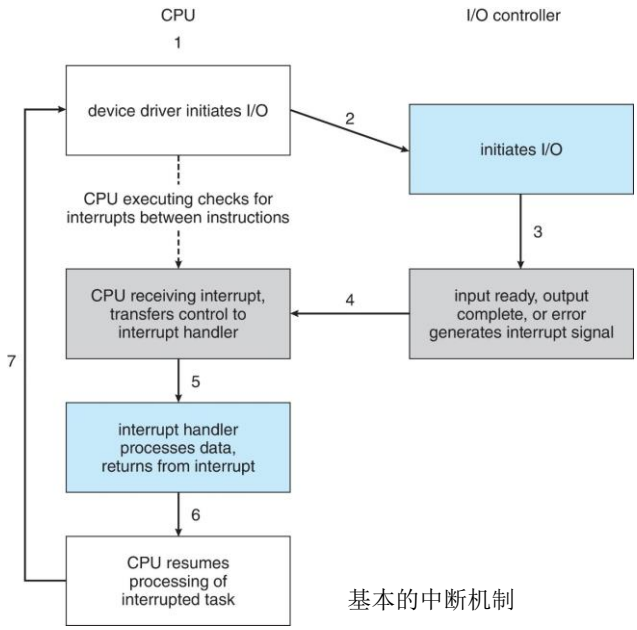


Computer System Organization

- **中断**
  - 中断是计算机体系结构的重要组成部分。每种计算机设计都有自己的中断机制。
  - 基本中断机制的工作原理：使CPU能够响应异步事件（设备控制器准备好服务时）
    - CPU在执行每个指令后感知中断请求行。控制器可以在线路上断言信号。CPU读取中断号并跳转到中断处理程序例程。
      - 设备控制器通过在中断请求行上断言一个信号来引发中断，CPU捕获中断并将其发送给中断处理程序，处理程序通过服务设备来清除中断。
    - ISR地址指针表（该表又称中断向量）用于提供必要的速度。
      - 该表存储在低内存中。
      - 它由中断请求中给出的唯一中断号索引。
      - Windows和UNIX调度中断就是以这种方式发生的。

Computer System Organization

- **中断**



基本的中断机制

Computer System Organization

- **中断**
  - 中断控制器
    - 在现代操作系统中，CPU和中断控制器硬件提供了更复杂的中断处理功能。
      (1)在关键处理期间延迟中断处理的能力。
      (2)为设备分派适当中断处理程序的有效方法。
      (3)多级中断，以便操作系统能够区分高优先级中断和低优先级中断，并以适当的紧急程度作出响应。
  - 可屏蔽和不可屏蔽中断
    - 大多数CPU有两条中断请求线。
      - 不可屏蔽（maskable）中断线
        - 为不可恢复内存错误等事件保留。
      - 可屏蔽中断线
        - 由设备控制器用于请求服务。
        - 在执行不可中断的关键指令序列之前，可由CPU关闭。

Computer System Organization

- **中断**
  - 英特尔处理器事件向量表。

	vector number	description
non-maskable (0-31)	0	divide error
	1	debug exception
	2	null interrupt
	3	breakpoint
	4	INTO-detected overflow
	5	bound range exception
	6	invalid opcode
	7	device not available
	8	double fault
	9	coprocessor segment overrun (reserved)
	10	invalid task state segment
	11	segment not present
	12	stack fault
	13	general protection
	14	page fault
	15	(Intel reserved, do not use)
	16	floating-point error
	17	alignment check
	18	machine check
	19–31	(Intel reserved, do not use)
maskable (32-255)	32–255	maskable interrupts

Computer System Organization

- **中断**
  - 中断链
    - 如果计算机的设备（中断处理程序）多于中断向量中的地址元素，则使用中断链接，其中中断向量中的每个元素都指向中断处理程序列表的开头。
    - 当一个中断被触发时，相应列表上的处理程序被逐个调用，直到找到一个可以为请求提供服务的处理程序为止。
    - 中断链结构是一种在庞大中断表的开销和分配给单个中断处理程序的低效性之间的折衷。
  - 中断优先级
    - 中断机制还实现了中断优先级系统。这些级别使CPU能够在不屏蔽所有中断的情况下延迟低优先级中断的处理，并使高优先级中断能够抢占低优先级中断的执行。

Computer System Organization

- **中断**
  - 中断类型和属性
    - 操作系统是中断驱动的。有三种中断类型：
      - 陷阱 (异常)
      - 外部中断
      - 系统调用
    - 各种中断属性
      - 异步与同步
      - 外部/硬件与内部/软件
      - 隐式与显式

<table>
<tr><td></td><td colspan="2">中断类型</td><td></td></tr>
<tr><td>异步的</td><td>外部的中断</td><td></td><td rowspan="2">隐式的</td></tr>
<tr><td rowspan="2">同步的</td><td rowspan="2"></td><td>陷阱</td></tr>
<tr><td>系统调用</td><td>显式的</td></tr>
<tr><td></td><td>外部的/硬件</td><td>内部的/软件</td><td></td></tr>
</table>

Computer System Organization

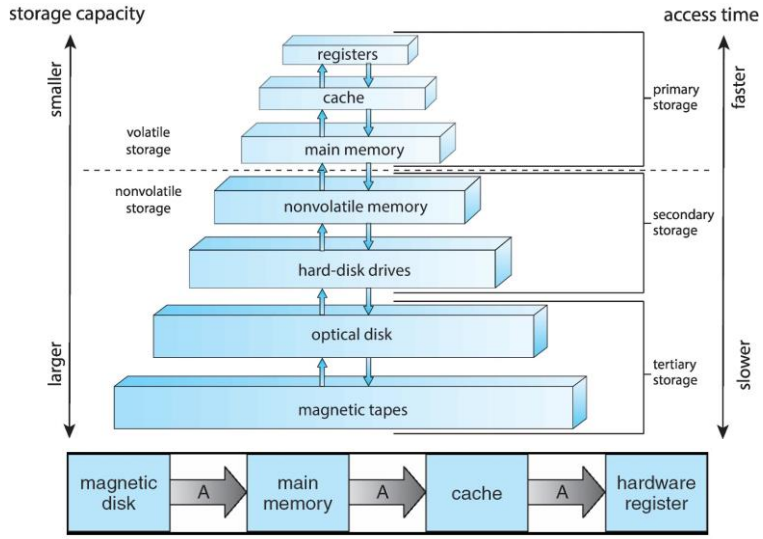
- 存储结构
  - 主存
    - CPU可以直接访问的唯一大型存储介质
    - 通常太小，无法永久存储所有需要的程序和数据
    - 易变的(易失的)
      - 当电源关闭或以其他方式丢失时，会丢失其内容
  - 辅助存储器
    - 大容量
    - 非易失性存储器（NVM，nonvolatile memory)
  - 硬盘
    - HDD是最常见的辅助存储设备。
      - 覆盖有磁记录材料的硬质金属或玻璃盘
      - 柱面、轨道和扇区
    - 磁盘控制器确定设备和主机之间的逻辑交互。

Computer System Organization

- 存储结构
  - 缓存
    - 缓存是计算机中在多个级别执行的一项重要原则
      - 正在使用的信息临时从较慢的存储复制到较快的存储。
      - 硬件、操作系统、应用程序。
    - 首先检查更快的存储（缓存）以确定是否存在信息。
      - 如果是，则直接从缓存使用信息。
        - 更快
      - 否则，数据将复制到缓存并使用。
    - 缓存小于正在缓存的存储。
      - 缓存管理是一个重要的设计问题。
      - 缓存大小和替换策略很重要
    - 在大多数情况下，缓存特别是指CPU内的SRAM

Computer System Organization

- **存储结构**
  - 存储层次结构



Computer System Organization

- **存储结构**
  - 不同存储级别的性能。

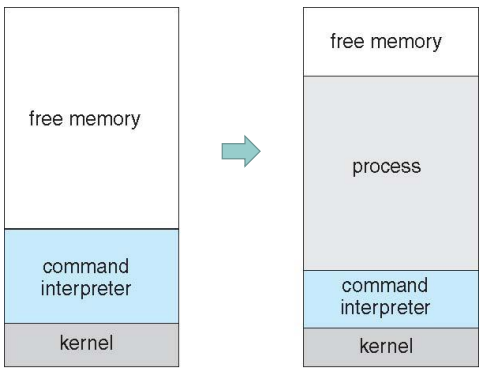
Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Computer System Organization

- **主存管理**
  - 初始内存管理技术
    - 最低限度管理
      - 一种自行管理内存的程序。
      - 这里没有内存保护问题。
    - 内存分割Split
      - 常驻监控程序(驻留程序) 和用户作业（用户程序）在它们之间分割内存。
    - 内存分区Division
      - 操作系统和一些用户作业在它们之间分配可用内存。

Computer System Organization

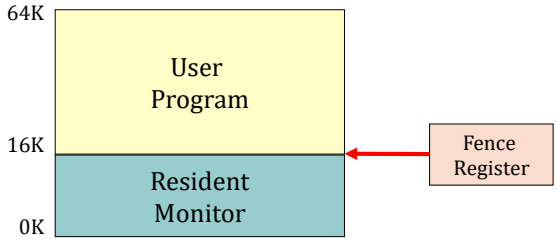
- **主存管理**
  - 初始内存管理技术



MS-DOS内存拆分

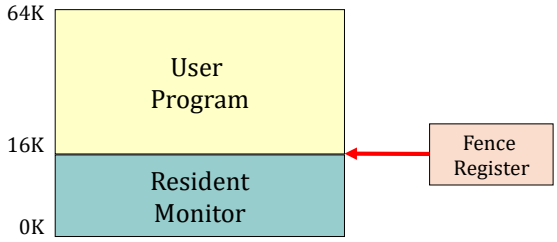
Computer System Organization

- 主存管理
  - 动态内存管理
    - 共享系统资源要求操作系统确保错误的程序不会导致其他程序错误执行。
    - 常驻监控程序是一个“受信任的程序”，但是如何保护它不受用户程序的破坏呢？
      - 解决方案：使用界限寄存器(Fence Register)和寻址访问逻辑



Computer System Organization

- 主存管理
  - 动态内存管理
    - 界限寄存器加载用户程序的基址（也是常驻监控程序的界限）
    - 用户程序可以读取任何地址，但寻址访问逻辑确保它只能写入大于界限寄存器值的地址。
    - 加载界限寄存器的指令必须具有特权（它只能由常驻监控程序执行）
      - 但如何确保这一点呢？



Computer System Organization

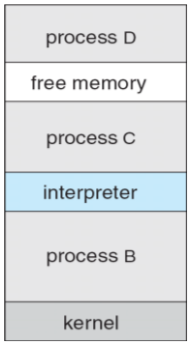
- 主存管理
  - 双模式操作
    - 提供硬件支持以区分至少两种操作模式：
      - 用户模式：代表用户执行。
      - 内核模式：代表操作系统执行。
    - 用户程序永远无法在内核模式下控制计算机。
    - 当发生任何类型的中断时，中断硬件切换到内核模式，执行内核地址空间中的正确ISR
    - 特权指令只能在内核模式下执行。
      - 解决方案：在计算机硬件中添加一个模式位指示当前模式
        - 模式位保存在状态寄存器中
        - 内核/系统/监控程序/管理员（0）或用户（1）。

Computer System Organization

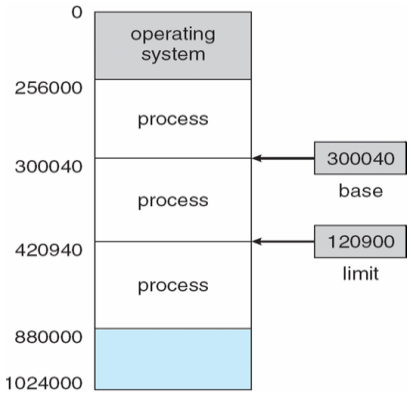
- 主存管理
  - 内存分区保护
    - 为了具有内存分区保护，添加了两个寄存器以确定程序可以访问的合法地址范围。
      - 基址寄存器 (Base Register)
        - 保存程序的最小合法物理内存地址。
        - 也被称为Lower Fence寄存器。
      - 限制寄存器 (Limit Register)
        - 包含范围的大小。
        - 也被称为Upper Fence寄存器。
    - 超出定义范围的内存受到保护。

Computer System Organization

- 主存管理
  - 内存分区保护



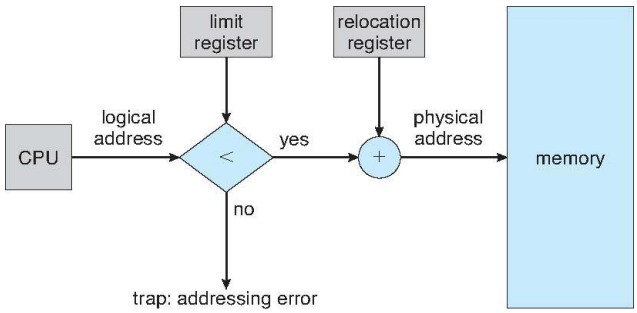
Unix内存分区



基址寄存器和限制寄存器示例

Computer System Organization

- 主存管理
  - 保护硬件
    - 在内核模式下执行时，操作系统可以不受限制地访问系统和用户的内存。
    - 基址寄存器和限制寄存器的加载指令是特权指令。
      - 特权指令只能在内核模式下发出。
      - 这些寄存器的读取指令不需要具有特权。



Computer System Organization

- **主存管理**
  - 陷阱Traps
    - 陷阱/异常是由程序错误引起的软件生成的中断。
      - 陷阱使用中断硬件切换到内核模式。
    - 实例
      - 算术溢出/下溢
      - 除零
      - 执行非法指令
      - 用户内存空间之外的引用
    - 陷阱也可以由程序中的显式陷阱指令启动
    - *陷阱、故障和可编程异常
      - 陷阱：在EIP中保存下一PC（例如，调试点）
      - 错误：在EIP中保存当前PC（例如，页面错误）
      - 可编程异常：系统调用（int 0x80），将下一PC保存在EIP中

（PC: Program Counter，指令地址）

Computer System Organization

- 主存管理
  - 内存保护摘要
    - 如何保护内存空间中的作业？
      - 使用界限寄存器和寻址访问逻辑
    - 如何保护界限寄存器？
      - 使用特权界限加载指令
    - 如何确保特权执行？
      - 使用模式位
    - 但是如何保护模式位呢？
      - 仅通过中断硬件更改为内核模式！

Computer System Organization

- **I/O结构**
  - 大部分操作系统代码专用于管理I/O。
    - 这对系统的可靠性和性能很重要。
    - 这也是因为设备的不同性质。
  - 中断驱动I/O
    - 移动少量数据（可能是1或64字节），可以接受
    - 批量数据移动时，将高开销
  - 直接存储器存取（DMA）
    - 在为I/O设备设置缓冲区、指针和计数器后，设备控制器将整个数据块（可能是216字节）直接传输到设备和主存，或从设备和主存传输，而无需CPU干预。
    - 每个块只生成一个中断，告诉设备驱动程序操作已完成，而不是为低速设备生成的每个字节一个中断。
    - 当设备控制器执行这些操作时，CPU可用于完成其他工作。

Computer System Architectures & Environments

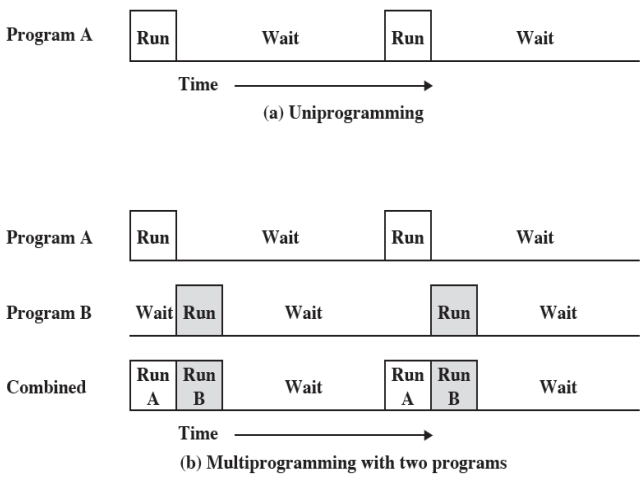
- **多道程序批处理系统**
  - 为了提高效率，需要多道程序设计（又称多任务）。
    - 单用户无法使CPU和I/O设备始终处于繁忙状态。
    - 多道程序设计组织作业（代码和数据），所以CPU总是有一个作业要执行。
    - 系统中所有作业的一个子集保存在内存中。
      - 通过作业调度选择和运行其中的一个作业。
      - 当运行的作业必须等待（例如，等待I/O）时，操作系统将切换到另一个作业。
  - 批处理多道程序不支持与用户交互。

多道程序批处理系统

- 批处理多道程序的内存布局
  - 多个作业同时保存在主存储器中，CPU在它们之间进行多路复用。

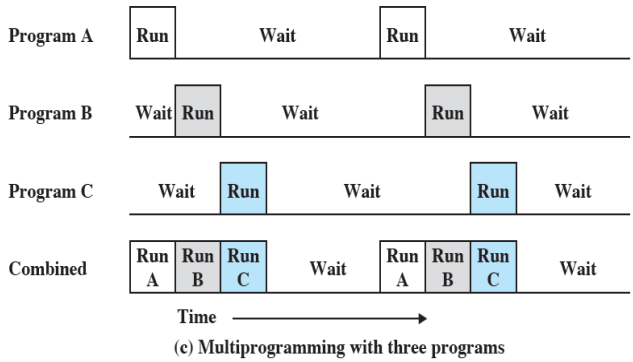
多道程序批处理系统

- 批处理多道程序的CPU效率。



多道程序批处理系统

- 批处理多道程序的CPU效率。



(c) Multiprogramming with three programs

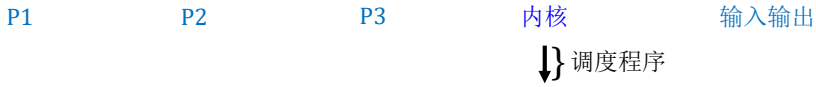
多道程序批处理系统

- 多道程序设计示例–时间序列图。

P1 P2 P3 内核 输入输出

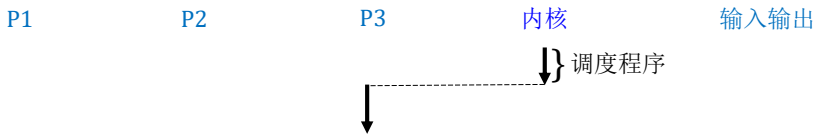
多道程序批处理系统

- 多道程序设计示例–时间序列图。



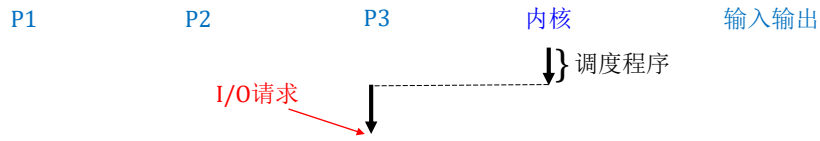
多道程序批处理系统

- 多道程序设计示例–时间序列图。



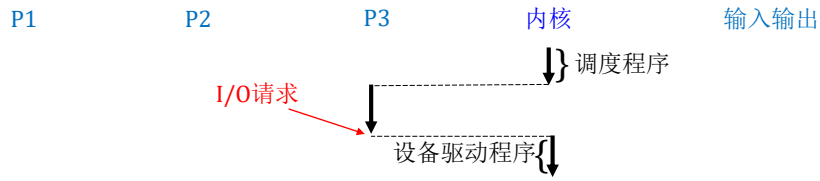
多道程序批处理系统

- 多道程序设计示例–时间序列图。



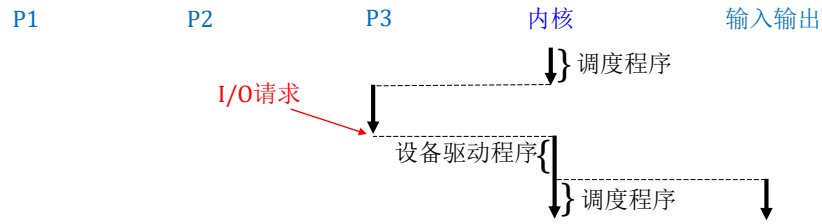
多道程序批处理系统

- 多道程序设计示例–时间序列图。



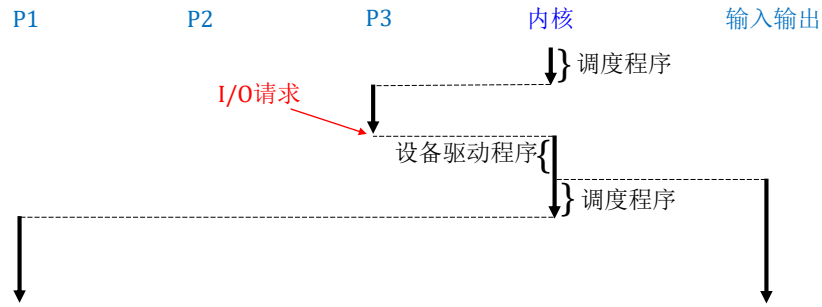
多道程序批处理系统

- 多道程序设计示例–时间序列图。



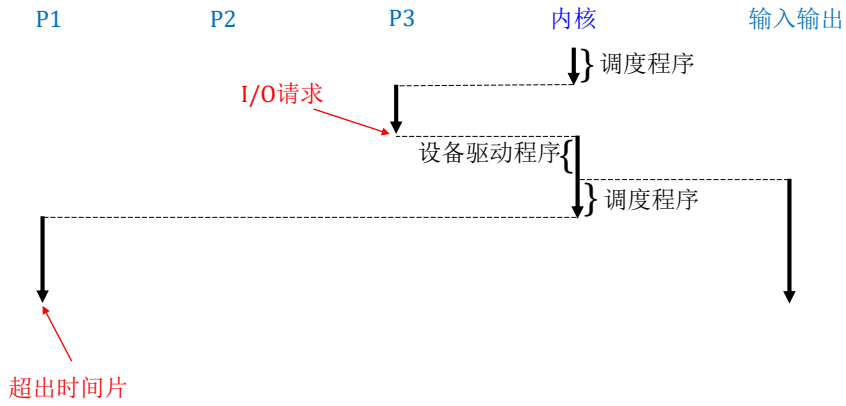
多道程序批处理系统

- 多道程序设计示例–时间序列图。



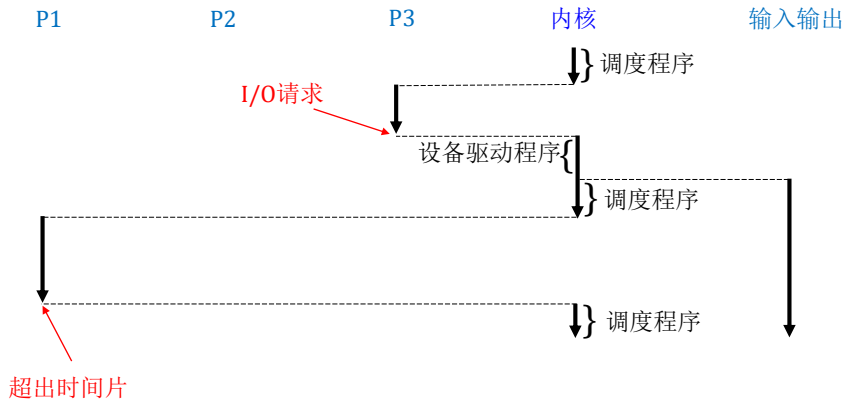
多道程序批处理系统

- 多道程序设计示例–时间序列图。



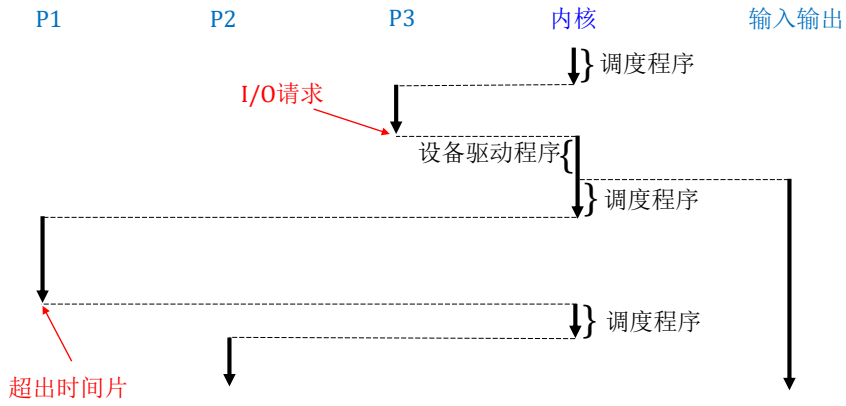
多道程序批处理系统

- 多道程序设计示例–时间序列图。



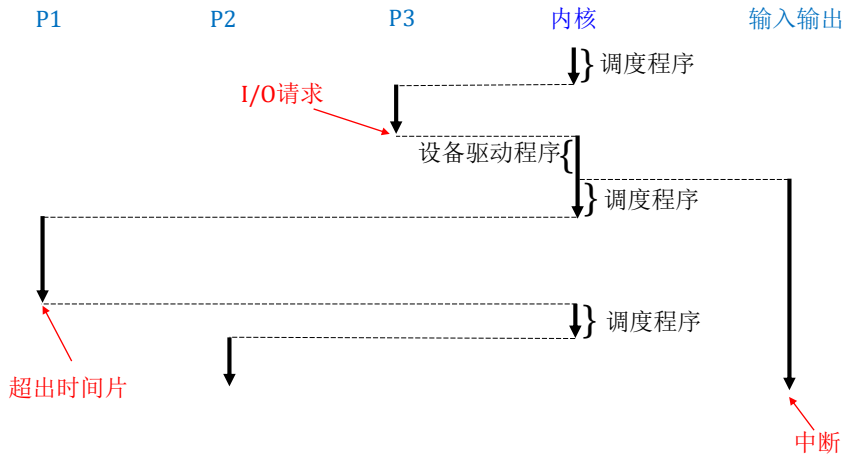
多道程序批处理系统

- 多道程序设计示例–时间序列图。



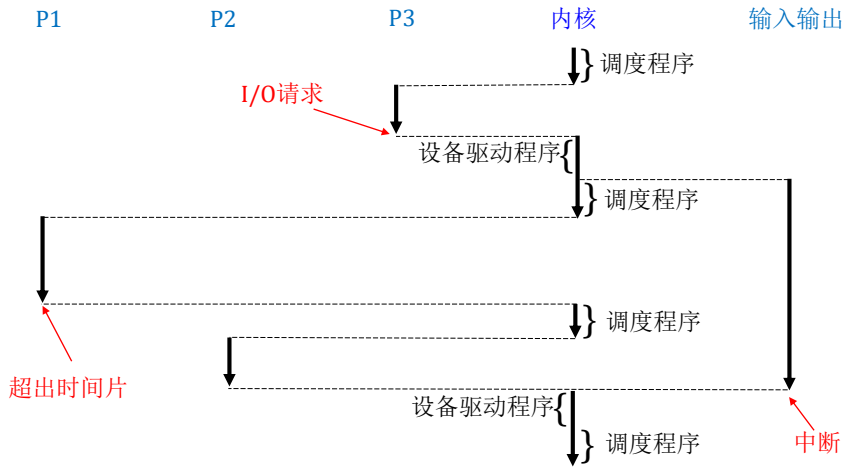
多道程序批处理系统

- 多道程序设计示例–时间序列图。



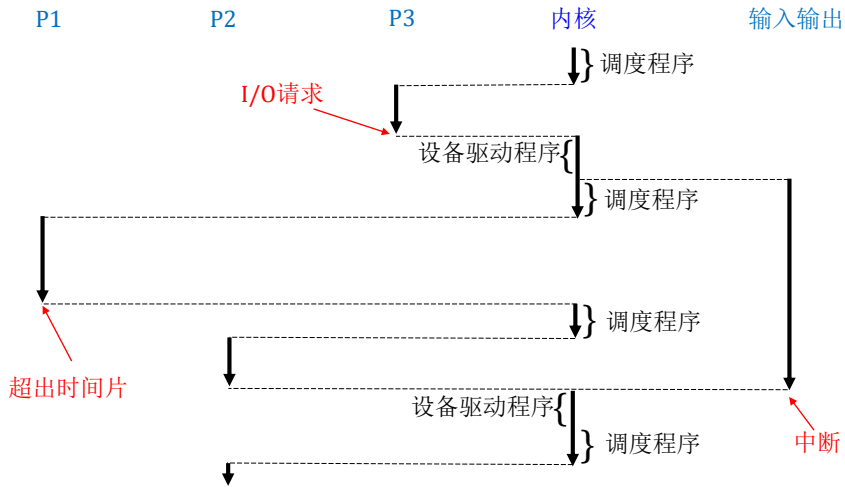
多道程序批处理系统

- 多道程序设计示例–时间序列图。



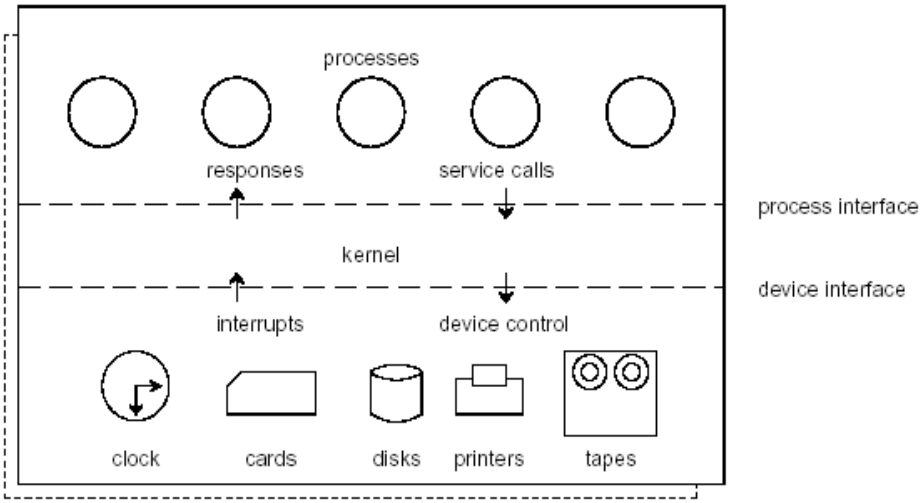
多道程序批处理系统

- 多道程序设计示例–时间序列图。



多道程序批处理系统

- 多道程序的组成部分



多道程序批处理系统

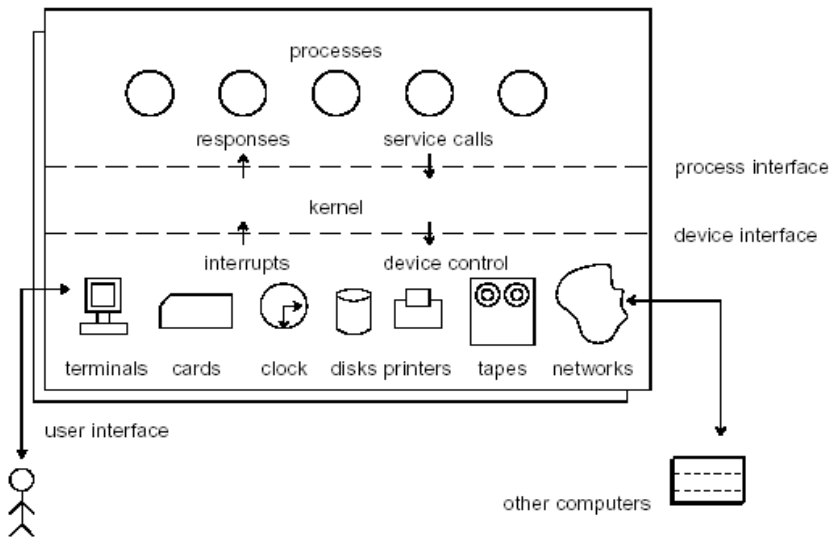
- 多道程序设计的要求
  - 硬件支持
    - I/O中断和DMA控制器
      - 以便在I/O设备忙时执行指令。
    - 计时器中断使CPU获得控制。
    - 内存管理
      - 必须在内存中保留几个准备运行的作业。
    - 内存保护（数据和程序）。
  - 来自操作系统的软件支持
    - 用于调度（下一步运行哪个程序）。
    - 管理资源争用。

分时系统

- 时间共享扩展了批处理多道程序设计，以处理多个交互式作业。
  - 这是交互式多道程序设计。
  - 多个用户通过终端输入的命令同时访问系统。
  - 处理器的时间由多个用户共享。
- 在分时（多任务）中，CPU频繁切换作业，用户可以在作业运行时与作业交互，从而创建交互式计算：
  - *响应时间应小于等于1秒。*
  - 每个用户至少有一个程序（进程）在内存中执行。
  - CPU调度支持多个准备同时运行的作业。
  - 如果进程不适合内存，交换会将它们移入和移出以运行。
  - *虚拟内存允许执行不完全在内存中的进程。*

分时系统

- 分时架构



分时系统

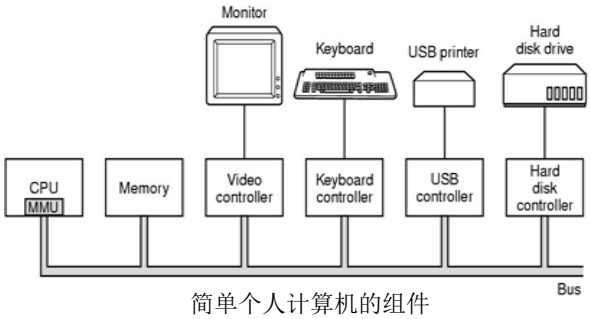
- 为什么分时工作有效？
  - 因为人类反应时间慢
    - 典型用户每分钟需要2秒的处理时间。
    - 然后，许多用户应该能够共享同一个系统，而不会明显延迟计算机的反应时间。
    - 用户应该获得良好的响应时间。
  - 它符合大型计算机安装的经济基础。

实时系统

- 实时（RT）系统是需要遵守截止日期（即时间限制）的专用系统（非通用）。
- *计算的正确性不仅取决于逻辑结果，还取决于结果产生的时间。*
- 硬实时
  - 硬实时系统必须在规定期限内完成。
  - 它与通用操作系统不支持的分时系统冲突。
  - 通常用作专用应用中的控制设备：
    - 工业控制
    - 机器人学
  - 辅助存储器有限或不存在，数据/程序存储在短期存储器或只读存储器（ROM）中。
- 软实时
  - 截止日期是可取的，但不是强制性的
  - 工业控制或机器人技术的有限实用性
  - 适用于需要高级操作系统功能的现代应用（多媒体、视频会议、虚拟现实）

个人/台式计算机

- 个人计算机-专用于单个用户的计算机系统。
- I/O设备
  - 键盘、鼠标、显示屏、小型打印机
- 用户的便利性和响应能力
- 可以采用为更大的操作系统开发的技术；通常个人只使用计算机，不需要高级CPU利用保护功能
- 可以运行几种不同类型的操作系统
  - Windows、MacOS、UNIX、Linux



简单个人计算机的组件

个人/台式计算机

- 办公环境
  - 连接到LAN的PC，连接到大型机或小型计算机的终端，提供批处理和分时服务
  - 允许网络和远程系统访问相同资源的门户
- 家庭网络
  - 单一系统，调制解调器
  - 防火墙，网络化

多处理器系统

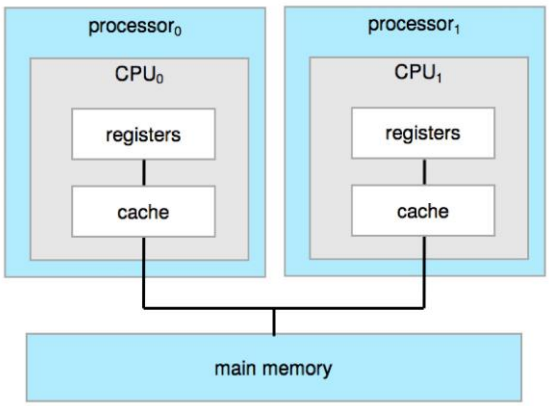
- SISD和MIMD计算机
  - 单指令单数据（SISD）
    - 单个处理器执行单个指令序列，对存储在单个内存中的数据进行操作。
      - 单处理器。
  - 多指令多数据（MIMD）
    - 一组处理器在不同的数据集上同时执行不同的指令序列。
      - 多处理器。

多处理器系统

- 多处理器系统
  - 传统上，多处理器系统有两个（或更多）处理器，每个处理器都有一个单核CPU。
    - 处理器共享计算机总线，有时还共享时钟、内存和外围设备。
    - 通信通常通过共享内存进行。
    - 也称：并行系统，紧耦合系统
  - 多处理器系统的使用和重要性正在增长，在计算领域占据主导地位。
    - 增加吞吐量
    - 规模经济
    - 提高可靠性
      - 故障弱化或容错
  - N个处理器的加速比小于N。
    - 是什么降低了额外处理器的预期增益
      - 多处理器协作中的开销
      - 争夺共享资源

- **多处理器系统**
  - 非对称和对称多处理模型
    - 非对称多处理
      - 主处理器调度特定的工作并将其分配给从处理器。
    - 对称多处理（SMP）
      - SMP是最常见的体系结构。
        - 每个对等CPU处理器都有自己的寄存器集。
        - 所有处理器通过系统总线共享物理内存。
        - 每个处理器运行一个相同的操作系统副本，并从可用进程池中进行自调度。
        - 多处理器的存在对用户是透明的。
      - 平衡：操作系统跨所有处理器调度进程/线程，降低处理器之间的工作负载差异。
        - 多个进程可以同时运行，而不会导致性能显著下降。
      - 健壮性：单CPU故障不会停止系统，只会降低性能。
      - 增量增长：只需添加另一个CPU！

- **多处理器系统**
  - 对称多处理（SMP）



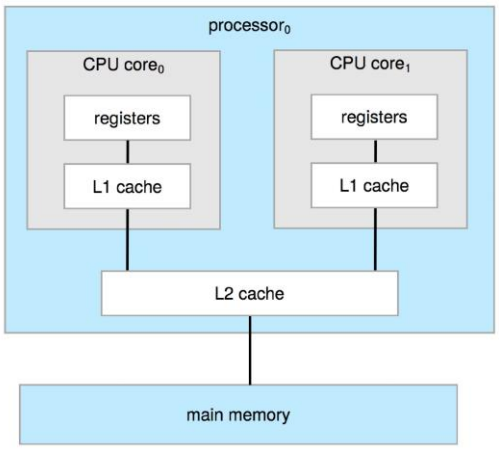
对称多处理体系结构

多处理器系统

- 多核系统
  - 多处理器的定义现在包括多核系统，其中多个计算核驻留在一个芯片上。
  - 单个芯片上的多核比每个芯片上有一个单核的多个芯片效率更高。
    - 片上通信速度更快。
    - 需要更少的电力。这对于移动设备和笔记本电脑来说都是一个重要问题。
  - 下一张幻灯片中的图说明了双核设计，在同一处理器芯片上有两个核，每个核都有自己的寄存器集和本地缓存（L1）。二级缓存是芯片的本地缓存，但由两个处理核心共享。
    - 较低级别的本地缓存通常比较高级别的共享缓存更小、更快。
    - 在操作系统中，具有N个核的多核处理器被视为N个标准CPU。
  - 几乎所有现代操作系统都支持多核SMP。
    - 例如，Windows、macOS、Linux、Android、iOS

多处理器系统

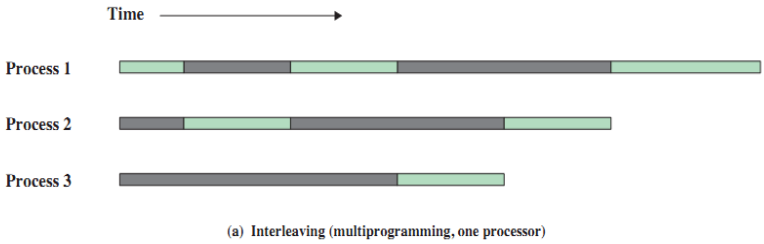
- 多核系统。



一种双核设计，在同一芯片上有两个核

多处理器系统

- 多道程序与多道处理



(a) Interleaving (multiprogramming, one processor)

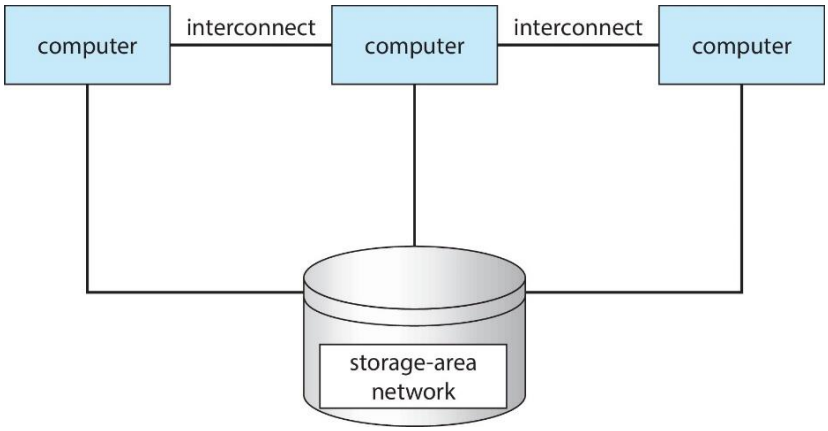
(b) Interleaving and overlapping (multiprocessing; two processors)

集群系统

- 集群系统是另一种将多个CPU聚集在一起的多处理器系统。它由两个或多个单独的多核系统组成
  - 考虑松耦合系统
- 群集允许单个系统共享外部存储并平衡CPU负载：
  - 处理器也有自己的外部存储器。
  - 通信通过高速通道进行。
  - 通常通过存储区域网络（SAN）共享存储
  - 提供高可用性服务，可在故障中生存
    - 优雅降级或容错
- 不对称和对称聚类
  - 非对称群集有一台机器处于热备用模式。
  - 对称集群有多个节点运行应用程序，相互监控。
- 一些集群用于高性能计算（HPC），其中必须编写应用程序才能使用并行化。

集群系统

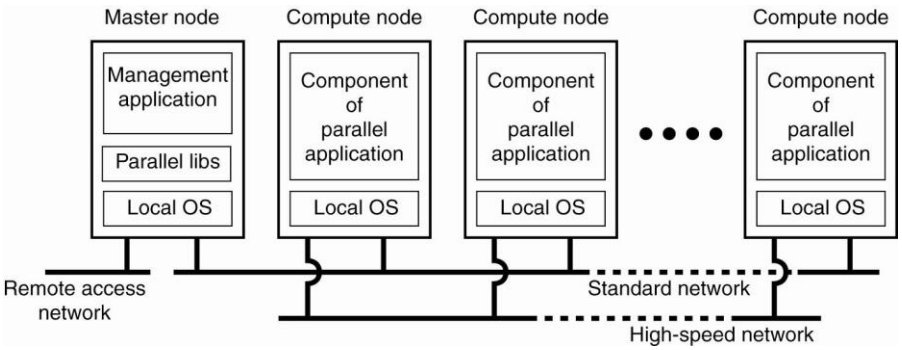
- 集群系统的体系结构



集群系统的一般结构

集群系统

- 集群系统的布局

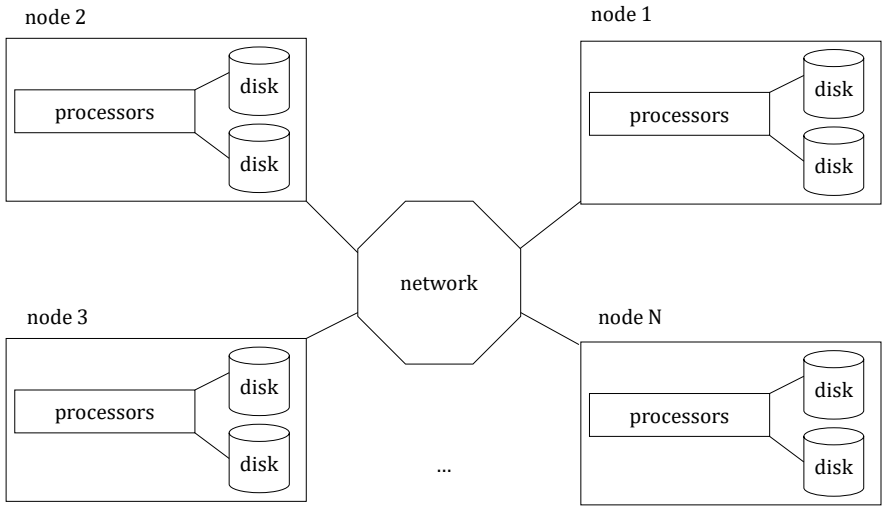


网络系统

- 网络系统在几个物理处理器之间分配资源和计算。
  - 需要网络基础设施
  - 大多数是局域网（LAN）或广域网（WAN）。
  - 也是松耦合系统
    - 每个处理器都有自己的本地内存。
    - 处理器通过各种通信线路相互通信。
- 优势：
  - 资源共享
  - 计算速度-负载共享
  - 可靠性
- 可以是集中式服务器、客户端/服务器或对等（P2P）系统

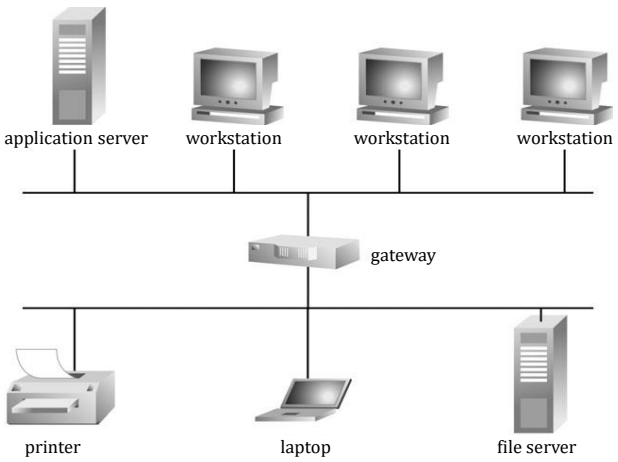
网络系统

- 网络系统的布局



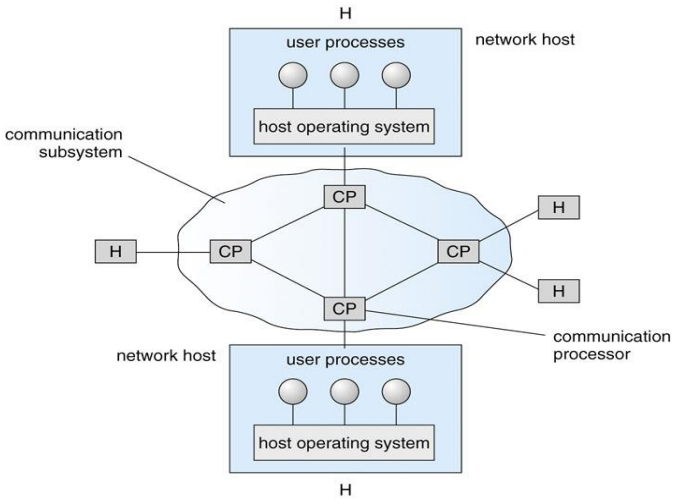
- **网络系统**
  - 局域网（LAN）
    - 设计用于覆盖小地理区域的局域网
      - 多址总线、环形或星形网络
        - 多路总线、环形网络或星型网络
      - *速度：10-100兆位/秒
      - 广播既快又便宜
      - 节点：
        - 通常是工作站和/或个人计算机
        - 几个（通常是一个或两个）大型机

- **网络系统**
  - 局域网（LAN）
    - 设计用于覆盖小地理区域的局域网。



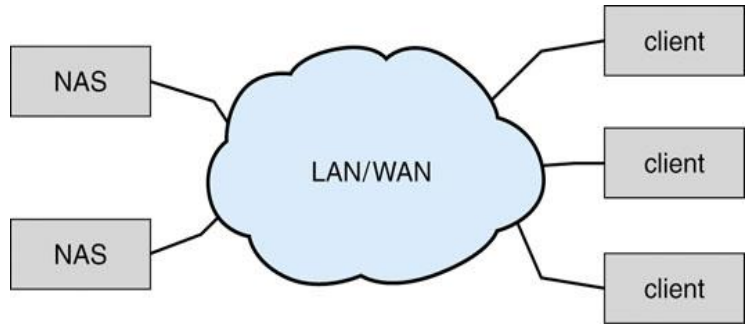
- **网络系统**
  - 广域网（WAN）
    - 链接地理位置不同的站点
    - 长途线路上的点对点连接
      - 例如，从一家电话公司租来的
    - *速度：1.544-45兆位/秒。
    - 广播通常需要多条消息
    - 节点：
      - 通常大型机的比例很高

- **网络系统**
  - 广域网（WAN）
    - 链接地理位置不同的站点



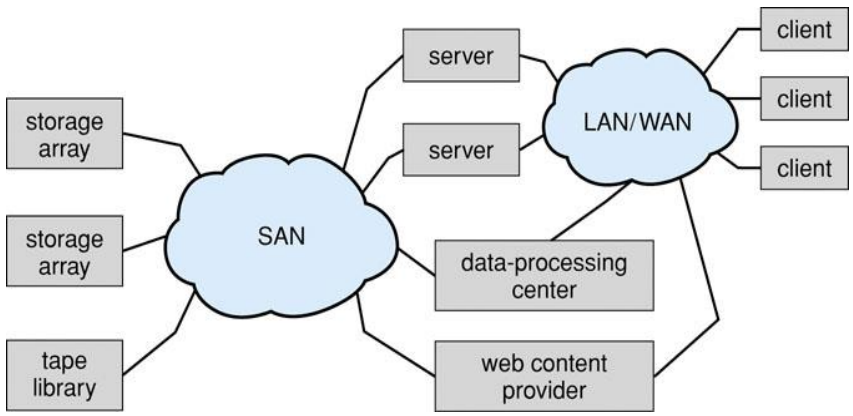
网络系统

- 网络附加存储（NAS，Network-attached Storage）

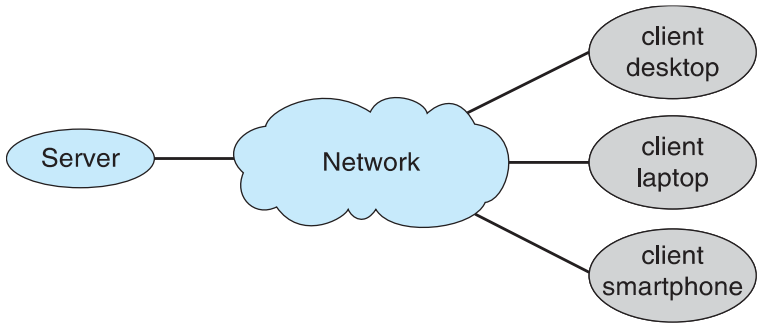


网络系统

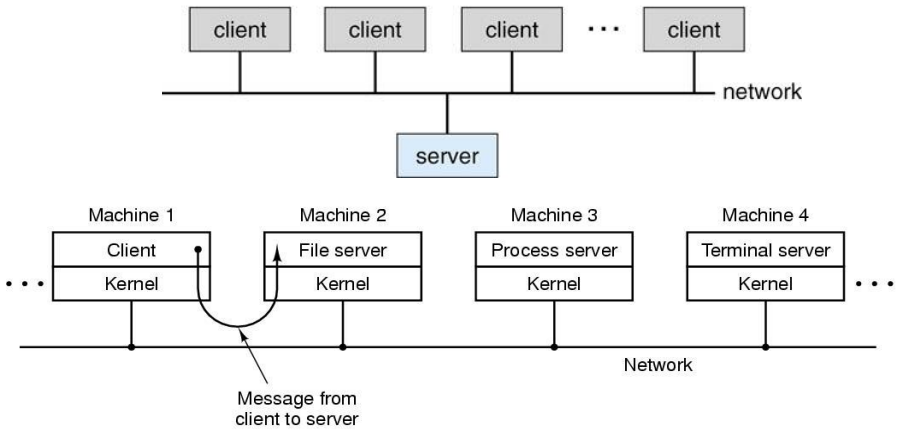
- 存储区域网络（SAN）



- **网络系统**
  - 客户机/服务器计算
    - 哑终端被智能个人计算机取代
    - 服务器响应客户端生成的请求
      - 计算服务器为客户端请求服务提供接口
        - 例如，数据库访问
      - 文件服务器为客户端提供存储和检索文件的接口



- **网络系统**
  - 客户机/服务器计算
    - 客户机/服务器系统的布局



- **网络系统**
  - 对等（P2P）系统
    - 对等网络不区分客户端和服务器
      - 所有节点都被视为对等节点
      - 每个节点可以充当客户端、服务器或两者
    - 加入P2P网络的节点必须：
      - 向网络上的中央查找服务注册其服务，或
      - 广播服务请求并通过发现协议响应服务请求。
    - 例子
      - Napster
      - Gnutella
      - 像Skype这样的VoIP

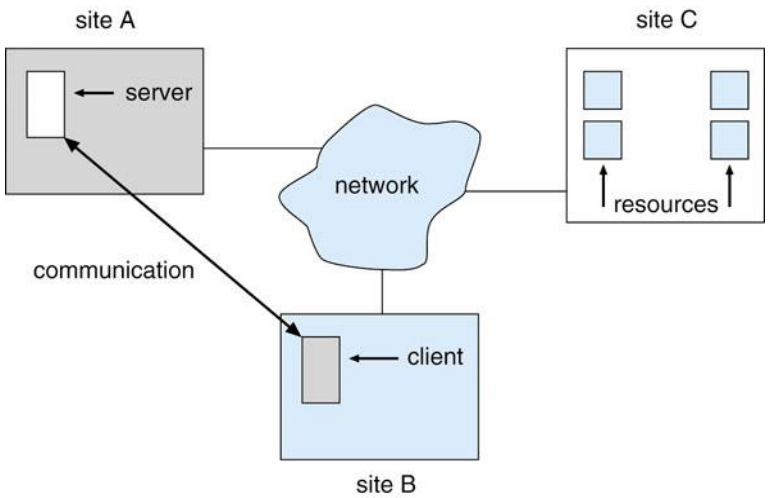
- **网络系统**
  - 网络操作系统（NOS）
    - 每台计算机都独立于网络上的其他计算机运行
      - 主要提供文件共享
    - 用户需要考虑到机器的多样性
    - 通过以下方式访问指定的远程机器的资源：
      - 远程登录到相应的远程计算机
        - Telnet，SSH。
      - 远程桌面
        - 微软视窗
      - 将数据从远程计算机传输到本地计算机
        - 文件传输协议（FTP）

分布式系统

- 分布式系统是由通信网络互连的松散耦合处理器的集合
  - 处理器被称为节点、计算机、机器和主机。
- 分布式系统的原因：
  - 资源共享
    - 在远程站点共享和打印文件
    - 在分布式数据库中处理信息
    - 使用远程专用硬件设备
  - 计算加速
    - 负荷分担
  - 可靠性
    - 检测站点故障并从中恢复，功能转移，重新整合故障站点
  - 表达
    - 消息传递

分布式系统

- 分布式系统的布局



分布式系统

- 分布式操作系统（DOS）
  - DOS给用户的印象是，只有一个操作系统控制着网络。
    - 网络基本上是透明的。
    - 这是一个强大的虚拟机。
  - 用户不知道机器的多样性。
    - 与NOS不同
    - 对远程资源的访问类似于对本地资源的访问
  - 数据迁移
    - 通过传输整个文件或仅传输即时任务所需的文件部分来传输数据
  - 计算迁移
    - 在整个系统中传输计算，而不是数据
  - 进程迁移
    - 在不同的站点执行整个流程或其中的一部分

基于Web的系统

- 网络已经变得无处不在.
- 个人电脑是最流行的设备。
- 更多设备联网以允许Web访问
- 用于管理类似服务器间Web流量的新设备类别
  - 负载平衡器
- 网格/云计算的基础

基于Web的系统

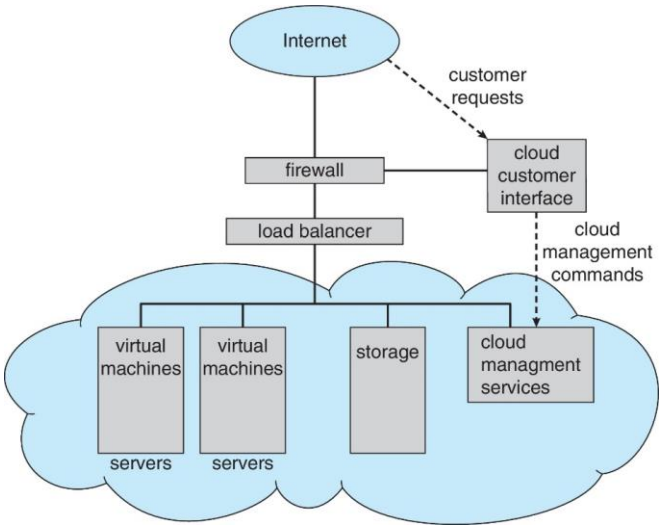
- 网格计算系统
  - 计算机资源的集合，通常由多方拥有，位于多个位置，连接在一起，以便用户可以共享访问，达成组合优势：
    - 可以轻松跨越广域网
    - 多样化环境
    - 跨越行政/地理界限
    - 支持虚拟组织（VO）
  - 例子：
    - EGEE埃吉
      - 为电子科学启用网格（欧洲）
    - OSG
      - 开放科学网格（美国）

基于Web的系统

- 云计算系统
  - 计算机资源的集合，通常由单个实体拥有，连接在一起，以便用户可以租用其组合资源的一部分
  - 云计算系统的特点：
    - 位置独立性
      - 用户可以使用带有任何（支持的）系统的任何设备从任何地方访问所需的服务。
    - 成本效益
      - 整个基础设施归供应商所有，无需由用户提供基础投资
    - 可靠性
      - 通过多个冗余站点进行增强，然而也会出现用户无法补救的服务中断状态
    - 可伸缩性
      - 用户需求可以根据需求与可用资源进行定制——成本效益显而易见

- **基于Web的系统**
  - 云计算系统
    - 云计算系统的特点：（续）
      - 安全
        - 尽管需要解决敏感数据控制方面的问题，但由于集中化，数据丢失的风险较低。
      - 易用
        - 用户通常不需要做太多的部署或定制工作，因为提供的服务容易获得且随时可用
    - 例子
      - Amazon EC2（弹性计算云）
      - 谷歌应用引擎
      - IBM企业数据中心
      - 微软微软Azure
      - SUN云计算
      - 国内：阿里云、腾讯云、华为云……

- **基于Web的系统**
  - 云计算系统



手持系统和移动系统

- 手持系统
  - 手持系统也专用于：
    - 个人数字助理（PDA）
    - 移动电话
  - 问题
    - 有限内存
    - 慢处理器
    - 小显示屏
    - 支持多媒体（音频、视频、图像）
- 移动系统
  - 手持智能手机、平板电脑等
  - 使用IEEE 802.11无线或蜂窝数据网络进行连接
  - 额外功能
    - 更多操作系统功能，如GPS、陀螺仪
    - 允许新类型的应用程序，如增强现实
  - iOS、Android、鸿蒙OS是主流发展趋势