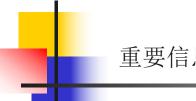
## 操作系统 Operating System

中国传媒大学计算机学院





## ■ 教材与参考资料

- 《计算机操作系统》第四版, 汤小丹,梁红兵,哲凤屏,汤子瀛
- 谌卫军等,操作系统,清华大学出版社
- William Stallings著,陈向群等译。操作系统—精髓与设计原理, 电子工业出版社

## ■ 从本课中可以学到什么?

- 掌握OS的工作原理
- 理解OS设计思想、方法、策略
- 加深对使用的OS的理解,有利于深入编程
- 思维方式和计算机学科方法论的学习





- 课程特点
  - ■理论性、实践性和综合性
- 学习要求/方法
  - ■认真听讲
  - 完成作业与实验
  - ■阅读教材和有关资料
  - 思考,讨论

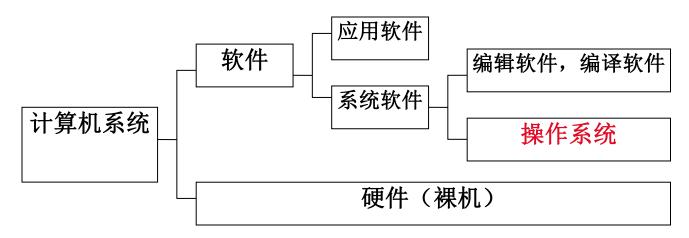


## ■ 第一章 序论

- ■本章内容
  - 操作系统的目标和作用
  - 操作系统的发展过程
  - 操作系统的基本特性
  - 操作系统的主要功能
  - 操作系统的结构设计
  - 操作系统的硬件环境



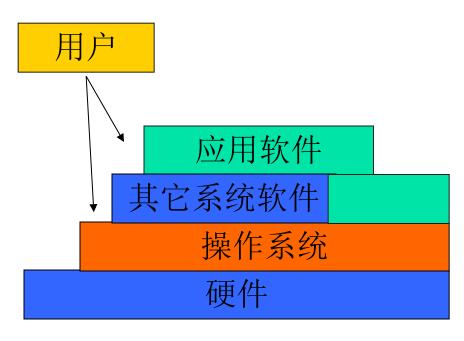
## 计算机系统的组成:







- 计算机系统由硬件和 软件组成
- 操作系统在硬件基础 上的第一层软件
- 是其他软件和硬件之 间的接口



计算机系统的层次结构



## ■配置OS的目标

- 方便用户使用
  - 软件与硬件之间的接口,将裸机改造成一台功能更强、服务质量更高、用户更方便使用、安全可靠的虚拟机。
- ■有效性
  - 提高系统的资源利用率和吞吐量
- ■可扩展性
  - 方便地增加新的功能和模块,并修改原有的功能和模块,适应硬件和体系结构的发展和功能需求的变化
- 开放性
  - 使应用在不同的系统中具有可移植性和互操作性,不同系统可以 通过网络有效集成.要求系统遵循标准规范

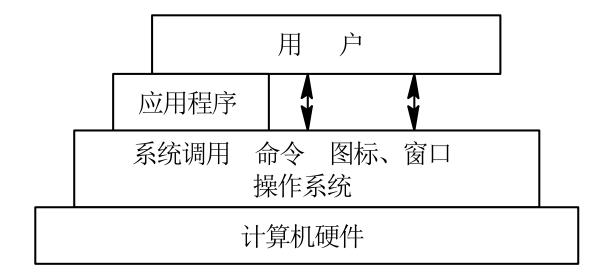


## ■ 操作系统的定义

- 操作系统是管理系统资源、控制程序执行,改善人机界面、提供各种服务、合理组织计算机工作流程和为用户使用计算机提供良好运行环境的一种系统软件。
- 有效地控制和管理计算机软硬件资源,合理地组织计算机工作流程,以提高资源利用率,并为用户提供强有力的使用功能和灵活方便的使用环境的程序集合。
- ■操作系统是一组控制和管理计算机硬件和软件资源,合理地对各类作业进行调度,以方便用户使用的程序的集合。



- 1 OS作为用户与计算机硬件系统之间 的接口
  - 用户通过OS利用计算机软硬件资源运行 自己的程序解决问题:
  - 命令方式、系统调用、GUI





- 2 OS作为计算机系统资源的管理者
  - 硬件资源: 处理器、存储器、 I/O设备
  - 软件资源: 数据,程序
  - 处理机管理: 用于分配和控制处理机;
  - 存储器管理: 主要负责内存的分配与回收;
  - I/O设备管理: 负责I/O设备的分配与操纵;
  - 文件管理: 负责文件的存取、共享和保护



## ■ 3 OS对计算机资源的抽象

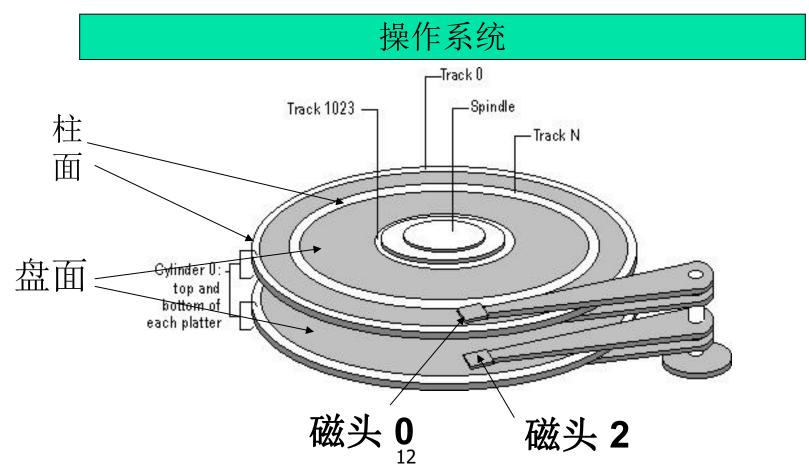
- 裸机: 直接面对硬件
- 裸机+管理软件 虚拟机
- ■用户所看到的机器,是一台比裸机功能更强、使用更方便的机器。
- 操作系统的层次结构

虚 ・子符命令 ・ と と と と と と と と と と と と と と と と と と	
机 条件系统 •系统调用	
裸机 •机器指令	<b>♦</b>



文件

种抽象





- 不断提高计算机资源利用率
- 方便用户
- *器件的不断更新换代*
- 计算机体系结构的不断发展
- 新用户,新的服务、工具和功能



## ■ 第一章 序论

- ■本章内容
  - 操作系统的目标和作用
  - 操作系统的发展过程
  - 操作系统的基本特性
  - 操作系统的主要功能
  - 操作系统的结构设计
  - 操作系统的硬件环境



## ■ 一 无操作系统的计算机系统

1945-1955, 真空管, 纸带、卡片机)

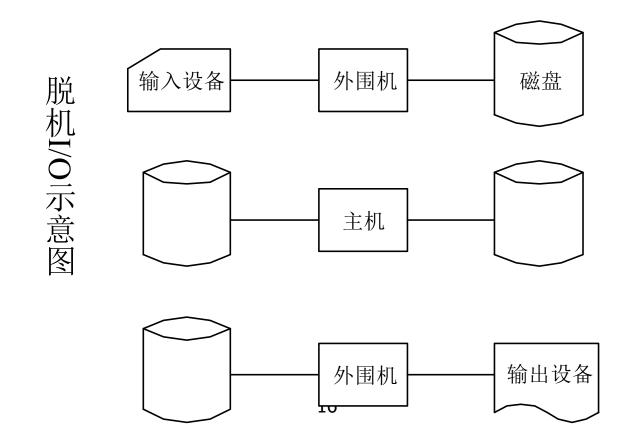
- 机器语言、通过控制台控制;用户独占资源
- 人工操作方式
  - 手工操作时间长; CPU等待人工操作
- 脱机输入输出方式
  - 外围机控制把卡片上的数据读到磁带上,计算机需要数据时从磁带读入。先输出到磁带,然后外围处理机从磁带输出到输出设备。

## 4

## 脱机输入/输出(Off-Line I/O)方式

这种脱机I/O方式的主要优点如下:

- ((1))减少了CPU的空闲时间。
- (2) 提高I/O速度。





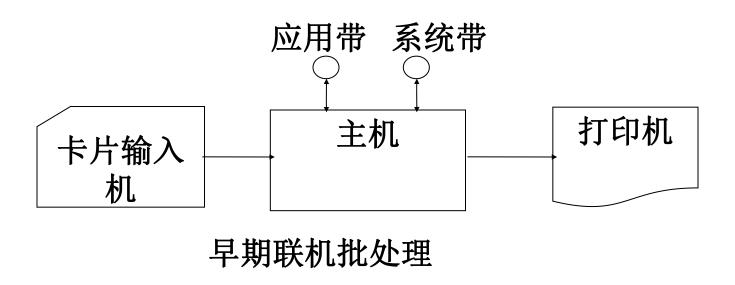
## ■ 二 (单道)批处理系统

- 1955—1965, 晶体管, 穿孔纸带提交, 磁带输入
- 高级语言、科学计算, 监控程序
- 作业控制
  - 用户提交作业(纸带)
  - 操作员把一批作业存入磁带
  - 常驻**监控程序**自动地从磁带装入程序、运行、撤出作业
- 自动性, 顺序性, 单道性
- 联机批处理(CPU直接处理输入输出)
- 脱机批处理(专门的卫星机负责输入输出)
  - 实现了作业的自动转接,但**I/O**设备与主机仍然式串行工作。

## 批处理方式

#### (1) 联机批处理

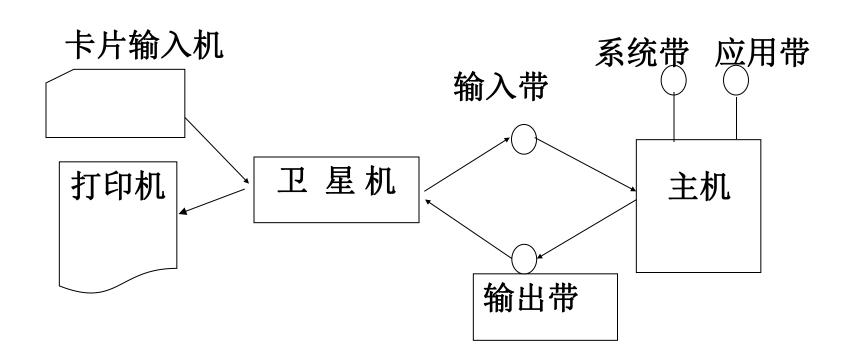
早期的联机批处理的硬件控制方式是:作业的输入、计算和输出都是在CPU直接控制下进行的。 模型:

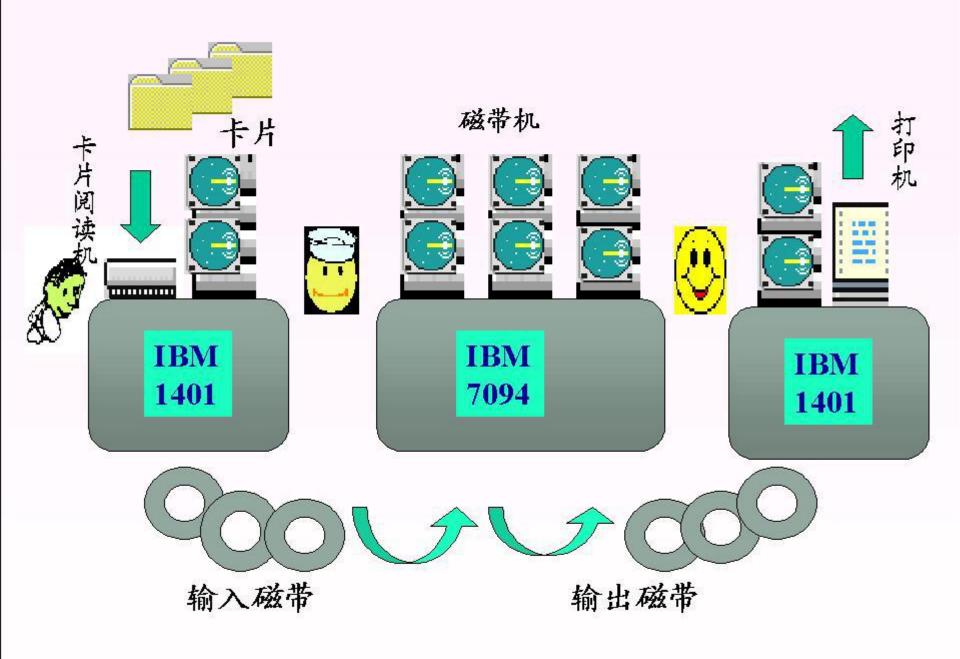


问题:输入输出时,CPU处于等待状态。

## 2) 脱机批处理(缓冲技术的一种)

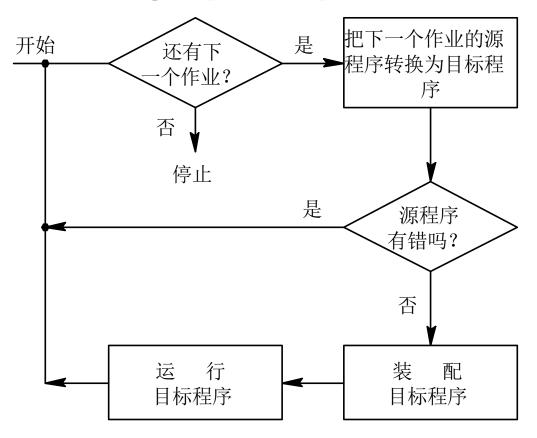
增加卫星机完成输入/输出功能。主机与卫星 机可并行工作。





# 

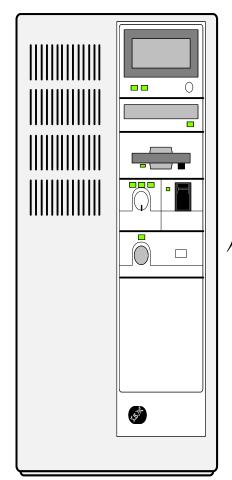
## 单道批处理系统(Simple Batch Processing System)的处理过程





## 三集成电路时代的操作系统

- *计算机有大、中、小 等机型*
- 操作系统
  - ■多道批处理系统
  - ■分时系统
  - ■实时系统



小型机



## 作业1

作业2

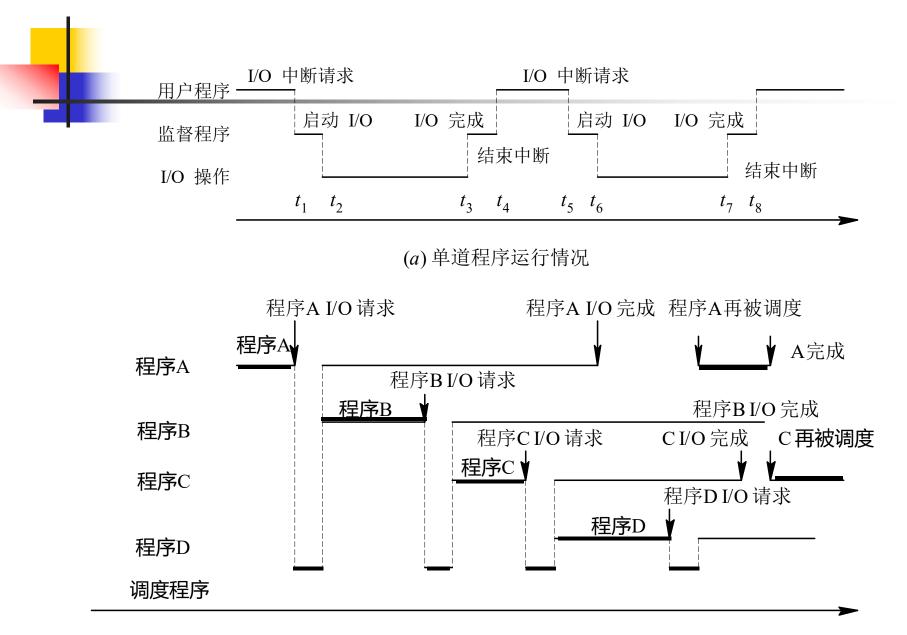
作业3

操作系统

内存驻 留多道 作业

#### 多道批处理系统

- 1965-1980,集成电路
- 多道程序
  - 多道程序设计(multiprogramming)是指允许多个程序(作业)同时进入一个计算机系统的内存储器并启动进行交替计算的方法。也就是说,计算机内存中同时存放了多道(二个以上相互独立的)程序,它们均处于开始和结束点之间。
  - 从宏观上看是并行的,多道程序都处于运行过程中,但都未运行结束;从 微观上看是串行的,各道程序轮流占用CPU,交替地执行。
  - 引入多道程序设计技术的根本目的是提高CPU 的利用率,充分发挥计算机系统部件的并行性
- 多道批处理系统
  - 用户所提交的作业都先存放在外存上并排成一个队列,称为"后备队列",然后,由作业调度程序按一定的算法从后备队列中选择若干个作业调入内存,使它们共享CPU和系统中的各种资源。
- 建立在中断和通道提供的硬件基础之上的 CPU和I/O设备的并行
- 提高CPU、内存、I/O设备利用率
- 资源利用率高,作业吞吐量大。
- 缺点是作业周转时间长,不能交互计算,调试程序困难。
- 应用
  - 计算、商业。IBM360



(b) 四道程序运行情况 单道和多道程序运行情况



## 多道批处理系统需要解决的问题

- ((1))处理机管理问题。
- (2) 内存管理问题。
- (3) I/O设备管理问题。
- (4) 文件管理问题。
- (5)作业管理问题。
- (6) 用户接口问题。

多道程序设计系统的出现标志操作系统的形成

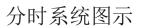


## ■分时操作系统

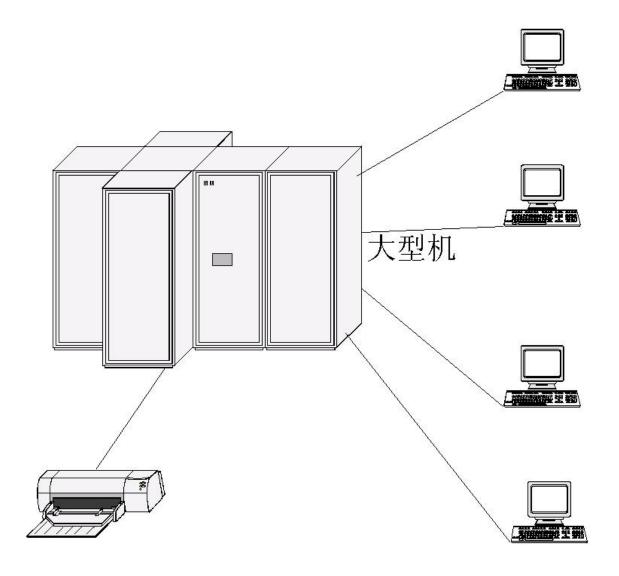
- 批处理系统中用户不能干预程序运行,引进 Time-sharing Operating System
- 工作方式
  - ■用户在终端控制程序的运行;
  - 把计算机的系统资源(尤其是CPU时间) 进行时间上的分割,每个时间段称为一个 时间片(time slice),每个用户依次轮流 使用时间片。
  - 时间片的划分是影响系统的关键因素之一

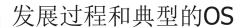
时间片(如0.1秒)  $ag{ ag{B}} B C D A B C D A \dots$ 

时间t











#### ■ 分时操作系统

- 特点:
  - 多路性;多个用户同时工作。共享系统资源,提高了资源利用率。促进了计算机的普遍应用,提高资源利用率:远地用户通过终端(较便宜)联机使。
  - 独立性;各用户独立操作,互不干扰。
  - 及时性;系统能及时对用户的操作进行响应,显著提高调试和修改程序的效率:缩短了周转时间
  - 交互性: 支持人机对话, 交互式应用。
- 公共计算服务系统: MULTICS (MULTtiplexed Information and Computing ):BELL,MIT,GE
- Unix





#### ■ 批处理 VS 分时

- 目标不同:资源利用率,及时反应
- 适用作业类型不同
- 资源利用率不同
- 作业控制方式

#### ■ 分时和批处理相结合

- 分时优先,任务分为前台、后台
- "前台": 需频繁交互的作业; "后台": 时间性要求不强的作业。



## 实时系统

所谓"实时",是表示"及时",而实时系统(Real-Time System)是指系统能及时(或即时)响应外部事件的请求,在规定的时间内完成对该事件的处理,并控制所有实时任务协调一致地运行。

#### 1. 应用需求

- ((1))实时控制。
- (2) 实时信息处理。



## 2. 实时任务

- 1) 按任务执行时是否呈现周期性来划分蕌
- ((1))周期性实时任务。
- (2) 非周期性实时任务。

外部设备所发出的激励信号并无明显的周期性,但都必须联系着一个截止时间(Deadline)。它又可分为:

- ① 开始截止时间——任务在某时间以前必须开始执行;
- ② 完成截止时间——任务在某时间以前必须完成。



- 2) 根据对截止时间的要求来划分蕌
- (1) 硬实时任务(hard real-time task)。系统必须满足任务对截止时间的要求,否则可能出现难以预测的结果。
- (2) 软实时任务(Soft real-time task)。它也联系着一个截止时间,但并不严格,若偶尔错过了任务的截止时间,对系统产生的影响也不会太大。





- 四 微机,网络
  - **1980**,
    - 单用户单任务->单用户多任务->多用户多任务;
    - 界面友好; 软件丰富
  - 网络操作系统
    - 统一管理网络中共享资源的使用和协调处理任务。
    - 功能: 网络通讯、提供网络接口和提高网络服务质量。



## 其它OS

- 分布式操作系统(Distributed Operating System)
  - 多台分散计算机通过网络连接成一个系统,其中的每台机器 既自治又相互协同完成任务
  - 特征
    - 1 是一个统一的操作系统,资源进一步共享
    - 2 透明性
    - **3** 自治性: 处于分布式系统的多个主机处于平等地位,无主 从关系
  - 优点:处理能力增强、速度更快、可靠性增强
  - 分布式计算应用
    - SETI@HOME。该项目在1999年初开始将分布于世界各地的 200万台个人电脑组成计算机阵列,搜索射电天文望远镜信 号中的外星文明迹象。项目组称,在不到两年的时间里,这 种计算方法已经完成了单台计算机345000年的计算量。



## ■ 其它OS (续)

- ■嵌入式操作系统
  - ■嵌入式系统
    - 嵌入式(计算机)系统嵌入在各种设备、装置或系统中, 完成特定功能的软硬件系统,它是某一设备、装置或 系统中的一部分,这个设备、装置或系统可以不是 "计算机"。

#### ■嵌入式OS

- 嵌入式操作系统,是运行在嵌入式智能芯片环境中,对整个智能芯片以及它所操作、控制的各种部件装置等等资源进行统一协调、调度、指挥和控制的系统软件
- 受硬件平台的限制;非通用微型化;实时性、可靠性;可定制、易移植;成本低占用资源少等特点。





#### 个人机操作系统

#### DOS

- 1981年: PC-DOS 1.1: IBM PC, 只支持软盘的个人操作系统;
- DOS BIOS(Basic Input/Output System): 由一组与硬件相关的设备驱动程序组成,实现基本的输入/输出功能;
- DOS核心:提供一套独立于硬件的系统功能:内存管理、文件管理、字符设备和输入/输出、实时时钟等;
- 命令处理程序: 对用户命令进行分析和执行;





#### ■ 个人机操作系统(续)

- DOS的特点
  - 字符用户界面;作业管理: 命令行, 批处理程序(BAT 文件)
  - 编程时通过软中断调用(int 21h)来使用系统功能
  - 不支持虚拟存储,没有存储保护。采用段式分配(内存块),可直接访问的最大地址空间为1MB。其余的内存只能通过作为扩展内存(XMS)或扩充内存(EMS)来使用。
  - 文件系统为FAT(File Allocation Table)格式(磁盘卷, 多级目录,文件名 8+3 个字符;分区容量最大为 2GB);有文件属性,没有区分用户的访问权限保护。
  - 设备驱动程序在系统起动时加载。





#### ■ 个人机操作系统(续)

- Windows
  - 1990年, Windows 3.1 95 (脱离dos, 32位, P&P, 多媒体, 内置网络) NT (商用版, SMP, 多线程, 容错, 集成网络计算) 2000, XP (家用和商用合而为一)
  - 支持对称多处理机
  - 真正的32位操作系统;
  - 完全的代码可重入(reentrant): 同一段代码可由多个应用同时访问;
  - 图形用户界面GUI(和字符用户界面)。
  - 抢先式多任务和多线程。支持动态链接。
  - 虚拟存储:段页式(有存储保护)。
  - 兼容16位Windows应用:
  - 文件系统: NTFS(HPFS), 支持安全控制
  - 设备驱动程序: VxD(virtual driver)。
  - 可移植: 适用于多种硬件平台。
  - 容错能力。
  - 面向对象特性: 用对象来表示所有资源。



#### ■ 个人机操作系统(续)

- Mac OS
  - GUI(施乐公司Palo Alto研究中心)

#### Linux

- Free software 遵循GPL(General public license)规则。使用的自由,获得源程序自由,自己修改自由,有复制和推广的自由,收费的自由。得到MIT支持
- UNIX的PC机实现, Linus Torvalds





#### ■ 中小型机操作系统

#### Unix

- UNIX是现代操作系统的代表。 安全性、可靠性以及强大的计算能力。
- 通用、交互型的分时操作系统,开放型,公开源代码。
- 1965年: MIT的Multics,由于规模和进展而没有达到目标;
- 1969年: AT&T, PDP-11上的16位操作系统;
- 1974年: UNIX系统正式发表(第五版), 在大学得到使用和好评;
- 1980年: University of California at Berkeley为VAX11发表BSD4.0;
  以后,UNIX就以AT&T和Berkeley为主分别开发,有多种变种;
- 1989年: UI (UNIX International)发表UNIX system V Res4.0; 使 BSD和System V在用户界面上统一;
- 1991年芬兰大学生Linus Benedict Torralds开发了第一个Linux版本。
- 最早由BELL Lab, AT&T, 一种标准而不是一个系统。IBM (AIX),SUN(Solaris),BSD,SGI(Irix)



#### ■嵌入式操作系统

#### VxWorks

- 通用性: 支持各种工业标准
- 实时性:系统的核心是一高效率的微内核,微内核支持 各种实时功能,包括快速多任务处理,中断支持,抢 占式和轮转式调度
- 高效性: 微内核设计减轻了系统负载并可快速响应外 部事件
- 可裁剪性:从只需几千字节存储器的深嵌式产品设计 到复杂高端实时系统设计,开发人员有八十多个选件 并可构成上百个不同的配置



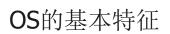
#### ■嵌入式操作系统

- Windows CE
  - C代表袖珍(Compact)、消费(Consumer)
  - 用于手持设备和信息家电。可以看作是Windows 95的 简化版。
- 掌上电脑操作系统
  - Palm , PocketPC, windows CE, Linux
- 手机操作系统
  - iOS, Android, Symbian, Windows phone
- ■通讯、视频处理专用操作系统等。



## ■ 第一章 序论

- ■本章内容
  - 操作系统的目标和作用
  - 操作系统的发展过程
  - 操作系统的基本特性
  - 操作系统的主要功能
  - 操作系统的结构设计
  - 操作系统的硬件环境





### ■ 并发性(Concurrence)

- 并发性是指两个或两个以上的事件或活动在 同一时间间隔内发生。
- 操作系统的并发性指它应该具有处理和调度 多个程序同时执行的能力。
- 并发与并行(parallel)
- 进程与线程的引入



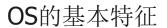
### 共享(Sharing)

- 在系统中的资源可供内存中多个并发执行的进程 (线程)共同使用。
- 由于资源属性的不同,进程对资源共享的方式
  - 互斥共享方式
    - 应规定在一段时间内只允许一个进程(线程)访问该资源(互 斥资源或独占资源)。
    - 打印机
  - 同时访问方式
    - 允许在一段时间内由多个进程"同时"对它们进行访问。这 里所谓的"同时"往往是宏观上的,而在微观上,这些进程 可能是交替地对该资源进行访问。
    - 磁盘、文件等。



## ■ 并发与共享是基本特征

- 互相依存, 互为存在的条件
  - 资源共享是以程序(进程)的并发执行为条件的, 若系统不允许程序并发执行,自然不存在资源共 享问题;
  - 若系统不能对资源共享实施有效管理,协调好 诸进程对共享资源的访问,也必然影响到程序并 发执行的程度,甚至根本无法并发执行。





- 异步性(Asynchronism)
  - 各任务执行进度不可预知。
- 虚拟性 (Virtual)
  - 是把物理上的一个实体变成逻辑上的多个对应物,或把物理上的多个实体变成逻辑上的一个对应物的技术。
  - 时分, 空分



## ■ 第一章 序论

- ■本章内容
  - 操作系统的目标和作用
  - 操作系统的发展过程
  - 操作系统的基本特性
  - 操作系统的主要功能
  - 操作系统的结构设计
  - 操作系统的硬件环境



### ■ 主要任务

- 为多道程序提供良好的运行环境,以保证多道程序高效运行,最大程度利用系统资源,方便用户使用。
- ▶ 为实现上述任务,操作系统应具备的功能包括:

处理机管理功能 文件管理功能 存储器管理功能 设备管理功能 用户接口



- 处理机管理功能
  - 进程控制、进程同步、进程通信、调度
- 存储器管理功能
  - 内存分配、内存保护、地址映射、内存扩充
- 设备管理功能
  - 缓冲、设备分配、设备处理
- 文件管理功能
  - 文件存储空间的管理、目录管理、文件的读/写管理 和保护
- ■用户接口
  - ■命令接口、程序接口



- ■新功能
  - ■安全
  - ■网络功能与服务
  - ■支持多媒体



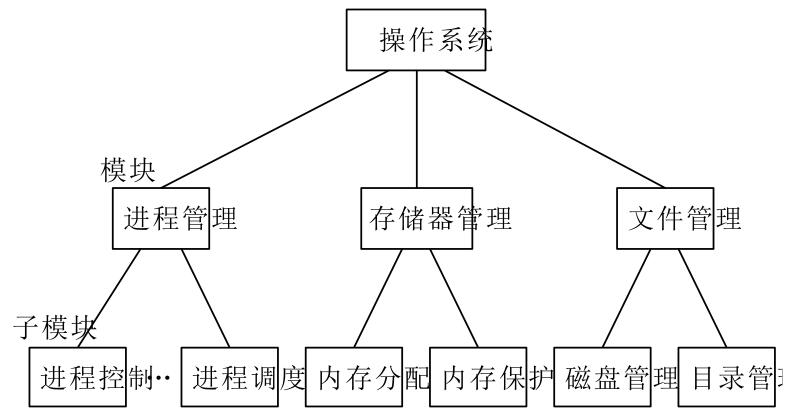
## ■ 第一章 序论

- ■本章内容
  - 操作系统的目标和作用
  - 操作系统的发展过程
  - 操作系统的基本特性
  - 操作系统的主要功能
  - 操作系统的结构设计
  - 操作系统的硬件环境



- 操作系统是一个十分复杂的大型软件。为了控制该软件的复杂性,在开发**OS**时,先后引入了模块化、分层、抽象和隐蔽等方法。开发方法的不断发展,促进了**OS**结构的更新换代。
- 1. 无结构操作系统
- 2. 模块化OS结构
  - 提高了OS设计的正确性、 可理解性和可维护性。
  - 增强了OS的可适应性。
  - 加速了OS的开发过程。
  - 缺点
    - 在开始设计**OS**时,对模块的划分及对接口的规定并不精确, 而且还可能存在错误,因而很难保证按此规定所设计出的模 块会完全正确,满足实际需求;
    - 由于管理上的差异,又会使模块间存在着复杂的依赖关系使 **OS**结构变得不清晰。设计齐头并进,无法做到设计中的每一步都是建立在可靠的基础上。







## ■ 3. 分层式OS结构

■ 从改进设计方式上说,应使我们的每一步设 计都是建立在可靠的基础上。我们可以从物 理机器开始, 在其上面先添加一层具有一 定功能的软件A1, 由于A1是建立在完全确定 的物理机器上的,在经过精心设计和几乎是 穷尽无遗的测试后,可以认为A1是正确的: 然后再在A1上添加一层新软件A2, ...., 如 此一层一层地自底向上增添软件层,每一层 都实现若干功能,最后总能构成一个能满足 需要的OS。



- 3. 分层式OS结构
  - 优点: 易保证系统正确性, 易扩充和易维护
  - 缺点:系统效率会降低



- 4 客户/服务器模式(Client-Server Model)
  - 客户
    - 具有一定的处理能力
    - 向服务器请求服务
  - 服务器
    - 为客户提供服务
  - 通信,网络
  - 交互
    - 客户发送请求消息——服务器接收消息——服务器回送消息—— 客户机接收消息
  - 优点
    - 数据分布存储和处理
    - 便于集中管理
    - 灵活可扩充
  - 缺点
    - 可靠性,瓶颈



- 5 面向对象技术
  - ■可重用性好
  - ■易于修改和扩展
  - ■易于保证系统正确性和可靠性



- 正确性、灵活性、易于维护、可扩充
- ■基本思想
  - 足够小的内核
    - 精心设计的、能实现现代OS核心功能的小型内核
    - 硬件紧密相关、基本功能、客户与服务器通信
  - 基于客户/服务器模式
    - 基本功能放入内核
    - 大部分功能放在微内核外的服务器实现:进程服务器,存储服务器,设备管理服务器
    - 通过微内核提供的消息传递机制实现信息交互
  - 机制与策略分离
    - 实现某一功能的具体执行机构
    - 在机制基础上,借助某些算法实现该功能的优化
  - 采用面向对象技术



- ■微内核的基本功能
  - ■进程管理
    - 微内核: 进程调度
    - 管理服务器: 调度策略
  - 低级存储器管理
    - 微内核: 地址映射
    - 存储管理服务器: 虚拟存储管理
  - ■中断和陷入处理
    - 微内核:响应中断
    - 服务器: 处理中断



- 优点
  - 提高了系统可扩展性
    - 修改和增加服务器扩展系统
  - 增强系统可靠性
    - 微内核的正确性保证
    - 服务器运行在用户态,服务器之间通过消息通信,某服务器的错误不会蔓延
  - ■可移植性
    - 内核负责硬件,服务器硬件无关
    - 移植时修改较少
  - 支持分布式
    - 客户一服务器,服务器一服务器的消息通信机制
    - 进程可以在任何机器上
  - 采用面向对象技术



- 缺点
  - 效率降低
    - 传统os,用户请求服务一般进行两次上下文切换
      - 执行系统调用一由用户态进入系统态,系统完成用户的 请求后,系统态回到用户态
    - 微内核os
      - 1 客户 发消息给 内核 请求某服务器的服务
      - 2 内核 把用户请求服务消息发给 服务器
      - 3 服务器 完成请求 发消息给 内核
      - 4 内核 将消息发给 客户
    - 实际上一项任务需要多个服务器
  - 服务器功能转让内核
    - 减少切换,提高了微内核设计代价



## ■ OS设计原则与设计目标

- ■有效
- 系统的高效率,资源的高利用率
- 公平合理
- ■易于使用和维护升级
- 一一上述目标通常是矛盾的



## ■ 第一章 序论

- ■本章内容
  - 操作系统的目标和作用
  - 操作系统的发展过程
  - 操作系统的基本特性
  - 操作系统的主要功能
  - 操作系统的结构设计
  - 操作系统的硬件环境

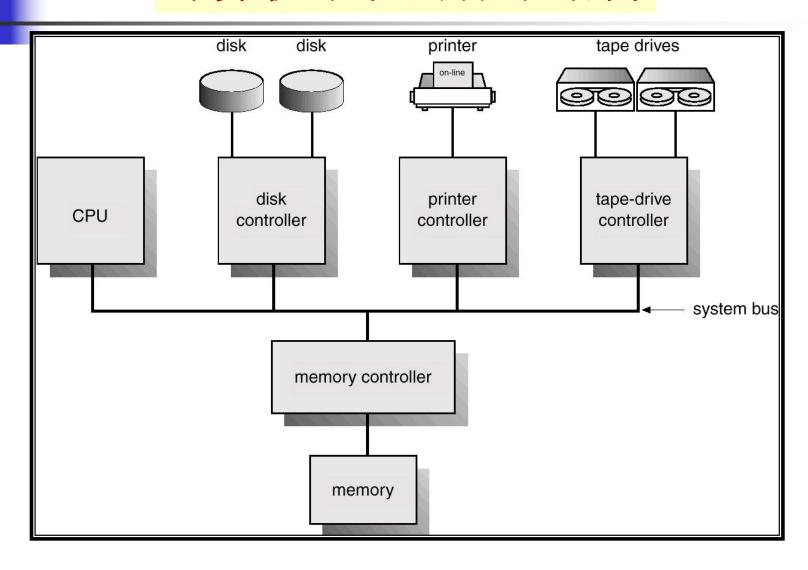


# 操作系统的硬件环境



任何系统软件都是对硬件功能的延伸,都是建立在硬件基础上的,离不开硬件设施的支持。而操作系统更是直接依赖于硬件,与硬件的关系尤为密切。

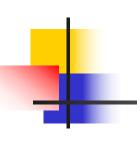
# 计算机系统的体系结构





# OS需要的硬件特性

- 受保护的指令;
- 系统调用;
- 内存保护;
- 中断机制;
- I/O系统;
- 时钟操作。



# 一、受保护的指令(特权指令)

有些指令只有操作系统才有权使用,例如:

- ★ 访问某些硬件资源的指令,这些硬件资源 禁止用户程序直接访问;
- ★ 对I/O设备的直接访问指令,如磁盘、打印机等;
- ◆ 对内存管理状态进行操作的指令(页表指针、刷新TLB等);
- \* 某些特殊的状态位的设置指令;
- \* 停机指令。



根据运行程序对资源和机器指令的使用权限, 把处理器设置为不同状态。

多数系统将处理器工作状态划分为管态和目态。

**管态**:操作系统的管理程序运行时的状态,较高的特权级别,又称为特权态、系统态、内核态

处理器处于管态时:可以执行所有的指令(包括特权指令)、使用所有的资源,并具有改变处理器状态的能力。



**目态:** 用户程序运行时的状态,较低的特权级别, 又称为普通态(普态)、用户态。

在此状态下禁止使用特权指令,不能直接使用系统资源与改变CPU状态,并且只能访问用户程序所在的存储空间。

有些系统将处理器状态划分核心状态,管理状态和 用户程序状态(目标状态)三种。



- 386、486、Pentium系列都支持4个处理器特权级别(特权环: R0、R1、R2和R3)
- · 从R0到R3特权能力依次降低
- R0相当于双状态系统的管态
- · R3相当于目态
- R1和R2则介于两者之间,它们能够运行的 指令集合具有包含关系:

$$I_{R0} \supseteq I_{R1} \supseteq I_{R2} \supseteq I_{R3}$$

# 实例: x86系列处理器

各个级别有保护性检查(地址校验、I/O限制)特权级别之间的转换方式不尽相同 四个级别运行不同类别的程序:

- · R0-运行操作系统核心代码
- · R1-运行关键设备驱动程序和I/O处理例程
- R2-运行其它受保护共享代码,如语言系统运行环境
- R3-运行各种用户程序

现有基于x86处理器的操作系统,多数UNIX、Linux以及Windows系列大都只用了R0和R3两个特权级别



#### <u>问题一:CPU怎么来判断当前运行的程序是</u> 系统程序还是用户程序呢?

## 程序状态字PSW

- 一个专门的寄存器,用来指示处理器的状态, PSW (Program Status Word), 通常包括:
- CPU的工作状态码——指明管态还是目态,用来说明当前在CPU上执行的是操作系统还是一般用户,从而决定其是否可以使用特权指令或拥有其它的特殊权力;
- 条件码——反映指令执行后的结果特征;
- 中断屏蔽码——指出是否允许中断



### 问题二: 状态之间如何转换?

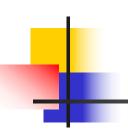
#### 管态 → 目态

通过设置PSW(修改程序状态字)来实现;

用户程序无法直接修改程序状态字;

那么用户程序如何才能去做一些带有"特权"的事情(如I/O)呢?

解决之道是 ———



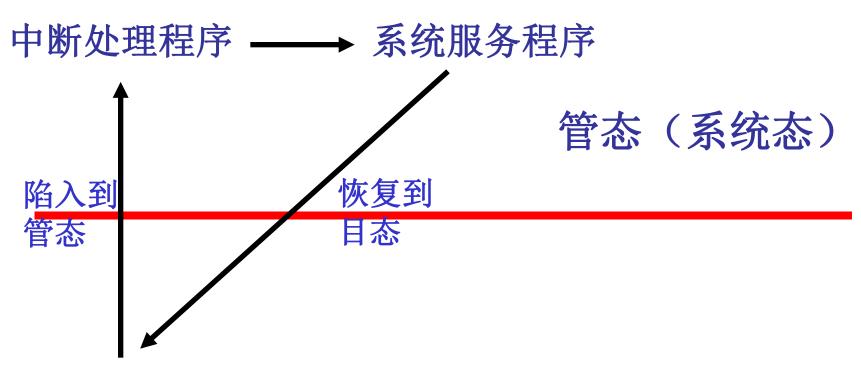
### 二、系统调用

用户程序通过特殊的访管指令,来请求操作系统为其提供某种功能的服务。系统调用指令的实现过程一般是:

- \* 当CPU执行访管指令时,即引起访管中断;
- ◆ 处理器保存中断点的程序执行上下文环境(PSW, PC和其他的一些寄存器), CPU切换到管态。
- \*中断处理程序开始工作,调用相应的系统服务;
- 中断处理结束后,恢复被中断程序的上下文环境, CPU恢复为目态,回到中断点继续执行。



# OS内核



系统调用(汇编指令)

用户程序



#### Linux0.11版本中的系统调用(72个)

<pre>int sys_fork();</pre>	创建进程
<pre>int sys_read();</pre>	读文件
<pre>int sys_write();</pre>	写文件
<pre>int sys_open();</pre>	打开文件
<pre>int sys_execve();</pre>	执行程序
<pre>int sys_time();</pre>	取当前时间
<pre>int sys_chdir();</pre>	更改当前目录
<pre>int sys_mount();</pre>	安装文件系统
***	***

数组sys\_call\_table[]记录了每个系统调用函数的入口地址,系统调用中断int 0x80



# OS需要的硬件特性

- ✓ 受保护的指令;
- ✓ 系统调用;
- 内存保护;
- 中断机制;
- I/O系统;
- 时钟操作。



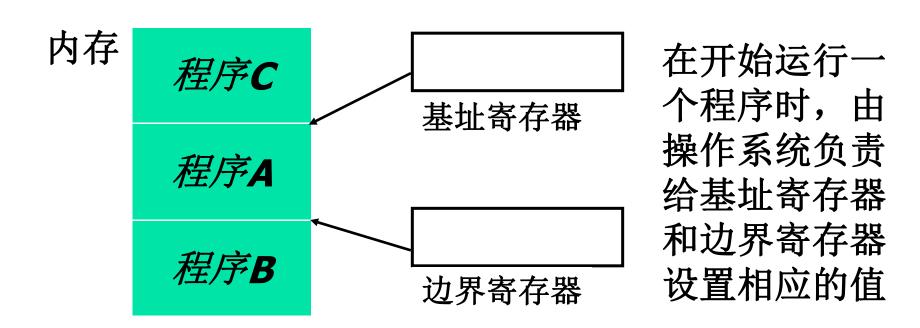
# 三、内存保护

# Why 内存保护?

- \* You can't hurt me, I can't hurt you
- ♣ 防止一个用户程序去访问其他用户程序的数据;
- \* 保护操作系统免受用户程序的破坏。

## 在硬件上如何支持?

最简单的做法:基址寄存器和边界寄存器



虚拟存储技术: 把内存和外存结合起来使用, 硬件提供虚、实地址映射的机制。



# 四、中断机制

中断对于操作系统的重要性就像机器中的驱动齿轮一样

有人把操作系统称为是由

"中断驱动"或者"(中断)事件驱动"

- 什么是中断(interrupt)?
- 指的是由于某个事件的发生,改变了正在CPU上 执行的指令的顺序;
- · 这种事件对应于CPU芯片内部或外部的硬件电路 所生成的电信号。

中断处理的过程:

当中断事件发生时,CPU暂停正在执行的程序, 保留现场后自动转去执行相应事件的处理程序, 待处理完成后返回断点,继续执行被打断的程序。

# 中断的类型

按照Intel公司的文档,中断可以分为两大类,即同步中断和异步中断。

- 同步中断: 指当CPU正在执行指令的时候,由 CPU的控制单元所发出的中断,也称为"异常";
  - ☞ CPU检测到的异常,包括:错误Fault、 陷阱Trap和中止Abort,例如:算术溢出、 被零除、用户态下使用了特权指令等;
  - 程序设定的异常,即程序员通过int、int3等指令来发出的中断请求,也称为软中断,主要用来实现系统调用服务。

- **异步中断**:指由其他的硬件设备在任意的时刻 所发出的中断,简称为"中断":
  - ☞ 可屏蔽中断,即I/O中断,它是当外部设备或通道操作正常结束或发生错误时所发生的中断。例如:打印机打印完成、缺纸,读磁盘时驱动器中没有磁盘等;
  - ☞ 不可屏蔽中断,例如:由掉电、存储器校验错等硬件故障引起的硬件中断;
  - ·每一个中断或异常都用一个0-255之间的整数来标识,称为中断向量,系统根据中断向量, 来为每一个中断或异常指定相应的处理程序。

# 五、I/0系统

完成计算机系统中信息的输入输出功能

## 六、时钟

时钟是操作系统运行时必不可少的硬件设施,在操作系统中需要时钟支持的工作有:

- 在分时系统中,间隔时钟实现进程间按时间片轮转;
- 在实时系统中,按要求的间隔输出正确的时间 信号给实时的控制设备;
- 记录用户和系统所需的绝对时间(年、月、日、时、分、秒)。

85



# 下课啦!



- 第一章作业
- 1.什么是操作系统?其主要功能和特征是 什么?
- 2.简要说明不同种类操作系统所要解决的 主要问题
- 3.什么是微内核机制?