计算机操作系统

任元勇

信息与软件工程学院

Email: lyren@uestc.edu.cn

2017年2月

个人信息

• 姓 名: 任立勇

• 联系电话: 83201884

• 电子邮件: <u>lyren@uestc.edu.cn</u>

• 办公地点: 主楼中220#

考核方式

- 理论和综合应用
 - 基础课程
 - 考研课程
 - 应用价值
- 成绩构成:
 - 期末考试占70%
 - 考勤和作业占10%
 - 中期考试占10%
 - 实验占10%
 - 课程设计(课外)

参考书籍

- 计算机操作系统,汤小丹,汤子瀛 etc.,西安电子科技大学出版社(第3版)
- 现代操作系统, Andrew S.Tanenbaum, 陈向群译, 机械工业出版社, 2009.7
- Operating System Internals and Design principles, William Stallings, TsingHua University Press
- Applied Operating System Concepts, Abraham Silberschatz etc., Higher Education Press

课程构成

• 总学时: 64(上课56学时、上机8学时)

- 第一章 操作系统引论 6学时

- 第二章 进程管理 13学时

- 第三章 处理器调度与死锁 8学时

- 第四章 存储器管理 11学时

- 第五章 设备管理 8学时

- 第六章 文件管理 7学时

- 第七章 接口 3学时

第一章 操作系统概论

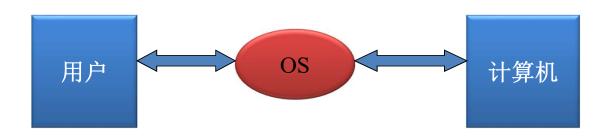
- 1.1 操作系统的目标和作用
- 1.2 操作系统的发展过程
- 1.3 操作系统的基本特征
- 1.4 操作系统的主要功能
- 1.5 OS结构设计

讨论

- 你知道哪些操作系统的名字?
- 你觉得操作系统是什么?
- · 当前对于OS还没有一个统一的、适用的定义!

1.1 操作系统的概念

•操作系统定义:操作系统是一组控制和管理计算机软硬件资源、合理地对各类作业进行调度以及方便用户使用的程序集合。



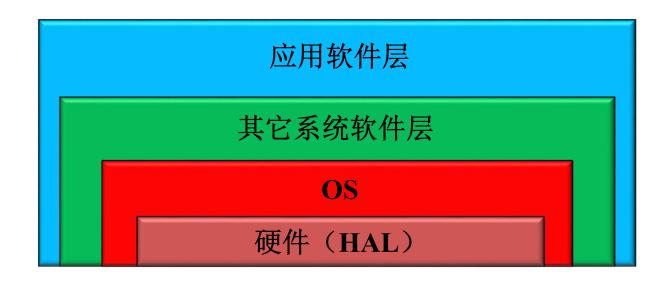
OS是计算机系统的灵魂!

其它定义

 操作系统定义:操作系统是位于硬件层(HAL)之上,所有其它系统 软件层之下的一个系统软件,使得管理系统中的各种软件和硬件资 源得以充分利用,方便用户使用计算机系统。

操作系统的位置

- 硬件抽象层(HAL)之上
- 所有其它软件层之下



1.1.1 操作系统的目标

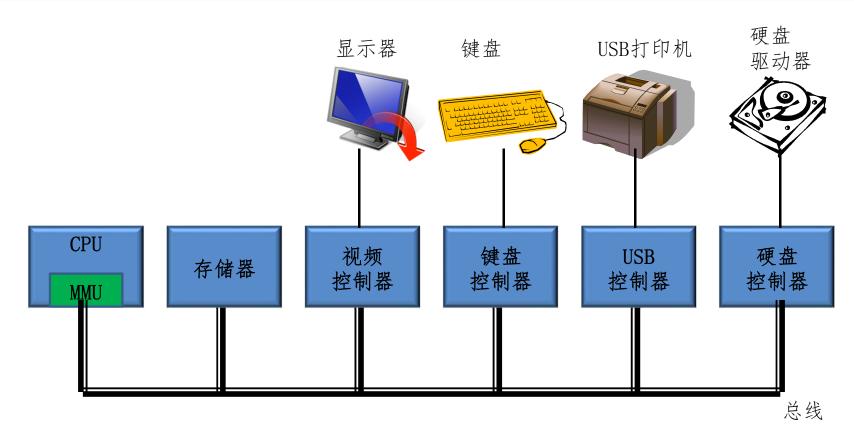
- •有效性:操作系统允许以更有效的方式使用计算机系统资源。
 - -提高系统资源利用率
 - -提高系统的吞吐量
- •方便性:操作系统使计算机更易于使用。



•开放性:实现应用程序的可移植性和互操作性,要求具有统一的开放的环境。



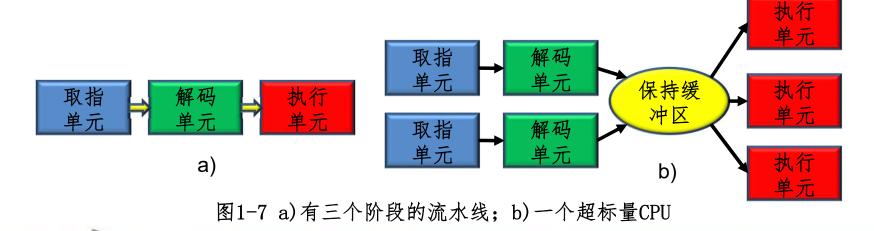
计算机硬件介绍



简化个人计算机中的一些部件

处理器

- 每个CPU有专用的指令集,按照"<mark>取指令、解码、执行</mark>"方式运 行程序
- 所有的CPU都有一些通用和专用寄存器,例如程序计数器PC、堆 栈指针、程序状态字PSW等。操作系统必须知晓所有的寄存器, 以便切换进程



存储器(内存)

- 对存储器的理想要求
 - 访问迅速
 - 充分大
 - 非常便宜

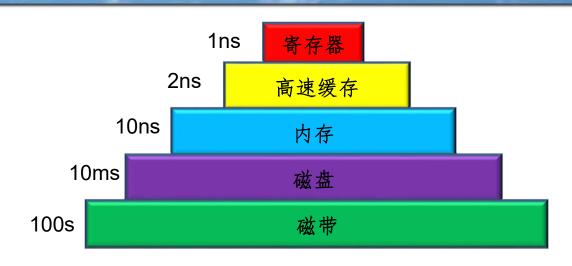


图1-9 典型的存储层次结构

- 缓存系统需考虑的问题
 - 何时把一个新内容放入缓存
 - 把新内容放入缓存的哪一行
 - 在需要时,应该把哪个内容从缓存中移走
 - 把新移走缓存放在某个较大存储器的何处

磁盘

- 与RAM相比,磁盘(硬盘):
 - 成本更低
 - 容量更大
 - 访问时间慢
- 几个概念:
 - 磁道
 - 柱面
 - 扇区

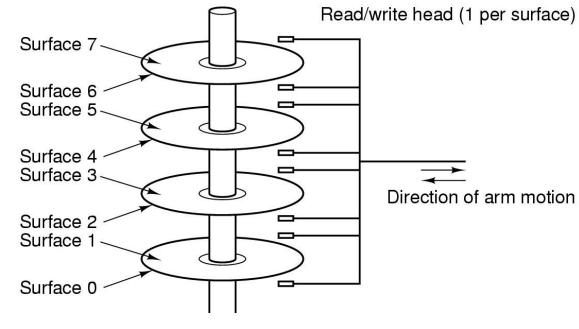


图1-10 磁盘驱动器的构造

I/O设备(1)

- I/O设备包括两个部分: 设备控制器和设备本身
 - 设备控制器任务是为操作系统提供一个简单的接口;每一类设备控制器需要专门的软件控制(称为驱动程序)
 - 设备本身有个相对简单的接口, 因为已经被标准化
- 驱动程序装入操作系统的三种方式:
 - 1. 与操作系统重新链接,并重启系统
 - 2. 在操作系统文件中设置一个入口,重启系统
 - 3. 操作系统运行时接受新的设备驱动程序,无须重启

I/O设备(2)

- 每个设备控制器都有少量用于通信的寄存器,所有设备寄存器的集合构成了I/O端口空间。
- 有些计算机将设备寄存器映射到操作系统的地址空间,不需专门的 I/O指令。而另外一些计算机中,每个寄存器有一个端口地址,有 内核态可使用的专门的IN和OUT指令,供设备驱动程序读写寄存器

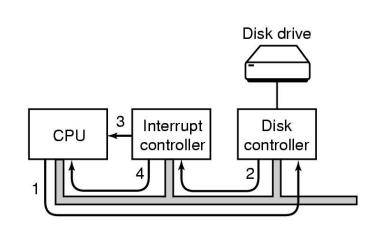
0

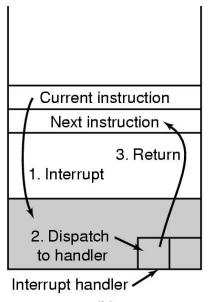
I/O设备(3)

- 实现输入和输出的方式有三种:
- 1. 用户程序发出一个系统调用,内核将其翻译为对应的设备驱动程序过程,该过程启动I/O并在**连续不断的循环**中检查该设备是否完成工作。当I/O结束后,设备驱动程序把数据送到指定的地方,并返回,然后操作系统将控制返回给调用者。这种方式为忙等待

I/O设备(4)

2. 设备驱动程序启动设备并且让该设备在操作完成时发出一个中断. 返回,操作系统在需要时阻塞调用者。



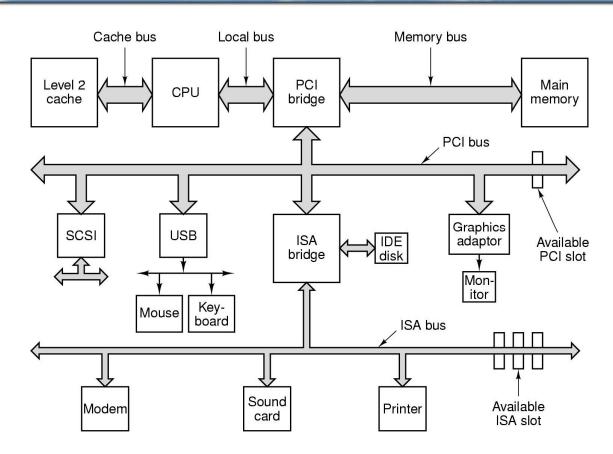


a) 启动一个I/O设备并发出中断的过程; b) 中断处理过程包括取中断、运行中断处理程序和返回到用户程序

I/O设备(5)

3. 直接存储器访问DMA,DMA芯片可以控制在内存和某些控制器之间的位流,而无须持续的CPU干预。CPU对DMA芯片进行设置,说明需要传送的字节数、有关设备和内存的地址以及操作方向,接着启动DMA。当DMA芯片完成时,它引发一个中断

总线



大型Pentium系统的结构

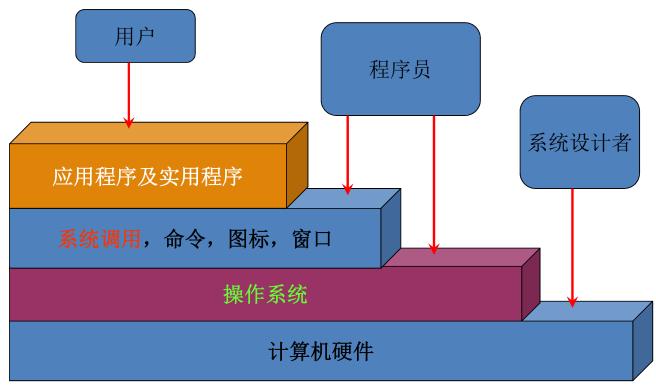
1.1.2 OS的作用

- 1. OS作为用户与计算机硬件系统之间的接口
- ●计算机用户需要的*用户命令*:由OS实现的所有用户命令所构成的集合常被 人们称为OS的<u>Interface</u>(*用户接口*);有时也称为命令接口。

命令的表示形式:

- 字符形式: 较灵活但因繁琐而难记;
- 菜单形式: 试图在字符终端上提供友好的用户界面
- 图形形式: 因直观而易记但不灵活。
- ●应用软件需要的<u>System Call(系统调用)</u>:由OS实现的所有系统调用所构成的集合被人们称为<u>程序接口</u>或<u>应用编程接口(Application Programming Interface, API)</u>。

1. 作为用户与计算机硬件系统之间的接口



OS作为接口的示意图

• 操作系统应<mark>隐藏</mark>复杂的、困难的、丑陋的、特殊的硬件细节,呈现给程序(程序员)良好、清晰、优雅、一致的抽象

操作系统的任务就是创建好的抽象,并实现和管理它所创建的抽象

自顶向下的 观点

2. OS作为计算机系统资源的管理者(软硬件资源)

- 1. 处理机管理, 用于分配和控制处理机;
- 2. 存储器管理,主要负责内存的分配与回收;
- 3. I/O设备管理,负责I/O设备的分配与操纵;
- 4. 文件管理,负责文件的存取、共享和保护。
- 对于多用户系统,需要管理共享资源,避免发生冲突。

- 操作系统的任务在相互竞争的程序之间有序地控制对硬件设备的分配
- 资源管理包括用以下两种不同的方式实现多路复用(共享)资源
 - 1. 在时间上复用: 当一种资源在时间上复用时,不同的程序"轮流" 实用它;例如CPU、打印机等
 - 2. 在空间上复用:每个客户都得到资源的一部分;例如内存、磁盘等

自底向上的 观点

3. OS用作扩充机器

- 裸机: 完全无软件的计算机系统。
- 操作系统用作扩充机器功能, 使其便于使用
- 在裸机上覆盖管理软件,实现对设备的操作,并向上提供一组操作命令。
- 隐藏对设备操作的具体细节,实现对硬件操作的多个层次的抽象。
- 通常把覆盖了软件的机器称为扩充机器或虚机器。

1.2 操作系统的发展

- 第一代(1945~1955)
 - 真空管和穿孔卡片(无操作系统)
- 第二代(1955~1965)
 - 晶体管和批处理系统(出现了操作系统, FMS)
- 第三代(1965~1980)
 - 集成电路芯片和多道程序设计(分时系统出现, MULTICS)
- 第四代(1980~至今)
 - 个人计算机(DOS, Windows, Linux)



操作系统的分类

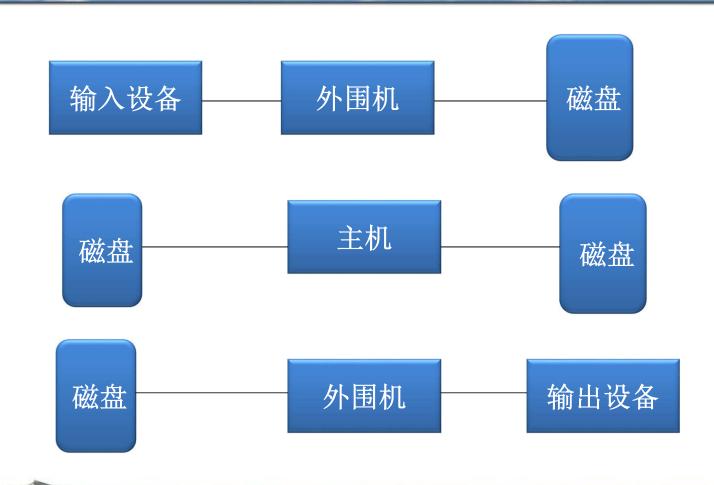
- 1. 无操作系统时的计算机系统
- 2. 单道批处理操作系统
- 3. 多道批处理操作系统
- 4. 分时操作系统(time-sharing system)
- 5. 实时操作系统(real time system)
- 6. 微机操作系统
- 7. 网络操作系统(network operating system)
- 8. 分布式操作系统(distributed operating system)
- 9. 多处理器操作系统(multi-processor system)
- 10. 嵌入式操作系统(embedded operating system)
- 11. 多媒体操作系统

1.2 操作系统的发展过程

1. 无操作系统时的计算机系统

- 1、 人工操作方式
- 一台计算机的所有资源由用户独占,降低了计算机资源利用率,人操作慢, 出现了严重的人机矛盾。
- 2、 脱机输入输出方式
 - -在外围计算机的控制下,实现输入输出。
 - -主要解决了CPU与设备之间不匹配的矛盾

图1-3 脱机1/0示意图



2.单道批处理系统

- 1、在脱机输入/输出方式基础上增加监督程序(Monitor)。在内存中仅存一 道作业运行,运行结束或出错,才自动调另一道作业运行。
- 2、单道批处理系统主要特征:自动性、顺序性、单道性。
- 3、单道批处理系统主要优点:减少人工操作,解决了作业的自动接续。
- 4、单道批处理系统主要缺点:平均周转时间长,没有交互能力。

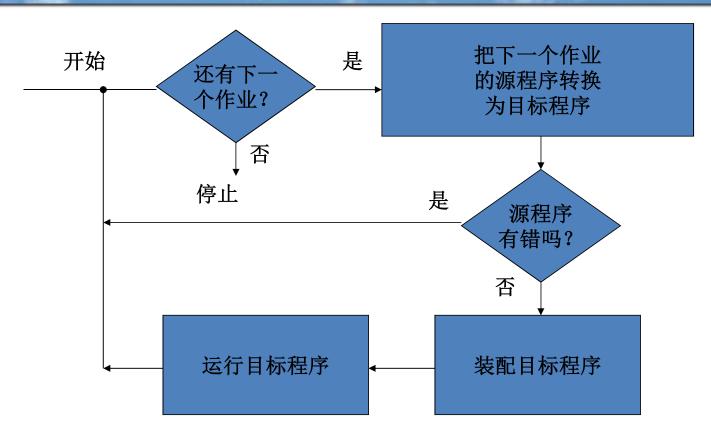
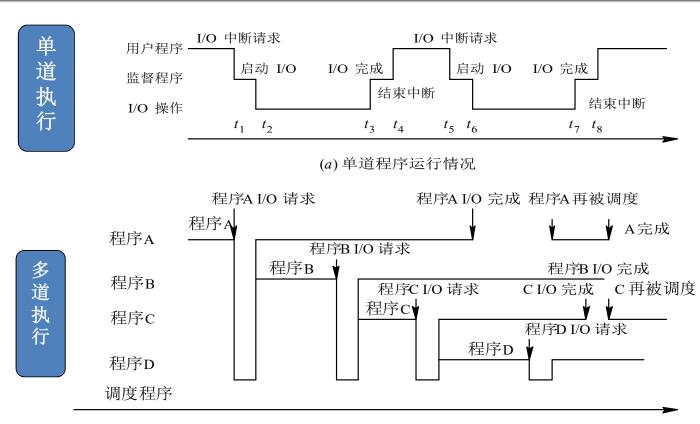


图 1-4 单道批处理系统的处理流程

3.多道批处理系统

- 一、多道程序的概念: 在内存中存放多道作业运行,运行结束或出错,自动调度内存中的另一道作业运行。
- ●多道程序带来的好处:
- 1、提高CPU的利用率。
- 2、提高内存和I/O设备利用率。
- 3、增加系统吞吐率。

举例1:多道执行情况



(b) 四道程序运行情况

图 1-5 单道和多道程序运行情况

■ 示例2

设内存中有三道程序A、B和C,它们按A、B、C的优先次序执行。它们的计算和I/O操作时间如下表所示。

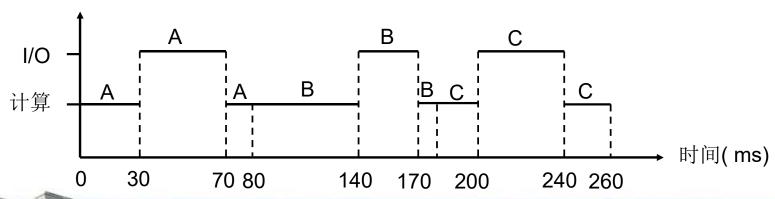
假设三道程序使用能够相同的设备进行I/O操作,即程序以串行方式使用设备,试画出单道运行和多道运行的时间关系图(调度程序的执行时间忽略不计)。在两种情况下,完成这三道程序各要花多少时间?

	Α	В	С
计算	30ms	60ms	20ms
I/O	40ms	30ms	40ms
计算	10ms	10ms	20ms

- 示例2
- 单道运行的甘特图

	Α	В	С
计算	30ms	60ms	20ms
I/O	40ms	30ms	40ms
计算	10ms	10ms	20ms



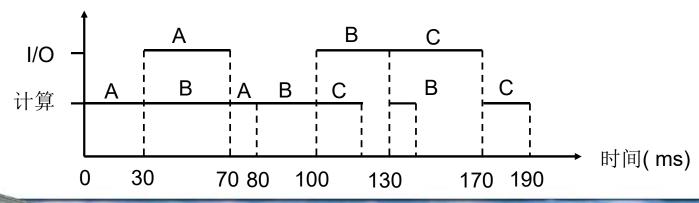


■ 示例2

■ 多道运行的甘特图

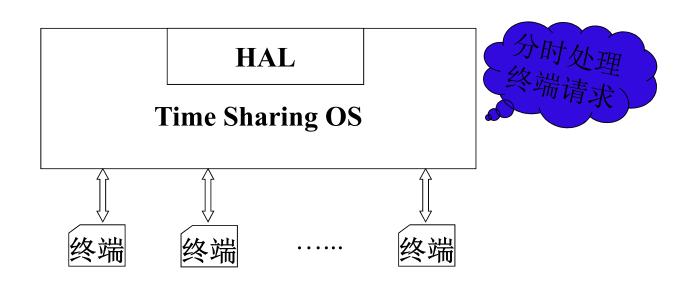
	А	В	С
计算	30ms	60ms	20ms
I/O	40ms	30ms	40ms
计算	10ms	10ms	20ms

活动列表



- 二、多道批处理系统主要特征:
 - 多道性、无序性、调度性(进程调度和作业调度)。
- 三、多道批处理的主要优点:提高了资源利用率和吞吐能力。
 - 多道批处理的主要缺点:平均周转时间长,没有交互能力。
- 四、多道批处理系统需要解决的5个问题
 - 1、处理机管理:分配和控制CPU。
 - 2、存储器管理:内存分配与回收
 - 3、I/0设备管理: I/0设备的分配与操纵。
 - 4、文件管理:文件的存取、共享和保护。
 - 5、作业管理:如何组织作业运行。

4. 分时操作系统(On-line)



界面1: 交互式命令语言(eg. shell, command)

界面2: 图形用户界面(GUI)

4 分时系统

一、分时系统的产生

用户需要:人机交互、共享主机、便于用户上机

二、分时系统实现中的关键问题:

及时接收:实现多个用户的信息及时接收。

及时处理:及时控制作业的运行。

- 三、分时系统实现的方法
- 简单分时系统
- 具有"前台"和"后台"的分时系统
- 多道分时系统

4 分时操作系统(Cont.)

• 特点:

- <u>多路性</u>:一个主机与多个终端相连;
- 独立性: 彼此独立操作, 互不干扰;
- 及时性:系统能在很短的时间得到回答;
- 交互性: 能实现人机对话(区别于批处理系统);
- 典型系统:
 - Multics (MIT)
 - UNIX

计算机考研《操作系统》单项选择题

UNIX操作系统是著名的()。

A.多道批处理系统 B.分时系统 C.实时系统 D.分布式系统 答案 B

引入多道程序的目的在于()。

- A.充分利用CPU,减少CPU等待时间
- B.提高实时响应速度
- C.有利于代码共享,减少主、辅存信息交换量
- D.充分利用存储器

答案A

下述()不属于多道程序运行的特征。

- A.多道 B.运行速度快
- C.宏观上并行 D.实际上多道程序是穿插运行的

答案 B

下列性质中,哪一个不是分时系统的特征()。

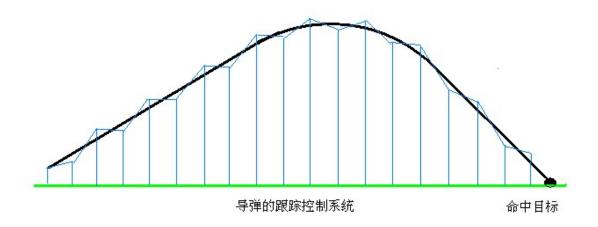
A.交互性 B.多路性 C.成批性 D.独占性 答案 C

5. 实时系统

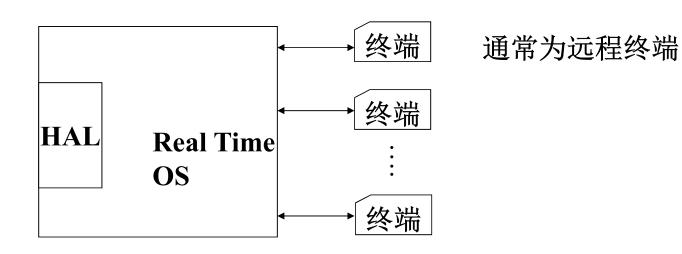
● 所谓实时系统:是计算机及时响应外部事件的请求,在规定 的时间内完成对该事件的处理,并控制所有实时设备和实 时任务协调一致的运行。

一、应用需求的类型

- 1、实时控制系统:工业控制,军事控制,医疗控制,……
- 2、实时信息处理系统: 航班定票, 联机情报检索, …….



实时信息处理



特点:

- (1) 响应及时 (prompt response)
- (2) 可靠性高 (high reliability)

二、实时任务的类型

- 1、按任务执行是否为周期性来化分
 - 周期性实时任务
 - 非周期性实时任务
- 2、按截止时间来化分
 - 硬实时任务
 - 软实时任务

三、实时系统的特征

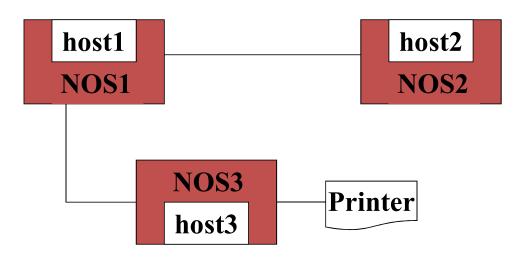
- 1、多路性:能对多个对象进行控制。
- 2、独立性:独立运行,不混淆,不破坏。
- 3、交互性:仅限于访问系统中某些特定的专用服务程序。
- 4、可靠性: 高可靠性, 应具有多级容错防护能力。
- 5、及时性:不同的系统要求不一样,控制对象必须在截止时间内完成。

6. 微机操作系统

- 配置在微型机上的操作系统
- 1. 单用户单任务操作系统,如CP/M,DOS;
- 2. 单用户多任务操作系统,如WINDOWS;
- 3. 多用户多任务操作系统,如UNIX,LINUX

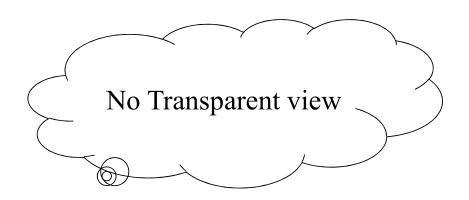
7. 网络操作系统

建立在宿主操作系统之上,提供网络通讯、网络资源共享、网络服务的软件包。



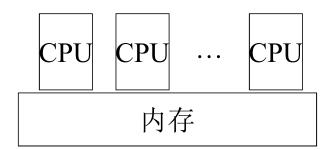
网络操作系统的目标

- 相互通讯
- 资源共享(信息,设备)
- 提供网络服务
 - database server
 - ftp server
 - e-mail server
 - telnet server
 - etc.



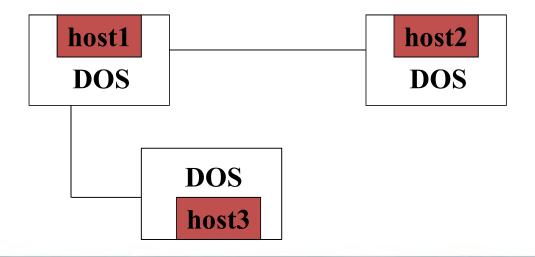
8. 分布式操作系统

- 紧耦合: (tightly coupled)
 - 由多机系统发展而来(多CPU)
 - 有公共内存
 - 多处理机操作系统



8. 分布式操作系统

- 松散耦合: (loosely coupled)
 - 由计算机网络发展而来(多Host)
 - 无公共内存,无公共时钟



8. 分布式操作系统(Cont.)

- 分布式软件系统(Distributed Software Systems)
 - 是支持分布式处理的软件系统,是在由通信网络互联的多处理机体系结构上执行任务的系统。它包括分布式操作系统、分布式程序设计语言及其编译 (解释)系统、分布式文件系统和分布式数据库系统等。
- 分布式操作系统特征:
 - 统一的操作系统
 - 资源的进一步共享
 - ・内存, CPU
 - 可靠性
 - 透明性

8.分布式操作系统(Cont.)

- 目标: 进一步共享资源, 使负载均衡, 计算加速。
 - CPU
 - 内存
- 途径: 迁移 (migration)
 - 作业迁移
 - 进程迁移(线程一般随同进程迁移)
- 例子:
 - Solaris MC

9. 多处理机操作系统

- 多处理机系统
 - 具有公共内存的多CPU系统
- 对称多处理机系统(SMP-symmetric multi-processor)
 - 没有主从关系的多处理机系统
- 多处理机操作系统
 - 有效管理和使用多个CPU的操作系统
 - 复杂性:多个主动体(CPUs)
- 例子:
 - UNIX, Linux, Windows

10. 嵌入式操作系统

- 嵌入在掌上电脑、通讯设备、车载系统、信息家电等非计算机类设施上的操作系统。用于对这些设备和产品进行控制和监控的操作系统。
- 特点: 微内核结构(Micro-kernel), 许多操作系统功能(文件系统,设备驱动) 以应用程序模式运行。
 - 核心小(基本内存管理,CPU管理,通讯程序), 适应范围广, 可靠性高
 - 效率低
- 例子:
 - Win CE .NET, android, PalmOS
 - Vxworks, uc/os



11.多媒体操作系统

- 指除具有一般操作系统的功能外,还具有多媒体底层扩充模块,支持高层多媒体信息的采集、编辑、播放和传输等处理功能的系统。
- 分3类:
 - 具有编辑和播放双重功能的开发系统;
 - 以具备交互播放功能为主的教育/培训系统;
 - 用于家庭娱乐和学习的家用多媒体系统。

1.3 操作系统的基本特征

- 现代OS的四个基本特征:
 - 1、并发性(最重要的特征)
 - 2、共享性
 - 3、虚拟性
 - 4、异步性
- 并发是最重要的特征,其它特征都以并发为前提。

考研题目:现代操作系统的两个基本特征是()和资源共享。

- A. 多道程序设计 B. 中断处理
- C. 程序的并发执行 D. 实现分时与实时处理

1.3.1 并发

1. 并发——并行性和并发性, 并发执行的过程。

-- 并行性: 是指两个或多个事件在同一时刻发生。

--并发性:是指两个或多个事件在同一时间间隔内发生。

• 任务共行

- 从宏观上看,任务共行是指系统中有多个任务同时运行
- 从微观上看,任务共行是指单处理机系统中的<u>任务并发(Task Concurrency: 即多个任务在单个处理机上交替运行)</u>或多处理机系统中的<u>任务并行(Task Parallelism: 即多个任务在多个处理机上同时运行)</u>。

2. 引入进程

- 程序:静态实体

进程:系统中能独立运行并作为资源分配的基本单位,由一组机器指令、数据和堆栈等组成的独立运行的活动实体。

3. 引入线程

- 进程作为资源分配的基本单位
- 线程作为独立运行和调度的基本单位

1.3.2 共享

共享:是指系统中的资源可供内存中多个并发执行的进程共同使用。

1、互斥共享方式:

- 把在一段时间内只允许一个进程访问的资源,称为<mark>临界资源</mark>。如打印机、 栈、表格等
- 系统中的临界资源可以提供给多个进程使用,但一段时间内仅允许一个 进程使用,称为互斥共享方式。

1.3.2 共享

2、同时访问方式:

- 从宏观上看,资源共享是指多个任务可以同时使用系统中的软硬件资源。
- --从微观上看,多个进程交替互斥地使用系统中的某个资源。例如磁盘。
- 并发和共享是操作系统的两个最基本的特征,它们又是互为存在的条件

1.3.3 虚拟性

虚拟:是指通过某种技术把一个物理实体变为(映射为)若干个逻辑上的对应物。

1. 时分复用技术

- 虚拟处理机: 分时实现

- 虚拟设备: SPOOLING技术

2. 空分复用技术

- 虚拟磁盘技术:逻辑分区

- 虚拟存储器: 虚拟存储管理实现

1.3.4 异步性

- 1、执行结果不确定,程序不可再现。
- 2、异步性,多道程序环境下程序(进程)以异步的方式执行,每 道程序在何时执行、各自执行的顺序、完成时间都是不确定的, 也是不可预知的。

1.4 操作系统的主要功能

- 操作系统的主要任务:
 - 为多道程序的运行提供良好的运行环境,以保证多道程序能有条不紊地、 高效地运行,并能最大程度地提高系统中各种资源的利用率和方便用户的 使用。
- 操作系统应具有五方面的功能:
- 1、处理机管理(CPU)
- 2、存储器管理
- 3、设备管理
- 4、文件管理
- 5、方便用户使用的用户接口。

主要功能:按照一定的算法把处理机分配给进程(线程),并对其进行有效的管理和控制。

(1) 进程控制

进程控制的主要功能:是为作业/任务创建进程、撤消已结束的进程,以及控制进程在运行过程中的状态转换。

(2) 进程同步

- 进程同步的主要任务是为多个进程的运行进行协调。
- 两种协调方式:
- ①进程互斥方式,这是指诸进程(线程)在对临界资源进行访问时,应采用互 斥方式;
- ②进程同步方式,指进程相互合作去完成共同的任务时,诸进程之间的协调。

(3) 进程通信

- 进程通信——是进程之间的信息交换。
 - 当相互合作的进程(线程)处于同一计算机系统时,通常在它们之间是采用直接通信方式,即由源进程利用发送命令直接将消息(message)挂到目标进程的消息队列上,以后由目标进程利用接收命令从其消息队列中取出消息。
- 通信方式: 共享存储器、消息、管道等。

(4) 调度

- 在后备队列上等待的每个作业,通常都要经过调度才能执行。
- 在操作系统中作业运行需经作业调度和进程调度才能执行完成。
- 作业与进程的关系
 - 作业进入内存后变为进程
 - 一个作业通常与多个进程相对应

1.4.2 存储器管理功能

• 主要任务:

为多道程序的运行提供良好的环境,方便用户使用存储器,提高存储器 的利用率以及能从逻辑上扩充内存。

- > 存储器管理功能
 - 内存分配
 - 内存保护
 - 地址映射
 - 内存扩充

1.4.2 存储器管理功能

(1) 内存分配

- 内存分配有两种方式:
- ①静态分配方式,每个作业运行之前分配好内存空间,在作业的整个运行期间 不再改变。
- ②动态分配方式中,每个作业在运行前或运行中,均可申请新的附加内存空间, 以适应程序和数据的动态增涨。

内存分配的机制中应具有这样的结构和功能:

- ①内存分配的数据结构,该结构用于记录内存空间的使用情况。
- ②内存分配功能——为用户程序分配内存空间;
- ③内存回收功能——当用户不再需要的内存时,系统能回收内存的功能。

- (2) 内存保护
- 内存保护的主要任务:

确保每道用户程序都只在自己的内存空间内运行,彼此互不干扰。

- 内存保护机制:
 - 设置两个界限寄存器 : 存放执行程序的上界和下界
 - 越界检查都由硬件实现

(3) 地址映射

- •地址空间,目标程序或装入程序限定的空间, 称为"地址空间"。单元的编号称为逻辑地址, 又称为相对地址。
- •内存空间—由内存中的一系列单元所限定的地址范围称为"内存空间",其中的地址称为"物理地址"。
- •地址映射—运行时,将地址空间中的逻辑地址转换为内存空间中与之对应的物理地址,称为地址映射。

(4) 内存扩充

- 借助于虚拟存储技术
 - 从逻辑上去扩充内存容量,使用户所感觉到的内存容量比实际内存容量大 得多;
- 扩充内存必须具有内存扩充机制:
 - (1) 请求调入功能。在程序运行过程中,若所需的程序和数据尚未装入内存,可由OS从磁盘中将所需部分调入内存,继续运行。
 - (2) 置换功能。将内存中的一部分暂时不用的程序和数据调出到磁盘上, 然后再将所需调入的部分装入内存

1.4.3 设备管理功能

• 设备管理的主要任务:

完成用户进程提出的I/O请求;为用户进程分配其所需的I/O设备;提高CPU和I/O设备的利用率;提高I/O速度;方便用户使用I/O设备。

- 设备管理具有的功能:
 - (1)缓冲管理:有效地缓和CPU和I/O设备速度不匹配的矛盾,提高CPU的利用率。

对于不同的系统, 可以采用不同的缓冲区机制

1.4.3 设备管理功能

(2) 设备分配

- ①设备分配的基本任务,是根据用户进程的I/O请求,按照某种设备分配策略,为之分配其所需的设备。
- ②为了实现设备分配,系统中应设置设备控制表、控制器控制表等数据结构,用于记录设备及控制器的标识符和状态,以供进行设备分配时参考。
- ③不同的设备类型(独占、共享)而采用不同的设备分配方式。

1.4.3 设备管理功能

(3). 设备处理

- 设备处理程序又称为设备驱动程序。
- 设备处理基本任务:是用于实现CPU和设备控制器之间的通信,即由CPU向设备控制器发出I/O命令,要求它完成指定的I/O操作;反之由CPU接收从控制器发来的中断请求,并给予迅速的响应和相应的处理。
- 处理过程:
 - 检查请求的合法性→设备空闲否?→向控制器发I/O命令→启动I/O执行。

1.4.4 文件管理功能

- 文件管理的主要任务:是对用户文件和系统文件进行管理,满足读写要求,并 保证文件的安全性。
- 文件管理的主要功能
- 1. 文件存储空间的管理
 - 对诸多文件及文件的存储空间,实施统一的管理。
 - 基于数据结构记录文件存储空间使用情况
 - 一 对存储空间进行分配和回收的功能。

1.4.4 文件管理功能

2. 目录管理

- 为每个文件建立目录项:文件名、文件属性、物理位置等;
- 并对众多的目录项加以有效的组织与管理(例如,按名存取,文件共享)。

3. 文件的读/写管理和保护

- (1) 文件的读 / 写管理: 是根据用户的请求, 从外存中读取数据或将数据写入外存。
 - (2) 文件保护: 即存取控制功能:
 - ①防止未经核准的用户存取文件;
 - ②防止冒名顶替存取文件;
 - ③防止以不正确的方式使用文件。

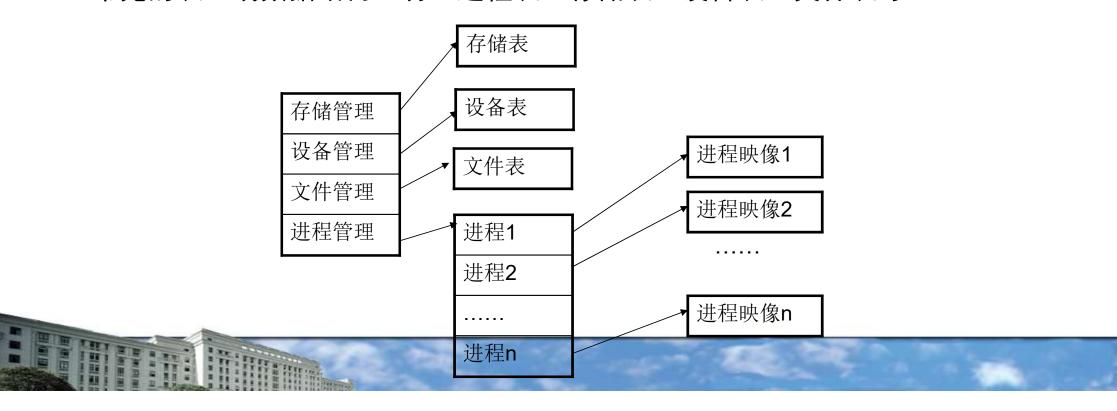
1.4.5 用户接口

方便用户使用的用户接口

- 1. 命令接口——用户可通过该接口向作业发出命令以控制作业的运行。
 - (1) 联机用户接口:
 - (2) 脱机用户接口:
 - (3) 图形接口(GUI)
- 2. 程序接口——由一组系统调用组成,每一个系统调用都是一个能完成特定功能的子程序,每当应用程序要求OS提供某种服务(功能)时,便调用具有相应功能的系统调用。

总结:操作系统控制结构

- 操作系统是管理计算机资源的、采用表格(或数据结构)来记载各资源的信息、 采用代码来实现对资源的管理、维护、更新等。
- 常见的表(或数据结构)有:进程表、存储表、设备表、文件表等。



1.5 OS结构设计

• 操作系统是一个大型系统软件, 其结构已经历了四代的变革:

第一代的OS是无结构的;

第二代OS采用了模块式结构;

第三代是层次式结构

现代OS结构是微内核结构;

1.5.1 传统的操作系统结构

- 操作系统中增加了越来越多的功能,并且随着底层硬件更高的性能,更加通用,操作系统的大小和复杂性也随着增加。
- 为了控制该软件的复杂性,在开发0S时,先后引入了分解、模块化、抽象和 隐蔽等方法。开发方法的不断发展,促进了0S结构的更新换代。

1. 无结构操作系统

- 在早期开发操作系统时,设计者只是把他的注意力放在功能的实现和获得高的效率上,缺乏首尾一致的设计思想。
- OS是为数众多的一组过程的集合,各过程之间可以相互调用,在操作系统内部不存在任何结构,因此,有人把它称为整体系统结构。

• 缺陷:

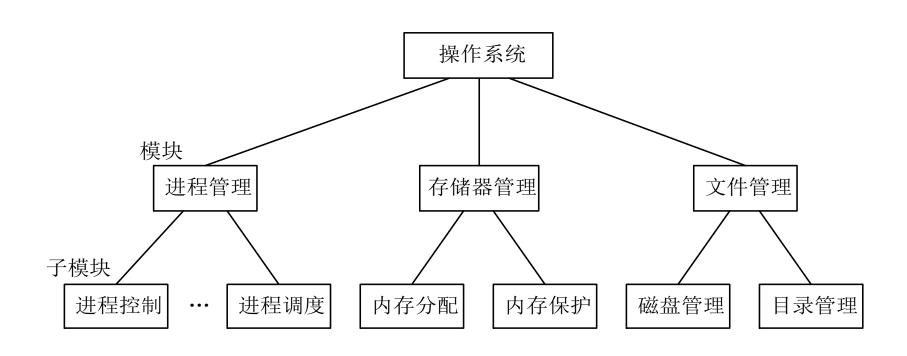
- 设计出的操作系统既庞大又杂乱, 缺乏清晰的程序结构。
- 编制出的程序错误很多,给调试工作带来很多困难;增加了维护人员的负担。

2. 模块化OS结构

(1)模块化结构(模块-接口法)

- 使用分块结构的系统包含若干module (模块);其中,每一块实现一组基本 概念以及与其相关的基本属性。
- 块与块之间的相互关系:
 - 所有各块的实现均可以任意引用其它各块所提供的概念及属性。

2. 模块化OS结构



模块化操作系统结构

(2)模块化OS的优缺点

优点:

- ①提高了OS设计的正确性、可理解性和可维护性。
- ②增强了0S的可适应性。
- ③加速了OS的开发过程。

缺点:

- ①对模块的划分及对接口的规定要精确描述很困难。
- ②从功能观点来划分模块时,未能将共享资源和独占资源加以区别;

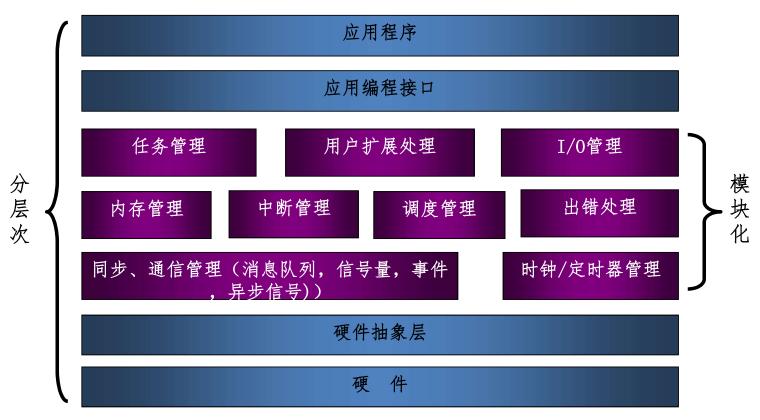
3. 分层式OS结构

- 使用分层系统结构包含若干layer(层);其中,每一层实现一组基本概念 以及与其相关的基本属性。
- 层与层之间的相互关系:
 - 所有各层的实现不依赖其以上各层所提供的概念及其属性,只依赖其直接 下层所提供的概念及属性;
 - 每一层均对其上各层隐藏其下各层的存在。

3. 分层式OS结构

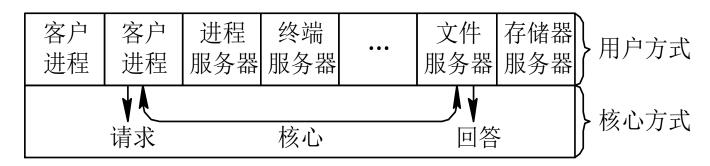
- 层次的设置时应考虑的几个因素
- (1)程序嵌套。通常OS的每个功能的实现,并非是只用一个程序便能完成的, 而是要经由若干个软件层才有可能完成,因此在考虑实现OS时,每个功能 可能形成的程序嵌套。
- (2) 运行频率。将那些经常活跃的模块放在最接近硬件的层。
- (3)公用模块。把供多种资源管理程序调用的公用模块,设置在最低层,以便调用。
- (4) 用户接口。命令接口、程序接口以及图形用户接口。这些接口应设置在OS的最高层,直接提供给用户使用。

操作系统层次化体系结构



DeltaCORE的体系结构: 层次十模块结构

1.5.2客户/服务器模式(Client-Server Model)



单机环境下的客户/服务器模式

• 优点

- 提高了系统的可扩展性
- 增强了系统的可靠性
- 可移植性好
- 提供了对分布式系统的支持
- 缺点
 - 运行效率有所降低:消息传递开销+模式切换开销

1.5.3 面向对象的程序设计技术 (Object-Orientated Programming)

1) 面向对象技术的基本概念

所谓对象,是指在现实世界中具有相同属性、服从相同规则的一系列事物的抽象,而把其中的具体事物称为对象的实例。OS中的各类实体如进程、线程、消息、存储器等,都使用了对象这一概念,相应地,便有进程对象线程对象、存储器对象等。



图 1-7 一个对象的示意图

2) 面向对象技术的优点

- (1) 可修改性和可扩充性。由于隐蔽了表示实体的数据和操作,因而可以改变对象的表示而不会影响其它部分,从而可以方便地改变老的对象和增加新的对象。
- (2) 继承性。继承性是面向对象技术所具有的重要特性。继承性是指子对象可以继承父对象的属性,这样,在创建一个新的对象时, 便可减少大量的时空开销。
- (3) 正确性和可靠性。由于对象是构成操作系统的基本单元,可以独立地对它进行测试,这样,比较易于保证其正确性和可靠性,从而比较容易保证整个系统的正确性和可靠性。

1.5.4 微内核OS结构

- 所谓微内核技术,是指精心设计的、能实现现代OS核心功能的小型内核,它与一般的OS(程序)不同,它更小更精炼,它不仅运行在核心态,而且开机后常驻内存,它不会因内存紧张而被换出内存。
- 当前比较流行的、能<mark>支持多处理机</mark>运行的OS几乎全部都采用了微内核结构, 如Mach OS, windows 2000

- □ 微内核所提供的功能,通常都是一些最基本的功能,如进程管理、低级存储器管理、中断和陷入处理、进程间通信、低级I/0功能。
- □微内核特点
 - > 足够小的内核。
 - ▶基于客户/服务器模式
 - ▶应用"机制与策略分离"原理
 - > 采用面向对象技术

作业

- 1. 解释操作系统在计算机系统中的地位和基本功能。
- 2. 简述批处理系统、分时系统以及实时系统各自的特点。
- 3. 并发和并行有何区别?
- 4. 什么是操作系统?从资源管理的角度去分析操作系统,它的主要功能是什么?
- 5. 操作系统的主要特征是什么?为什么会具有这样的待征?