

操作系统原理与实践

授课教师：赵太银

电子邮箱：269908185@qq.com



课程介绍

- 操作系统课程是计算机教育的**必修内容**
- 计算机专业的**核心课程**
- 计算机技术开发的**理论基础**
- 计算机相关专业的**考研科目**



课程目标

- C01：要求学生从系统级上**掌握分析和设计计算机操作系统的基本思路**，对内核、进程、线程等核心概念具有系统级的认识与理解。
- C02：要求学生**理解操作系统的设计原理、框架、目标、各部分的工作原理**，能够从系统级上应用内核、进程、线程等核心概念对操作系统进行设计并改进等。
- C03：要求学生能够**理解各类操作系统中的经典问题和典型应用**，并搭建Linux环境Centos 6.0，开展相应的实验，实现这些经典问题及应用。
- C04：要求学生理解计算机操作系统的关键功能，并搭建Linux环境Centos 6.0，开展相应的实验，实现这些关键功能，**培养系统能力和解决复杂工程问题的能力**。



课程指标点

- **GR1.3：理解系统的概念及其在软件工程领域的体现，能对复杂软件工程问题的解决方案进行分析，并尝试改进**
- **GR4.1：能够采用科学的方法对复杂工程问题中的关键环节，设计相应的实验方案，搭建实验环境，开展实验**



课程介绍

学习什么

使用操作
系统

不是

而是

- 理解操作系统如何工作
- OS的工作方式
- OS的内部结构与数据结构
- 设计OS过程中的问题、解决方案和折中权衡
- 操作系统中的典型技术和应用



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%



教材&参考书籍

■ 课程教材：

- 计算机操作系统，汤小丹、汤子瀛等,西安电子科技大学出版社（第四版）
- **Linux操作系统应用编程**，刘玢、陈佳、肖堃、丁熠，人民邮电出版社

■ 参考书目：

- 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：80（上课56学时、上机24学时）**

- **第一章 操作系统引论**
- **第二章 进程管理**
- **第三章 处理器调度与死锁**
- **第四章 存储器管理**
- **第五章 输入输出系统**
- **第六章 文件及文件系统**
- **第七章 UNIX/Linux系统入门**
- **第八章 文件操作与权限管理**
- **第九章 进程运行与监控**
- **第十章 Shell的交互功能与Shell程序设计**
- **第十一章 操作系统接口与应用开发**
- **第十二章 线程与线程控制**

2学时

10学时

8学时

10学时

2学时

4学时

2学时

4学时

4学时

2学时

6学时

2学时



讨论

- 你觉得**操作系统**是什么？
- 你知道**哪些操作系统**的名字？



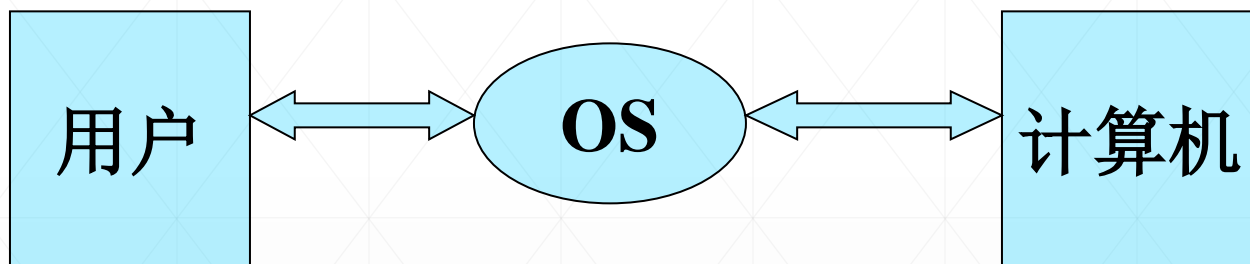
讨论 - 什么是操作系统

- **An Operating System is a program that**
 - ☐ **managers the computer hardware**
 - ☐ **provides a basis for application programs**
 - ☐ **acts as an intermediary between the computer user and the computer hardware**
 - **OS is a resource allocator**
 - ☐ **Manages all resources**
 - ☐ **Decides between conflicting requests for efficient and fair resource use**
 - **OS is a control program**
 - ☐ **Controls execution of programs to prevent errors and improper use of the computer**
-



讨论：操作系统的概念

- **操作系统定义1：** 操作系统是一组控制和管理计算机软硬件资源、合理地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 (汤 第三版)

- **操作系统定义2 (Operating System, OS) :** 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。





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 操作系统的结构设计



1.1.1 操作系统的目标

- **方便性:** 操作系统使计算机更易于使用。
- **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
- **可扩展性:** 在操作系统中,允许有效地开发, 测试和引进新的系统功能。
- **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。

A light blue starburst-shaped callout box with a jagged border, containing red text.

这两个目标最重要



1.1.2 OS的作用

根据OS的定义看作用

- 作为用户与计算机硬件系统之间的接口
- 作为资源管理者的操作系统
- 作为扩展机器的操作系统



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**Interface(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**System Call(系统调用)**：由OS实现的所有系统调用所构成的集合被人们称为**程序接口**或**应用编程接口(Application Programming Interface,API)**。



1.1.2 OS的作用

1. 作为用户与计算机硬件系统之间的接口

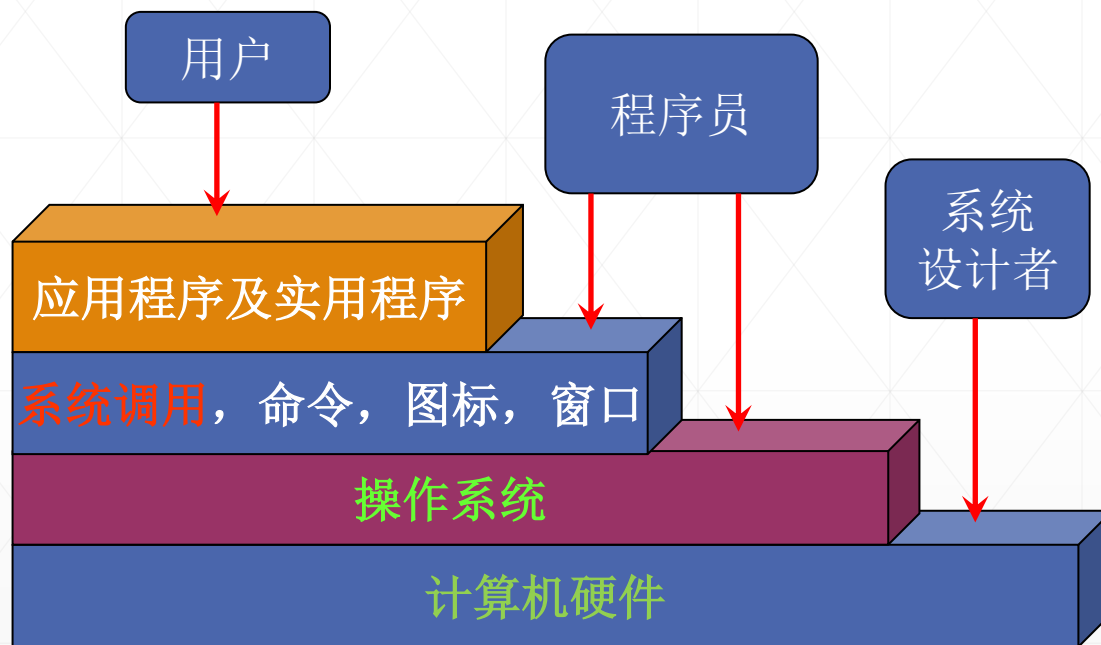


图1-2 OS作为接口的示意图



1.1.2 OS的作用

- 操作系统应**隐藏**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

自顶向下的观点

- 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



1.1.2 OS的作用

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。



1.1.2 OS的作用

- 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

自底向上的
观点

- 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

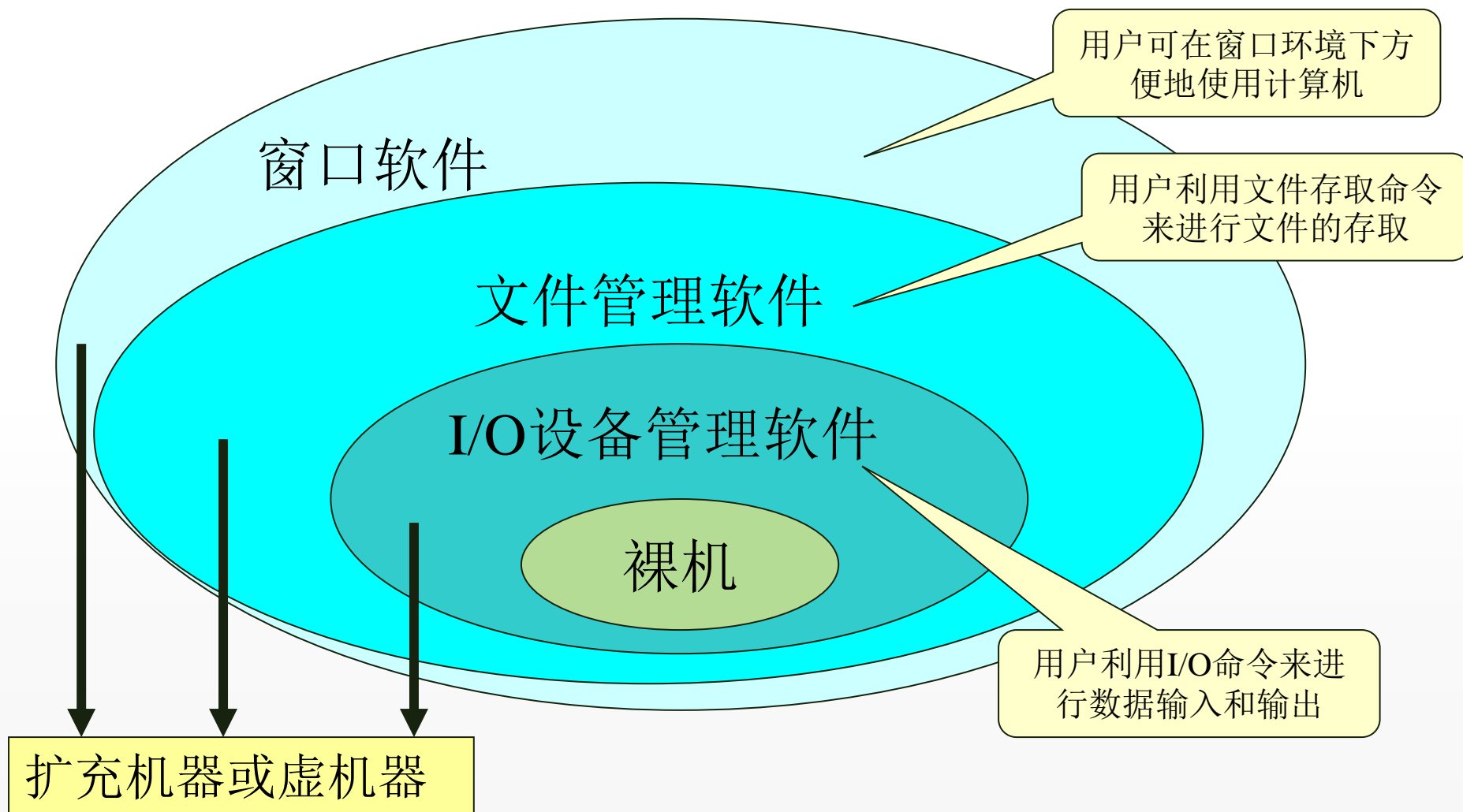


1.1.2 OS的作用

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。

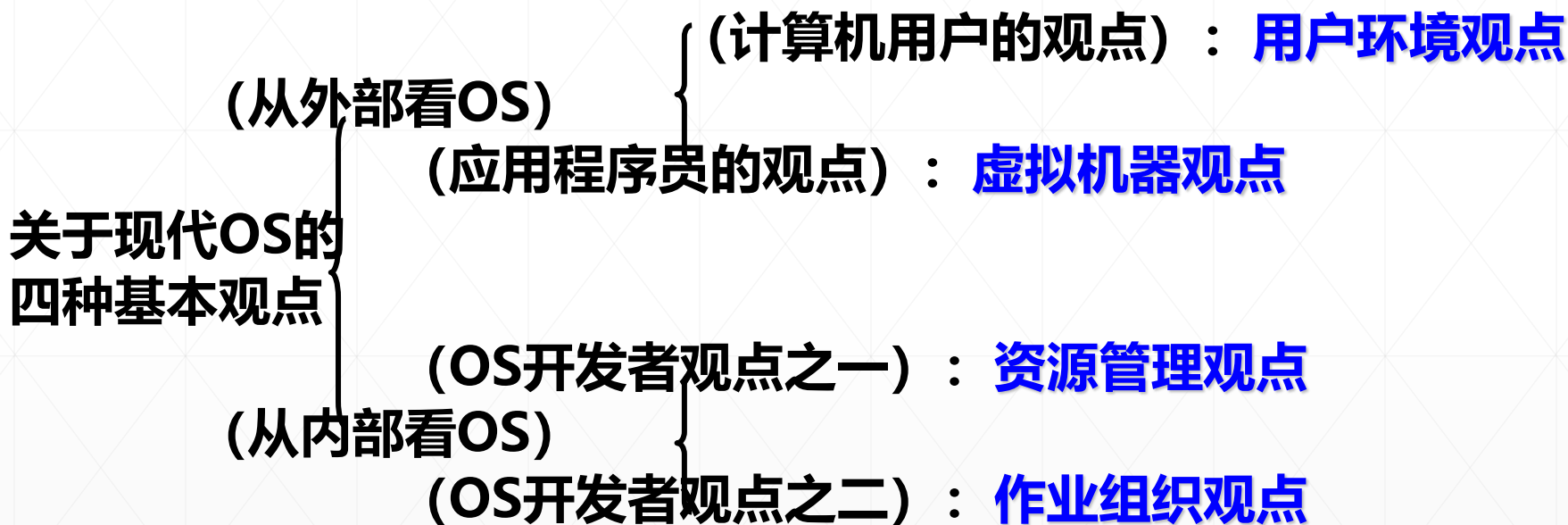
3. OS用作扩充机器





1.1.2 OS的作用

总结



1.2 操作系统的发展过程

- 无操作系统
 - 单道批处理操作系统
 - 多道批处理操作系统
 - 分时操作系统(time-sharing system)
 - 实时操作系统(real time system)
-



1.2.1 未配置操作系统的计算机系统

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。

缺点：

(1)用户独占全机

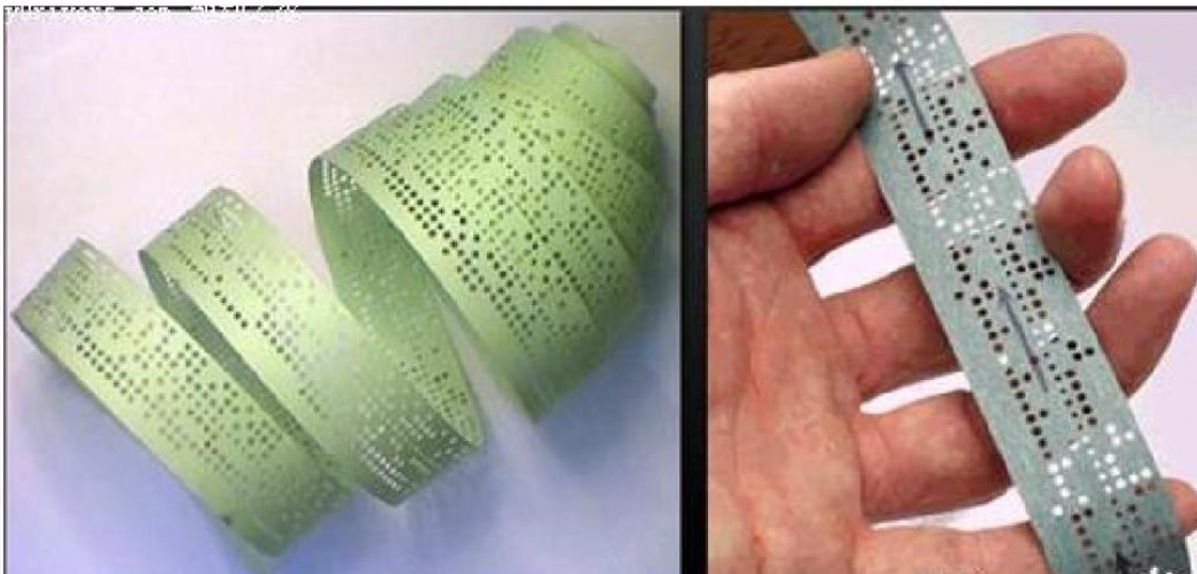
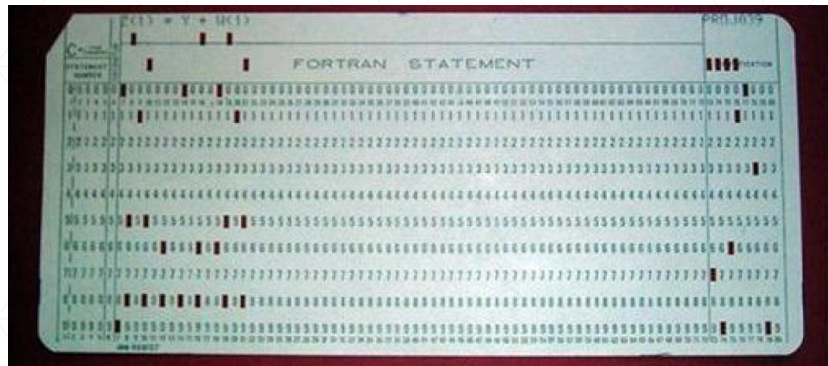
(2)CPU等待人工操作



1.2.1 未配置操作系统的计算机系统

1、人工操作方式

输入输出：穿孔卡片

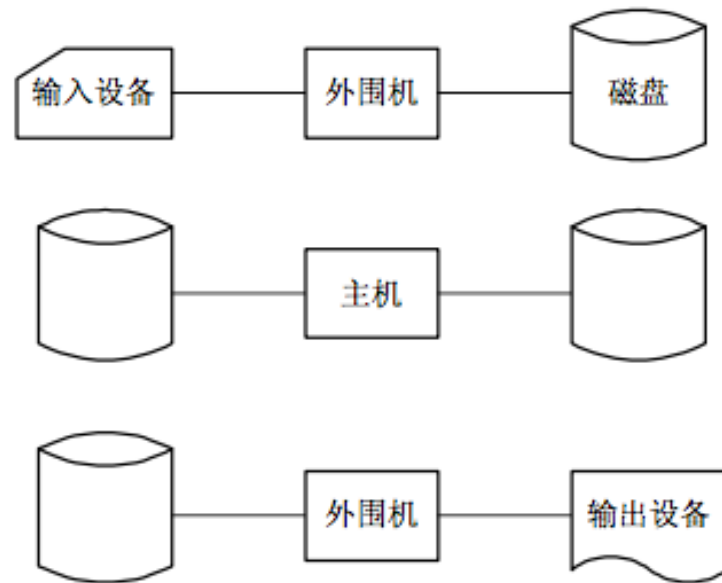




1.2.1 未配置操作系统的计算机系统

2、脱机输入输出方式

- 在外部计算机的控制下，实现输入输出。
- 主要解决了CPU与设备之间不匹配的矛盾，提高了I/O速度



脱机 I/O 示意图



1.2.2 单道批处理系统

系统对作业的处理都是成批进行的，且在**内存中始终仅存一道作业运行**，运行结束或出错，才自动调另一道作业运行，故称为**单道批处理系统**。

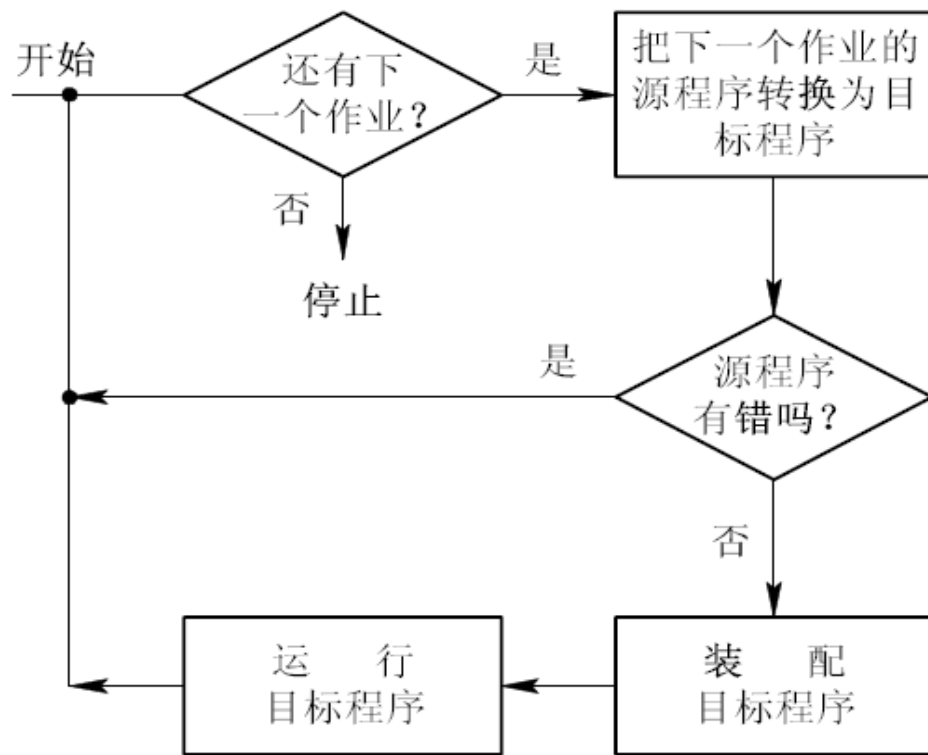


图1-3 单道批处理系统的处理流程



1.2.2 单道批处理系统

- 1、**单道批处理系统主要特征**：自动性、顺序性、单道性。
- 2、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。
- 3、单道批处理系统**主要缺点**：平均周转时间长，没有交互能力。



1.2.3 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:

- 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？

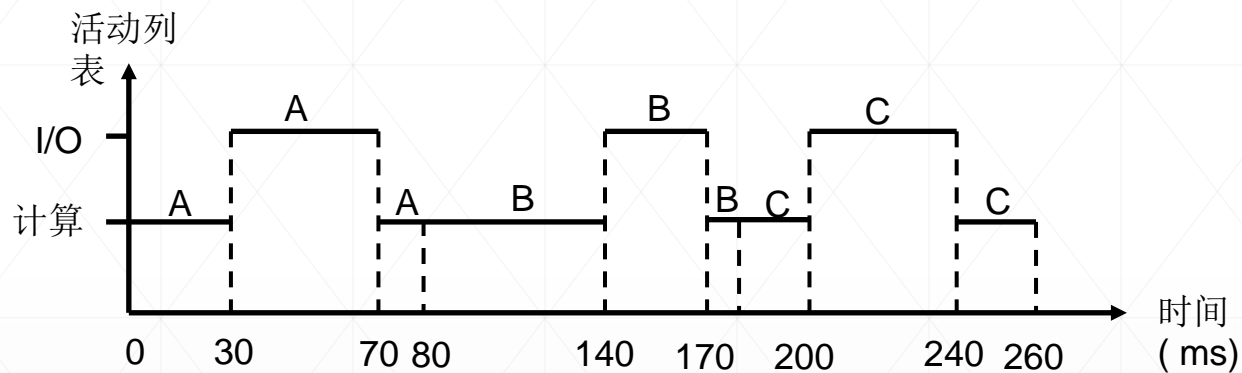
	A	B	C
计算	30ms	60ms	20ms
I/O	40ms	30ms	40ms
计算	10ms	10ms	20ms



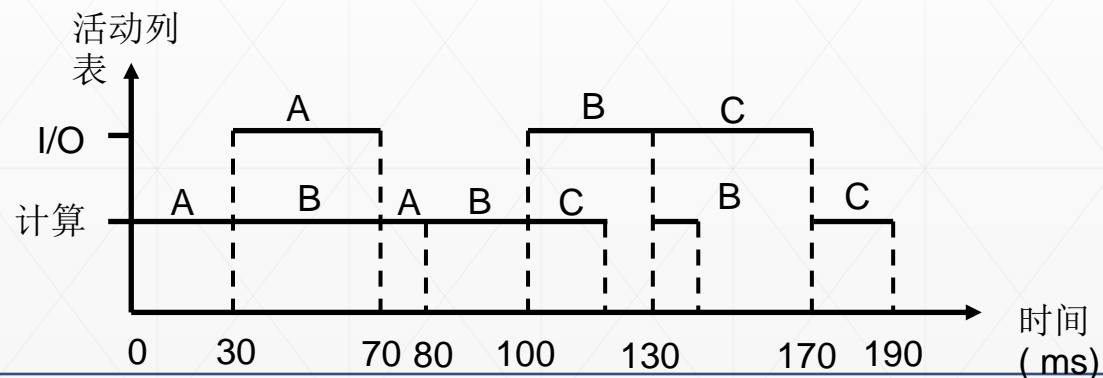
举例:

	A	B	C
计算	30ms	60ms	20ms
I/O	40ms	30ms	40ms
计算	10ms	10ms	20ms

■ 单道运行的甘特图



■ 多道运行的甘特图





1.2.3 多道批处理系统

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/O设备管理：I/O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。



1.2.4 分时操作系统

一、分时系统的产生

用户需求：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**



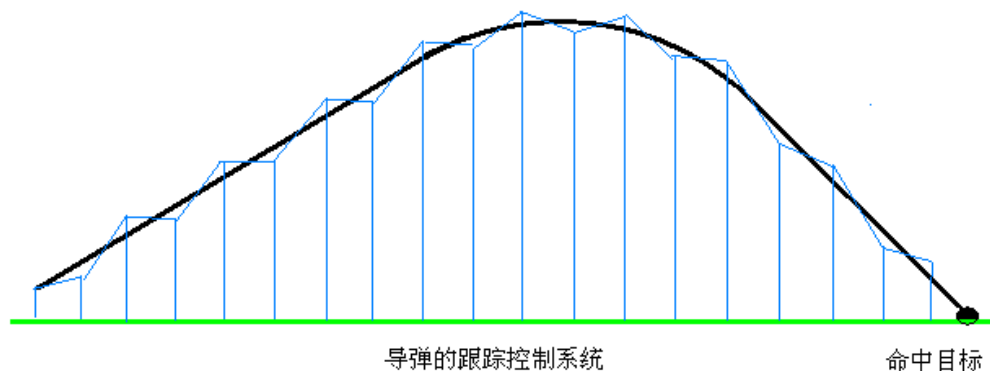
1.2.4 分时操作系统

- **特点：**
 - **多路性**：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



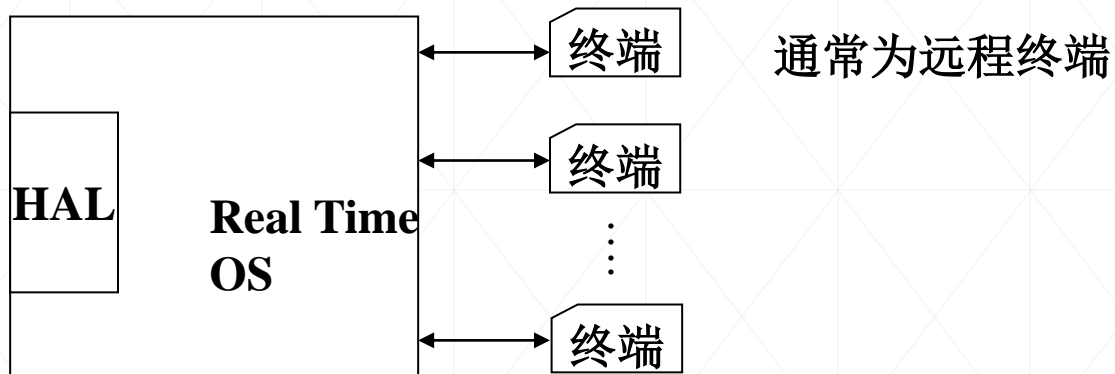
1.2.5 实时系统

- **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。**
 - 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。



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。



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**任务并发** (Task Concurrency：即多个任务在单个处理机上交替运行) 或多处理机系统中的**任务并行** (Task Parallelism：即多个任务在多个处理机上同时运行) 。



1.3.1 并发

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**临界资源**。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。**



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；



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OS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OS结构的更新换代。**



1.5.1 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。



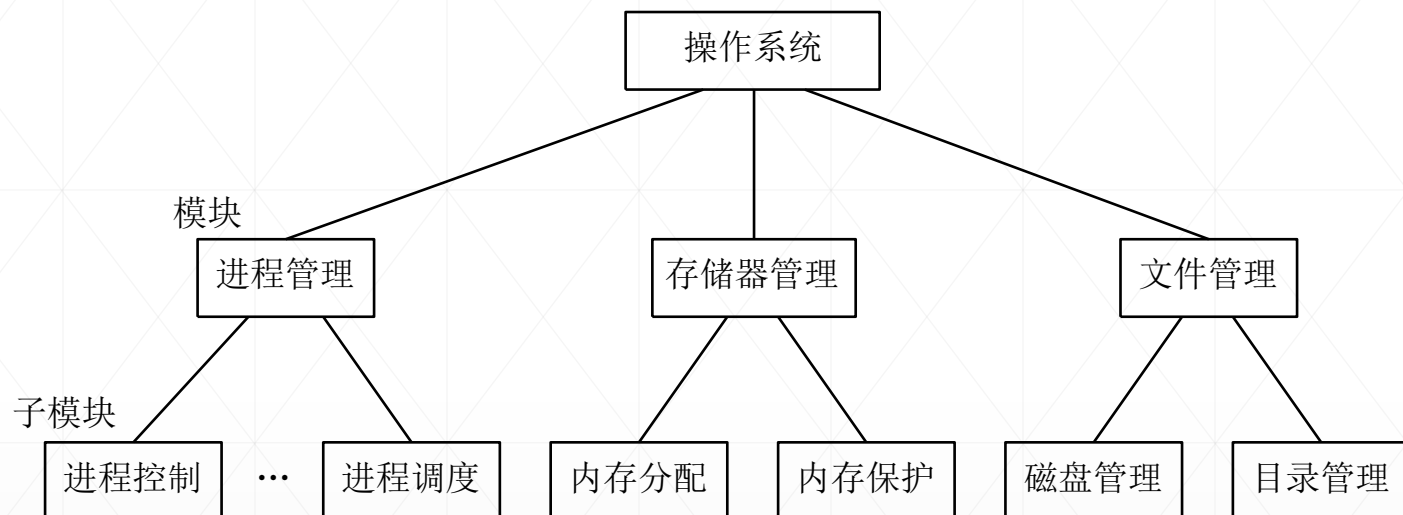
1.5.2 模块化OS结构

(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。



1.5.2 模块化OS结构



模块化操作系统结构



1.5.2 模块化OS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OS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。



1.5.3 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。



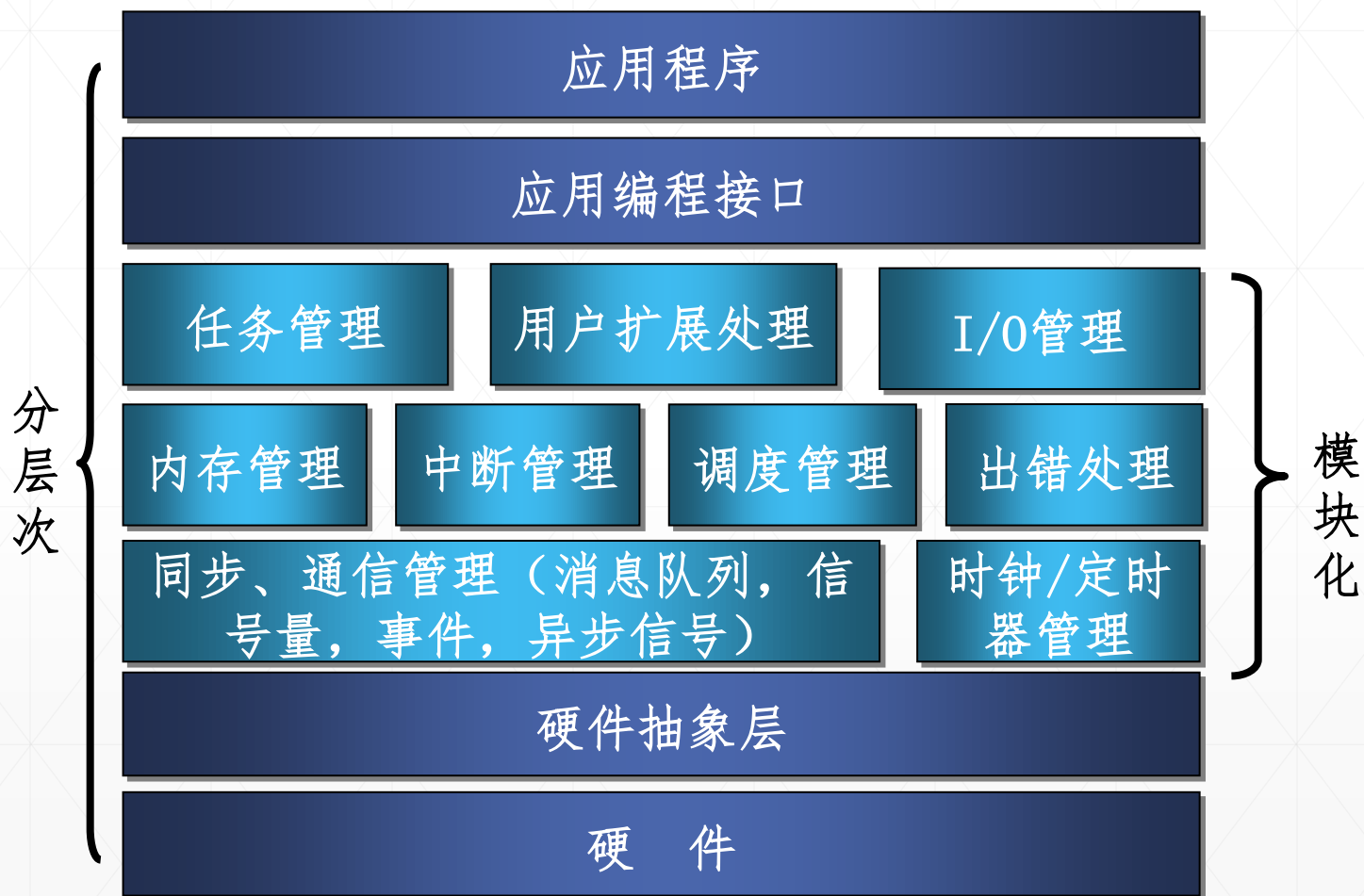
1.5.3 分层式OS结构

■ 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。



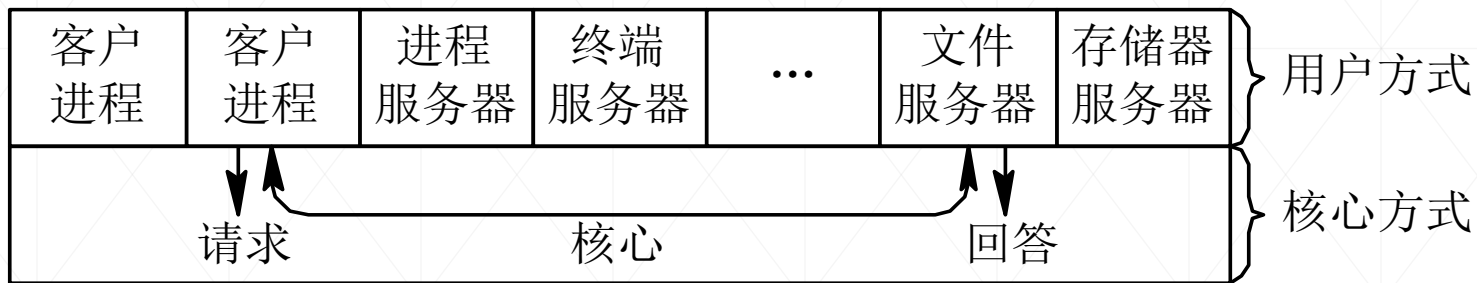
1.5.4 操作系统层次化体系结构



DeltaCORE的体系结构：层次 + 模块结构



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

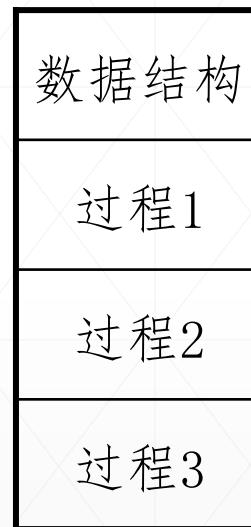


面向对象的程序设计技术 (OOP)

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。



一个对象的示意图



面向对象的程序设计技术 (OOP)

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



1.5.5 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**支持多处理机**运行的OS几乎全部都采用了微内核结构，如Mach OS, windows 2000



1.5.5 微内核OS结构

- 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/O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、 6、 8、 10、 11、 15、 16、 17、 18
