




操作系统

教材及参考书目

-  **教材：** 计算机操作系统（第四版）
-  汤小丹 梁红兵 哲凤屏 汤子瀛 编著
-  西安电子科技大学出版社

主要参考资料

- [1] 西尔伯斯查兹 《操作系统概念》 高等教育出版社
- [2] A. S. Tanenbaum 《操作系统：设计与实现》 电子工业出版社
- [3] H. M. Deitel 《操作系统原理基础》 北京科学出版社
- [4] 卢显良 《UNIX系统管理》 清华大学出版社
- [5] 张尧学 史美林 《计算机操作系统教程》 清华大学出版社
- [6] 庞丽萍 《操作系统原理》 华中科技大学出版社



第一章 操作系统引论

- ◆ 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 操作系统的基本特性
- ◆ 1.4 操作系统的主要功能
- ◆ 1.5 操作系统的结构设计



问题：你知道哪些操作系统？



目前主流的计算机操作系统

◆ Windows系列——微软公司

Windows是微软公司自DOS之后推出的图形用户界面的操作系统

◆ UNIX——1962年，贝尔实验室

◆ Linux——1991年，芬兰赫尔辛基大学

UNIX操作系统有版权，Linux没有版权
有图形用户界面和文本界面两种
有版权，只能在实验室内部使用

可多用户同时登陆，登录类似于UNIX操作系统

核心代码全部公开
主要用于大中型服务器

主要用于中小型服务器

1.1.1 操作系统的目标

1. 有效性
2. 方便性
3. 可扩充性
4. 开放性

1.1.2 操作系统的作用

1. OS作为用户与计算机硬件系统之间的接口
2. OS作为计算机系统资源的管理者
3. OS实现了对计算机资源的抽象

1.2 操作系统的发展过程

- ◆ 自世界上第一台计算机ENIAC(1800平方英尺)于1946年问世以来,计算机在运算速度、存储容量、外设功能、元件工艺及系统结构等方面都有了惊人的发展。
- ◆ 通常,人们按照计算机元件工艺的演变过程,将其发展划分为四个时代:

1.2.1 手工操作

◆ 手工操作过程：

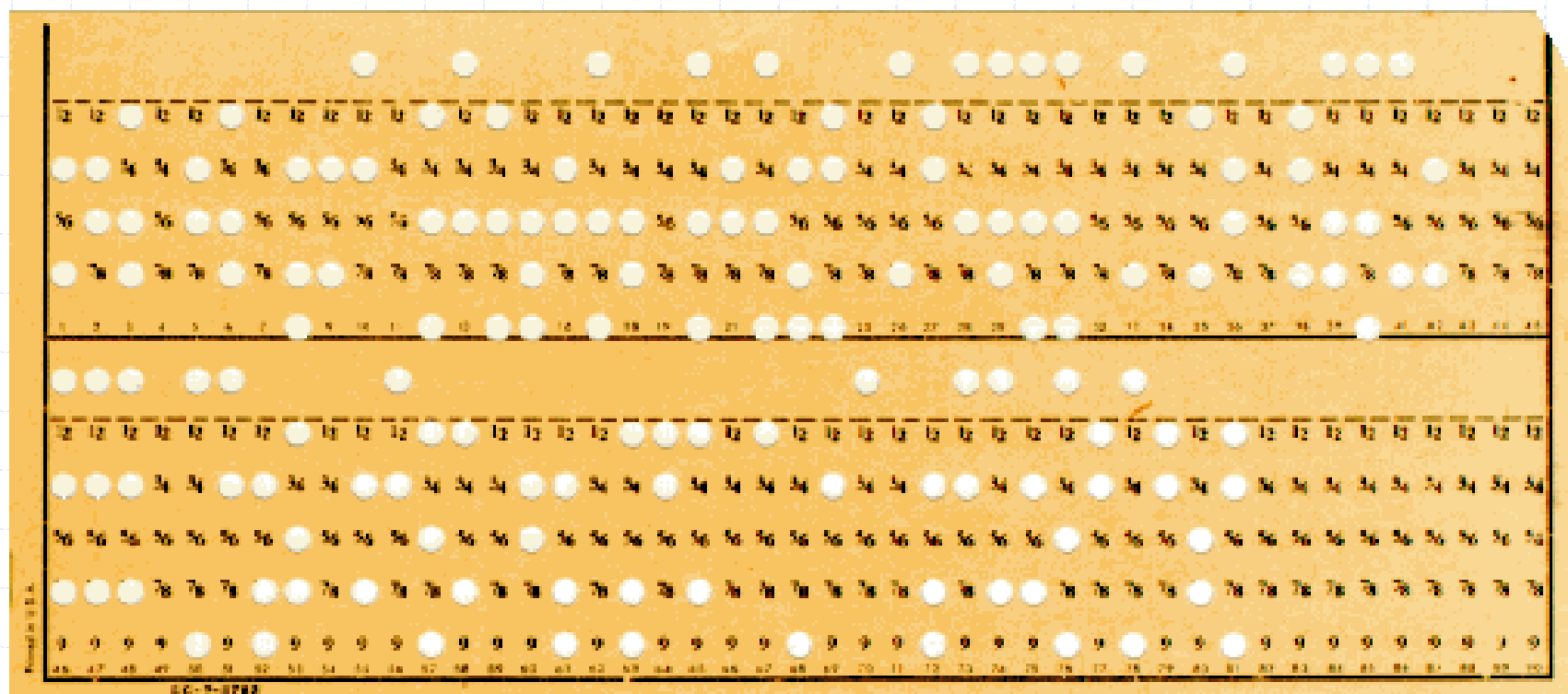
1. 先把程序纸带（或卡片）装上计算机
2. 然后启动输入机把程序输入计算机
3. 接着通过控制台开关启动程序运行。
4. 计算完毕，打印机输出计算结果，用户卸下并取走纸带（或卡片）。
5. 第二个用户上机，重复同样的步骤。

Early systems

◆ Early systems

- **No OS!** Programmer is also operator
- Large machines run from a console; programs loaded through switches and card readers

◆ 50年代早期出现了穿孔卡片
在卡片上然后读入计算机



1.2.2 单道批处理系统 (simple batch processing)

- ◆ 计算机发展的早期，没有任何用于管理的软件，所有的运行管理和具体操作都由用户自己承担，任何操作出错都要重做作业，CPU的利用率甚低。

批处理中的作业的组成

- ◆ “批”的含义：供一次加载的磁带或磁盘，通常由若干个作业组装成，在处理中使用一组相同的系统软件（系统带）。
- ◆ 说明：通常，把计算机完成用户算题任务所需进行的各项工作称为一道作业

Simple batch systems

- ◆ Simple batch systems were the **first real OS**
 - Setup time was a problem -> hire an operator
 - Operator ran related jobs **together**
 - OS was a simple program stored in one part of memory
 - ◆ Loads **a single job** from card reader into memory
 - ◆ Transfers control from one job to the next

1.2.3 多道批处理系统 (multiprogramming system)

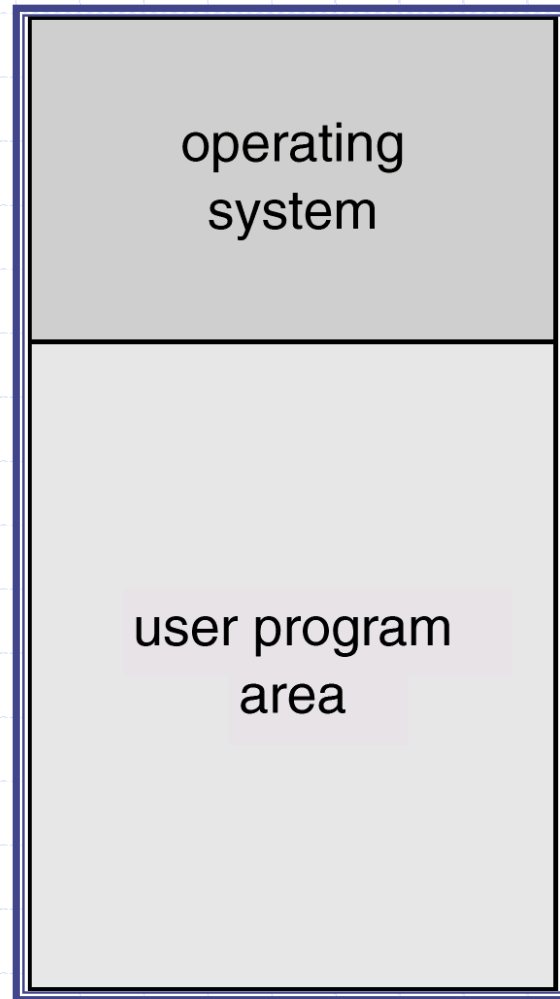
早期的批处理可能出现两种情况：

- ◆ 对于以**计算为主**的作业，输入输出量少，外围设备空闲；
- ◆ 对于以**输入输出为主**的作业，主机空闲。

问题的提出

- ◆在早期的单道批处理系统中，**内存中仅有单个作业在运行**，致使系统中仍有许多资源空闲，设备利用率低，系统性能较差。
- ◆如下图所示，当**CPU**工作时，外部设备不能工作；而外部设备工作时，**CPU**必须等待。

Memory Layout for a Simple Batch System



Only one
user job!

多道程序设计(Multiprogramming)

- ◆多道程序设计(Multiprogramming)是指允许多个程序同时进入一个计算机系统的主存储器并启动进行计算的方法。
- ◆多道程序合理搭配输入输出为主与计算为主程序交替运行，充分利用资源，提高系统效率。

多道程序的运行特点:

- ◆ 多道程序的运行特点:
- ◆ **多道**: 计算机内存中同时存放多道相互独立的程序。
- ◆ **宏观上并行运行**: 同时进入系统的几道程序都处于运行状态, 但都未运行完。
- ◆ **微观上串行运行**: 各作业轮流使用CPU, 交替执行。
- ◆ **实现技术**: 在当前运行的作业需做I/O处理时, CPU转而执行另一个作业。(I/O完成后是否立刻恢复执行, 要等到其他程序再次I/O时)。

单、多道程序运行示意图对比

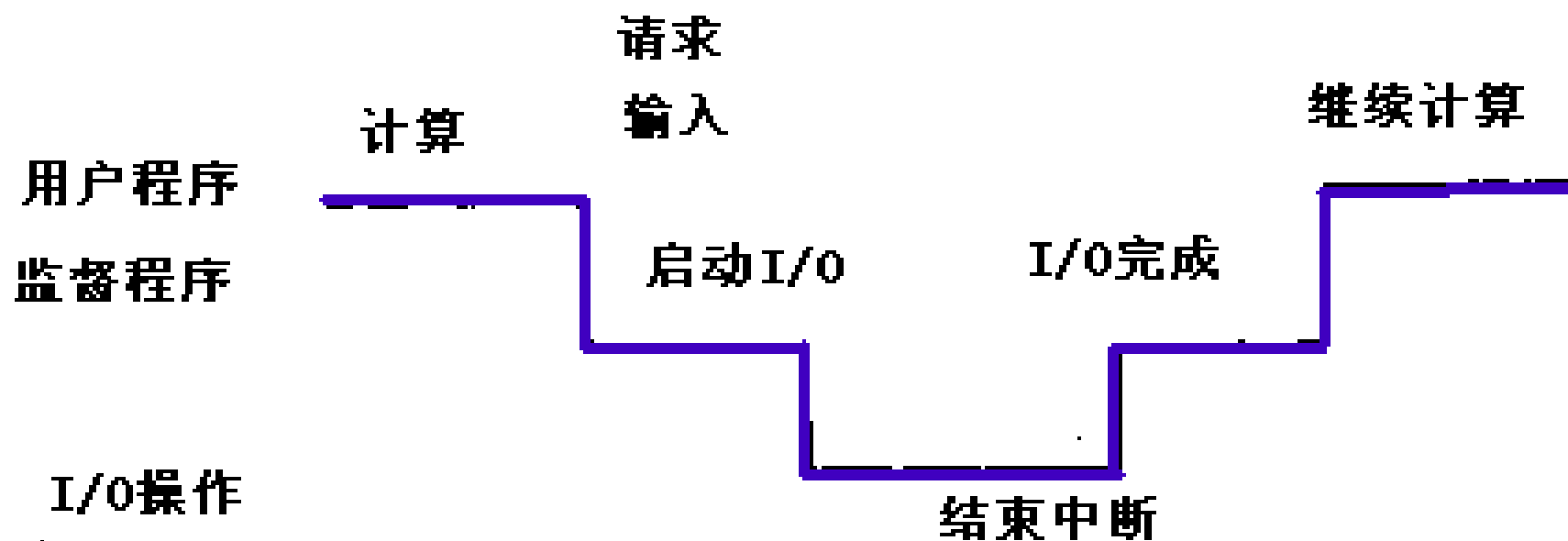


图 单道程序工作示例

单、多道程序运行示意图对比

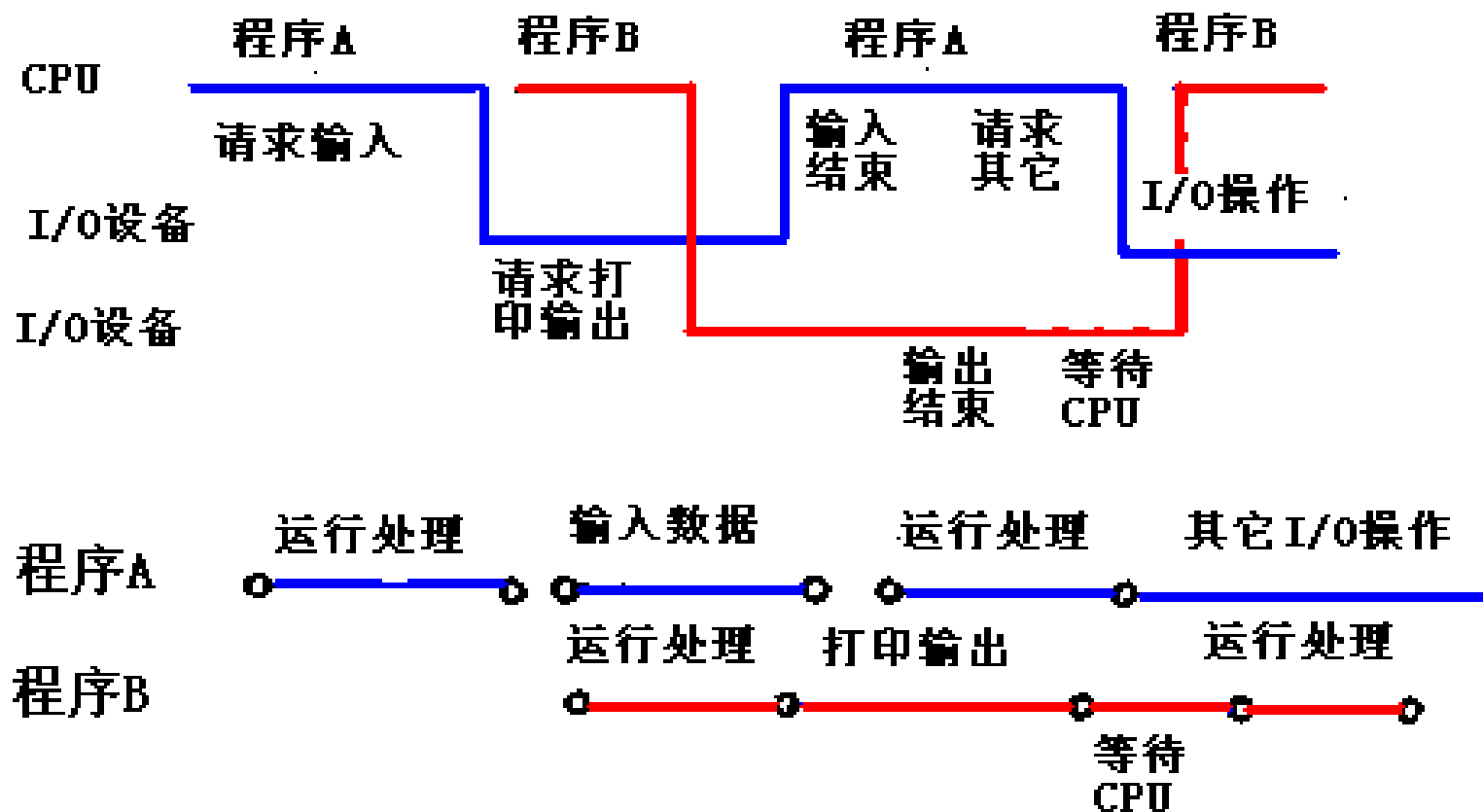
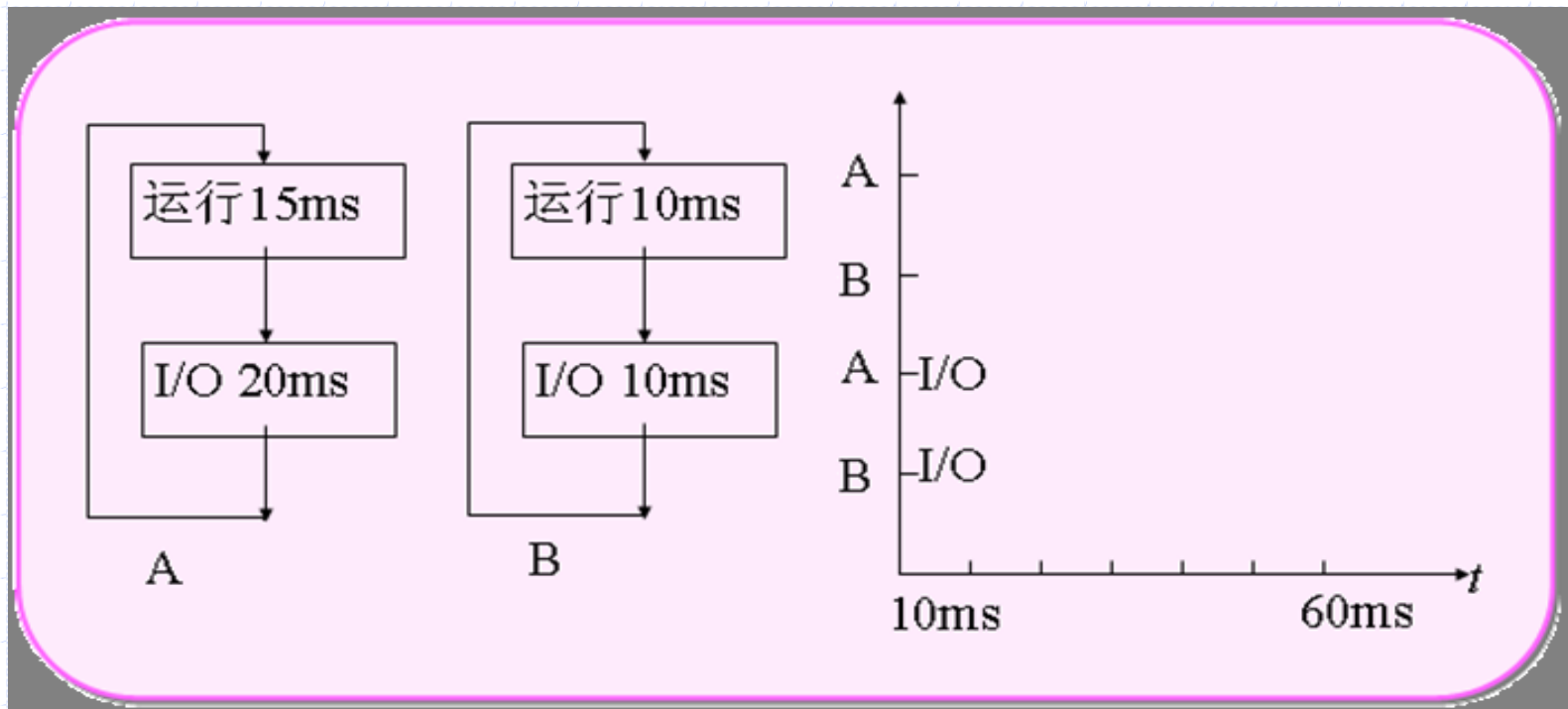


图 多道程序工作示例

思考作业-题

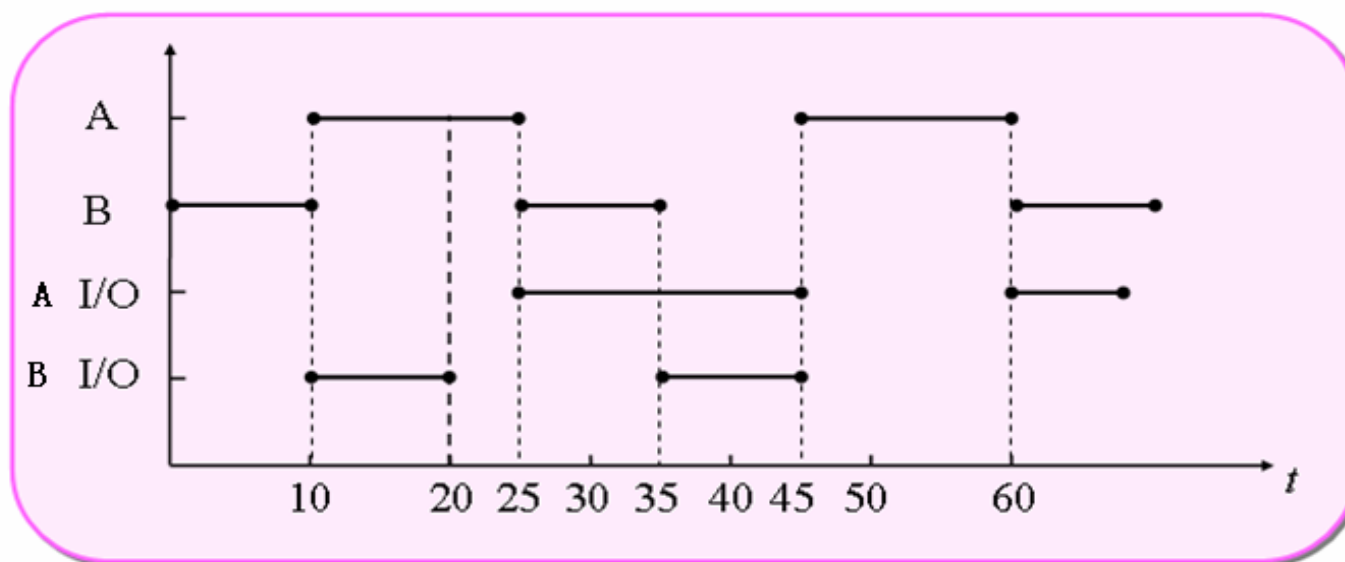
例题：有两道程序A、B，按下图以多道程序方式运行，要求在右图画出它们的运行轨迹，并计算在60ms内，CPU的利用率，假设起始时首先运行B，并允许忽略监督程序切换A、B的时间（**不考虑I/O的冲突**）。



思考作业-解

非剥夺式

解:



$$P_{\text{CPU}} = \frac{60 - (45 - 35)}{60} \times 100\% = \frac{50}{60} \times 100\% = 83.3\%$$

批处理系统总结

- ◆ 批处理系统的**主要优点**是解决了作业间的自动转换问题，提高了**CPU**的利用率，所以系统吞吐量大，资源利用率高
- ◆ **主要缺点**就是交互性差，一旦作业提交，其中间过程就很难控制。

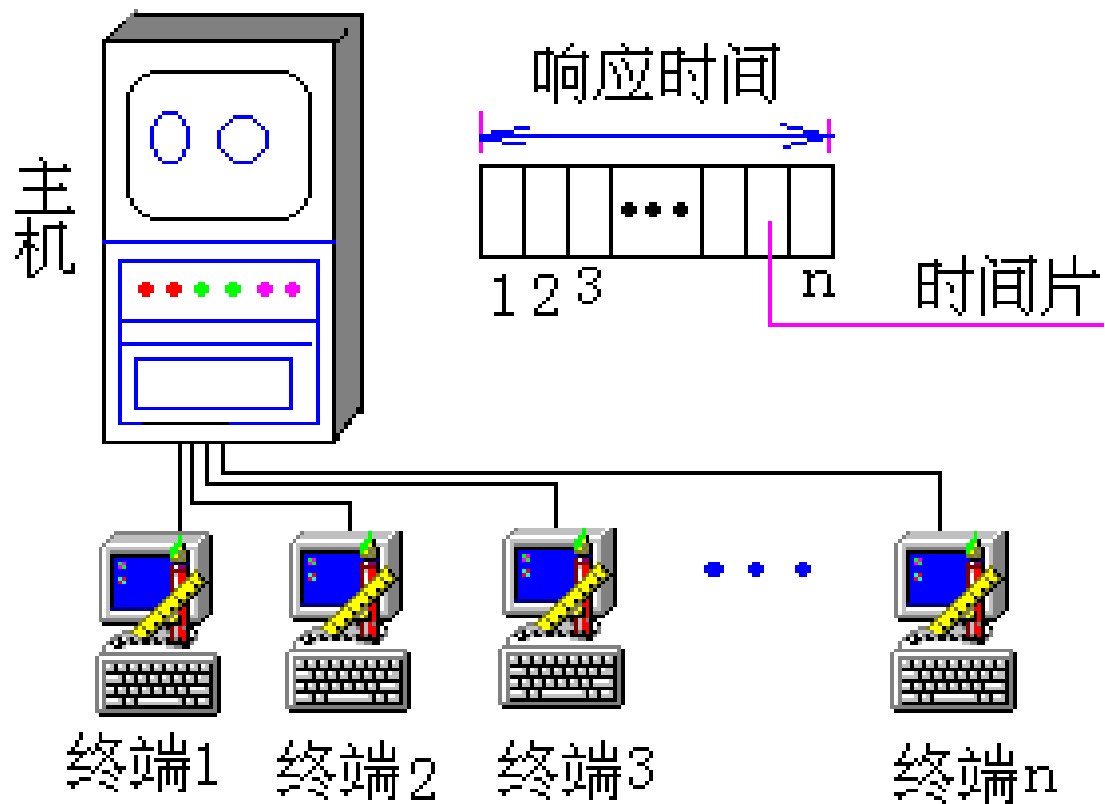
分时实际上是
“划分CPU的时间”

1.2.4 分时系统 (Time-sharing system)

- ◆ 分时(Time Sharing)是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。
- ◆ 分时技术：把处理机的运行时间分为很短的时间片，按时间片轮流把处理机分配给各联机作业使用。

分时系统的定义

- ◆ 分时操作系统（Time Sharing Operating System）是一种联机的多用户交互式的操作系统。一般采用时间片轮转的方式，使一台计算机为多个终端服务。对每个用户能保证足够快的响应时间，并提供交互会话能力。
- ◆ 如下图所示：



分时操作系统

分时系统的特点

- ◆ **交互性**：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。
- ◆ **多用户同时性**：多个用户同时工作，共享系统资源，提高了资源利用率。节省维护开支，可靠性高，**笨终端**——至今仍在使用的。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）联机使用。
- ◆ **独立性**：各用户独立操作，互不干扰。

分时系统的特点

- ◆也就是说：分时操作系统是一个联机的**(online)**多用户**(multiuser)**交互式**(interactive)**的操作系统。
- ◆多用户分时操作系统是当今计算机操作系统中使用最普遍的一类操作系统。
UNIX是当今最流行的一种多用户分时操作系统。

1.2.5 实时系统(real-time system)

- ◆产生背景：虽然多道批处理操作系统和分时操作系统获得了较佳的资源利用率和快速的响应时间，从而使计算机的应用范围日益扩大但它们难以满足实时控制和实时信息处理领域的需要。
- ◆于是，便产生了实时操作系统，目前典型的实时系统：实时控制系统、实时信息处理系统。

什么是实时系统？

- ◆ 实时操作系统（Real Time Operating System）是指当外界事件或数据产生时，能够接收并以足够快的速度予以处理，其处理的结果又能在规定的时间之内来控制监控的生产过程或对处理系统作出快速响应，并控制所有实行任务协调一致运行的操作系统。
- ◆ 实时系统要求有高可靠性和安全性，系统的效率则放在第二位

1.2.6 微机操作系统的发展

- ◆ 随着VLSI和计算机体系结构的发展, 以及应用需求的不断扩大, 操作系统仍在继续发展。由此先后形成了微机操作系统, 网络操作系统等。
- ◆ 下面对微机操作系统做简单介绍:

1.单用户单任务OS

1) CP/M

由Digital Research公司推出；
可配置在以intel 8080、8085、z80等8位芯片为基础的多种微机上；
在8位微机中占据统治地位。

2) MS-DOS

由IBM公司推出；
成为16位单用户单任务操作系统标准。

2.单用户多任务操作系统

- ◆只允许一个用户上机，但允许用户把程序分为若干个任务，使它们并发执行。
- ◆典型代表：微软公司推出的windows

WINDOWS的发展:

- ◆ Windows1.0、2.0基于Intelx86微处理器
- ◆ Windows3.0支持虚拟内存，windows3.1引入TrueType字体技术
- ◆ Windows95支持32位和16位混合操作系统
- ◆ Windows98支持FAT32文件系统及电源管理等
- ◆ Windows2000适用于服务器和工作站
- ◆ WindowsXP将家用型操作系统和商用型操作系统合二为一

3.多用户多任务OS

- ◆允许多个用户通过各自的终端使用同一台机器，共享主机系统中的各种资源，而每个用户程序又可进一步分为几个任务，使它们能并发执行。
- ◆典型代表：UNIX

UNIX及其变型

- ◆ 1969年诞生于美国的Bell实验室
- ◆ 变型的UNIX有：
SUN的Solaris, IBM的AIX, HP的UX

1.3 操作系统的概念和基本特性

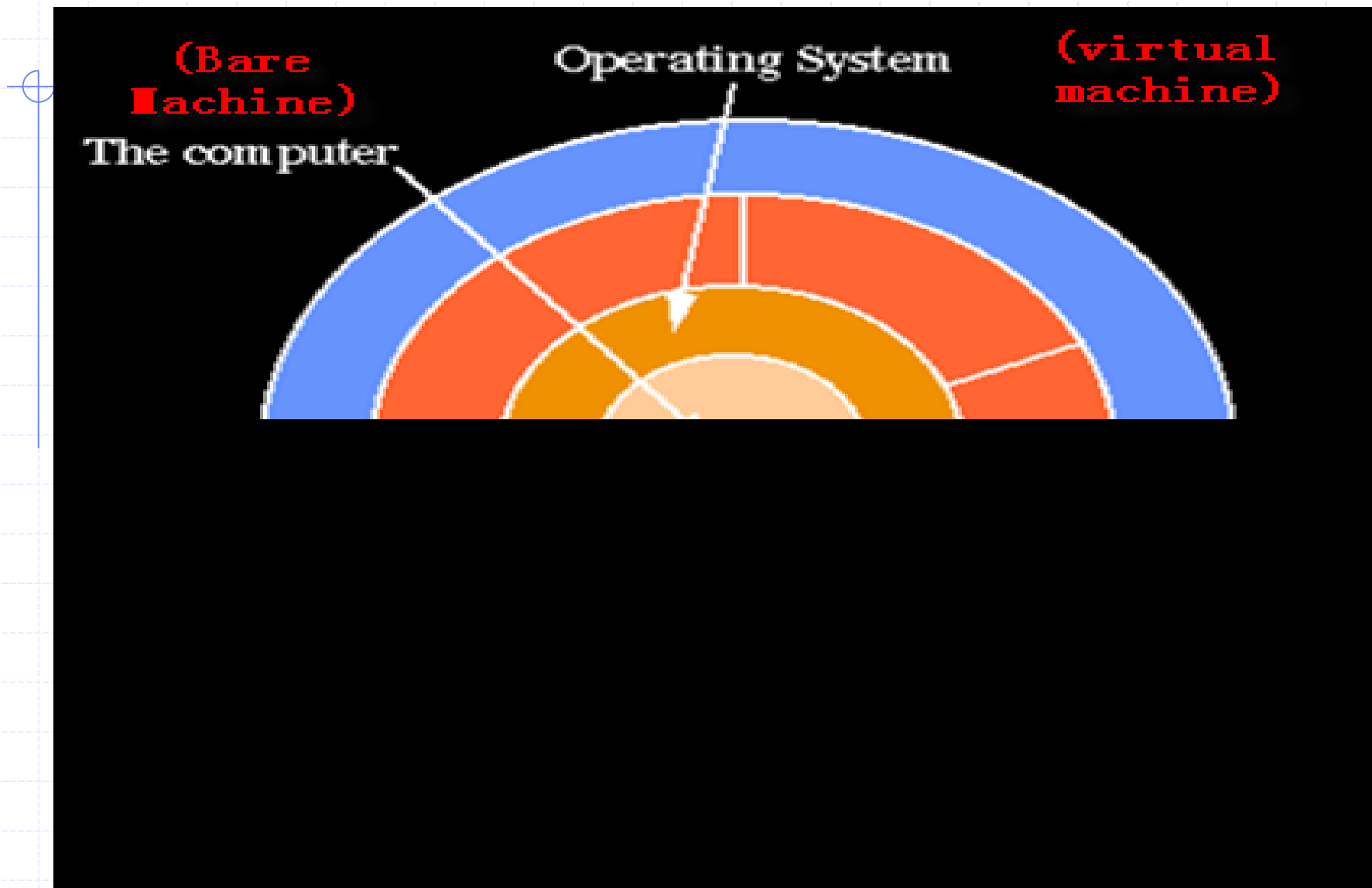
◆ 1.3.1 操作系统的概念

◆ 1.3.2 操作系统的基本特性

1.3.1 操作系统的概念

- ◆ 没有任何软件支持的计算机称为裸机(Bare Machine)，它仅仅构成了计算机系统的物质基础，而实际呈现在用户面前的计算机系统是经过若干层软件改造的计算机。
- ◆ 裸机在最里层，它的外面是操作系统，经过操作系统提供的资源管理功能和方便用户的各种服务功能 将裸机改造成功能更强，使用更方便的机器，通常称之为虚拟机(virtual machine)。
- ◆ 如下图所示：

计算机系统的分层观点



引入操作系统的目的

- ◆ 因此，引入操作系统的目的可从三方面来考察：
- ◆ **1.从系统管理人员的观点来看：** 操作系统是计算机资源的管理者。
- ◆ **2.从用户的观点来看：** 引入操作系统是为了给用户使用计算机提供一个良好的界面，以使用户无需了解许多有关硬件和系统软件的细节，就能方便灵活地使用计算机。
- ◆ **3.从发展的观点看：** 引入操作系统是为了给计算机系统的功能扩展提供支撑平台，使之在追加新的服务和功能时更加容易和不影响原有的服务与功能。

计算机系统的层次

- ◆ **硬件层**提供了基本的可计算性资源，包括处理器、寄存器、存储器，以及可被使用的各种I/O设施和设备，是操作系统和上层软件赖以工作的基础。
- ◆ **操作系统层**对计算机硬件作首次扩充和改造，主要完成资源的调度和分配，信息的存取和保护，并发活动的协调和控制等许多工作。
- ◆ 操作系统是其它软件的运行基础，并为编译程序和数据库系统等系统程序的设计者提供了有力支撑。

计算机系统的层次

- ◆ **系统程序层**的工作基础建立在操作系统改造和扩充过的机器上，利用操作系统提供的扩展指令集，可以较为容易地实现各种各样的语言处理程序、数据库管理系统和其它系统程序。
- ◆ **应用层**解决用户不同的应用问题，应用程序开发者借助于程序设计语言来表达应用问题，开发各种应用程序，既快捷又方便。
- ◆ 最终用户则通过**应用程序与计算机交互来解决他的应用问题。**

操作系统的概念：

- ◆有效地管理计算机系统资源，
- ◆合理地管理计算机系统的工作流程，
- ◆方便用户的程序的集合。

1.3.2 操作系统的基本特性

- ◆ 以多道程序设计为基础的现代操作系统具有以下几个主要特征：
- ◆ 并发性（Concurrence）
- ◆ 共享性（Sharing）
- ◆ 异步性（Asynchronism）或称不确定性（Nondeterministic）
- ◆ 虚拟性（Virtual）

1、并发性

- ◆ 并发性（Concurrence）是指两个或两个以上的运行程序在同一时间间隔段内同时执行。
- ◆ 操作系统是一个并发系统，并发性是它的重要特征，它应该具有：处理多个同时执行的程序的能力。

并行与并发的不同：

- ① 并行性是指两个或多个事件在同一时刻发生
- ② 并发性是指两个或多个事件在同一时间间隔内发生

什么是进程

- ◆ 进程的定义：一个具有一定独立功能的程序在某个数据集合上的一次动态执行过程。
- ◆ 进程是指在系统中能独立运行并作为资源分配的基本单位
- ◆ 简单的可以理解为一个程序的动态执行

2、共享性（sharing）

- ◆ 共享指操作系统中的资源（包括硬件资源和信息资源）可被多个并发执行的进程所使用。
- ◆ 出于经济上的考虑，一次性向每个用户程序分别提供它所需的全部资源不但是浪费的，有时也是不可能的。现实的方法是让多个用户程序共用一套计算机系统的所有资源，因而必然会产生共享资源的需要。

资源共享的方式

- ◆ 资源共享的方式可以分成两种：
- ◆ 第一种是互斥共享。系统中的某些资源如打印机、磁带机、卡片机，虽然它们可提供给多个进程使用，但在同一时间内却只允许一个进程访问这些资源。
- ◆ 这种同一时间内只允许一个进程访问的资源称临界资源，许多物理设备，以及某些数据和表格都是临界资源，它们只能互斥地被共享。

资源共享的方式

- ◆ 第二种是同时访问。系统中的还有许多资源，允许同一时间内多个进程对它进行访问，这里“同时”是宏观上的说法。典型的可供多进程同时访问的资源是磁盘。

2、共享性

- ◆与共享性有关的问题是资源分配、信息保护、存取控制等，必须要妥善解决好这些问题。
- ◆共享性和并发性是操作系统两个最基本的特征，它们互为依存：

3、异步性

- ◆ 异步性（Asynchronism）亦称不确定性（**Nondeterministic**）
- ◆ 在多道程序环境中，允许多个进程并发执行，由于资源有限而进程众多，多数情况，进程的执行不是一贯到底，而是“走走停停”。
- ◆ 例如，一个进程在CPU上运行一段时间后，由于等待资源满足或事件发生，它被暂停执行，CPU转让给另一个进程执行。

理解模拟

- ◆ 并发--“大家都前进了”;
- ◆ 共享--“一件东西大家用”;
- ◆ 异步性--"你走我停"。

(四) 虚拟性

- ◆ **虚拟性**是指操作系统中的一种管理技术，它是把物理上的一个**实体变成逻辑上的多个对应物**，显然，前者是实际存在的而后者是虚构假想的。
- ◆ 通过多道程序和分时使用**CPU**技术，物理上的一个**CPU**变成逻辑上的多个**CPU**

(四) 虚拟性

- ◆通过Spooling技术可把物理上的一台独占设备变成逻辑上的多台虚拟设备，
- ◆通过窗口技术可把物理上的一个屏幕变成逻辑上的多个虚拟屏幕，
- ◆虚拟存储器则是把物理上的多个存储器(主存和辅存)变成逻辑上的一个(虚存)的例子

1. 4操作系统功能

◆操作系统有如下几个基本功能:

◆处理机管理

◆存储管理

◆设备管理

◆文件管理

◆用户接口

（一）处理机管理

- ◆在传统的多道程序系统中，处理机的分配和运行都是以进程为基本单位，因而对处理机的管理可归结为对进程的管理：
- ◆进程控制；
- ◆进程同步；
- ◆进程通信；
- ◆调度。

(二) 存储管理

- ◆ 存储管理的主要任务是管理存储器资源，为多道程序运行提供有力的支撑。存储管理的主要功能包括：
 - ◆ (1) 存储分配与回收：存储管理将根据用户程序的需要给它分配存储器资源，并在其使用完毕后，回收之。
 - ◆ (2) 存储保护：存储管理要把各个用户程序相互隔离起来互不干扰，更不允许用户程序访问操作系统的程序和数据，从而保护用户程序存放在存储器中的信息不被破坏。

(二) 存储管理

- ◆ (3) 地址映射（变换）：进程逻辑地址到内存物理地址的映射。这样程序员无需知道自己的程序分配到内存的什么具体物理地址，仅仅知道自己的逻辑地址即可，体现了存储的无关性。
- ◆ (4) 内存扩充：由于物理内存容量有限，难于满足用户程序的需求，存储管理还应该能从逻辑上来扩充内存存储器，为用户提供一个比内存实际容量大得多的编程空间，方便用户的编程和使用，实现**虚拟内存**。

(三) 设备管理

◆ 设备管理的**主要任务**是：

- ① 管理各类外围设备，完成用户提出的I/O请求，加快I/O信息的传送速度，发挥I/O设备的并行性，提高I/O设备的利用率；
- ② 以及提供每种设备的设备驱动程序和中断处理程序，**向用户屏蔽硬件使用细节**。

（四）文件管理

- ◆ 上述三种管理是针对计算机**硬件资源**的管理。
- ◆ 文件管理则是对系统的**软件资源**的管理。
- ◆ 在现代计算机中，通常把程序和数据以文件形式存储在外存储器上，供用户使用。
- ◆ 这样，外存储器上保存了大量文件，对这些文件如不能采取良好的管理方式，就会导致混乱或破坏，造成严重后果。

（四）文件管理

- ◆ 为此，在操作系统中配置了文件管理，它的主要任务是：
 - ① 对用户文件和系统文件进行有效管理，实现按名存取；
 - ② 实现文件的共享、保护和保密，保证文件的安全性；
 - ③ 提供给用户一套能方便使用文件的操作和命令。

(五) 用户接口

- ◆ 为了使用户能灵活、方便地使用计算机和操作系统，操作系统还提供了一组友好的用户接口，包括：
 - ◆ **1) 程序接口：**提供一组广义指令(或称**系统调用**、程序请求)供用户程序和其他系统程序调用。
 - ◆ **2) 命令接口：**提供一组控制操作命令(或称作业控制语言，或象UNIX中的**shell**命令语言)供用户去控制自己作业的运行。
 - ◆ **3) 图形接口。**

1.5 操作系统的结构设计

- ◆ 操作系统规模庞大，结构复杂，其质量的好坏直接关系到计算机系统能否有效可靠的运行。
- ◆ 从操作系统的发展过程来看，操作系统的结构有如下几种：

1. 无结构OS

◆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。这种OS是无结构的。

◆整体式系统，是最常用的组织方式，又叫模块组合法，但常被人们形容为“一锅粥”，其结构其实就是“无结构”。

Monolithic (one unstructured program)

2. 模块化结构OS

- ◆ 将OS按其功能划分为若干个具有一定独立性和大小的模块，每个模块具有某方面的管理功能，并规定好各模块之间的接口，使各模块间能通过该接口实现交互。
- ◆ 关键问题是模块的划分和规定好模块之间的接口

3. 分层式结构OS

- ◆ 所谓分层结构，即是把操作系统划分为内核和若干模块（或进程）组成，这些模块（或进程）排列成若干层，各层之间只能是单向依赖关系，不构成循环。

分层式结构OS

- ◆ 层次结构是如此构造起来的，从裸机A0开始，在它上面添加一层软件，使机器的功能得以扩充，形成了一台功能比原来机器要强的虚拟机A1。
- ◆ 又从A1出发，在它上面添加一层新的软件，把A1改造成功能更强的虚拟机A2。
- ◆ 就这样“添加——扩充——再添加”，由底向上地增设软件层，每一层都在原来虚拟机的基础上扩充了原有的功能，于是最后实现一台具有所需操作系统各项功能的虚拟机。

4.微内核结构

- ◆ 由于Mach操作系统成功应用了微内核(Microkernel) 技术，微内核的概念近来得到了广泛的关注。
- ◆ 这种方法给系统带来了高度的灵活性、模块性和可移植性，已经在微机、工作站和服务器操作系统中得到广泛使用。

相关概念——处理机的状态

◆为什么要区分处理机的态?

◆答：操作系统是计算机系统的最重要的系统软件，为了能正确的进行管理和控制，其本身是不能被破坏的。为此系统应能建立一个保护环境，采用的办法就是区分处理机的状态。

相关概念—— 处理机的状态

◆在系统中有两类程序在运行——

◆1.管理程序

◆2.用户程序

相关概念——处理机的状态

◆ 1. 管理程序

◆ 其担负的任务是：

- a. 管理和控制者
- b. 为用户提供服务
- c. 负责管理和分配系统资源

◆ 2. 用户程序

◆ 其担负的任务是：

- a. 被管理和被控制者
- b. 所需资源必须向操作系统提出请求
- c. 享受系统提供的服务

什么是处理机的态?

- ◆处理机的态——处理机的工作状态，即处理机当前正在执行哪一类程序，依此来区分处理机的状态。
- ◆计算机系统在执行上述两种不同类型的程序时，就应该有不同的权限
- ◆而为此根据对资源和机器指令的使用权限，把处理执行时的两种工作状态分为核态和用户态。

关于“态”的几个概念

- ◆ **核态 (Kernel Mode)**，又称**管态或系统态**：CPU执行管理程序时所处的状态。
- ◆ 在此状态下允许CPU使用全部资源和全部指令，其中包括一组特权指令（如涉及外设的I/O、改变处理机状态、修改存储保护的指令），实现对系统资源的分配与管理，为用户提供使用外部设备的服务。

关于“态”的几个概念

- ◆ **用户态(User Mode):** 又叫目态或常态。
用户程序执行时CPU所处的状态。
- ◆ 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。

关于“态”的几个概念

- ◆ 当用户程序执行时，如果需要请求操作系统服务，则必然由用户态转为核态，即由操作系统得到控制权，在核态下执行其管理程序。
- ◆ 服务完毕后，由核态返回用户态，继续让用户程序执行。

一道思考题

◆ 在计算机系统中，为什么要区分管态与目态？

【解答】操作系统是计算机系统中最重要的系统软件，为了能正确地进行管理和控制，其本身是不能被破坏的。因此，系统采用了区分处理机状态的办法，为操作系统程序建立一个保护环境。这样，用户程序只能在目态下运行，只能执行非特权指令，只能访问自己的存储区，从而保护了操作系统程序的正常运行。

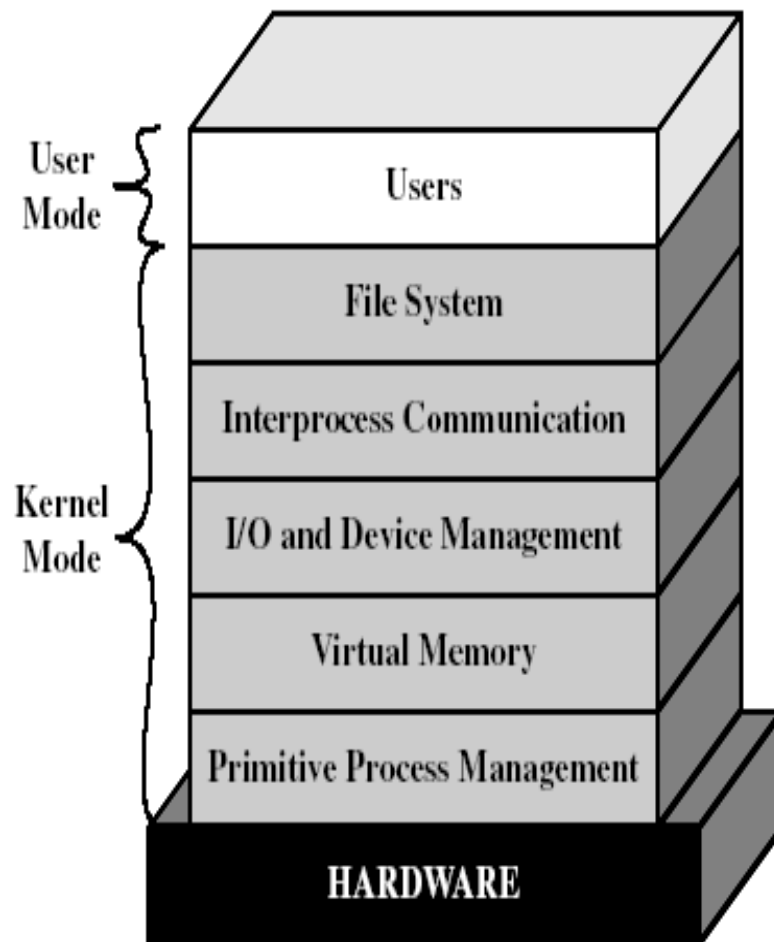
3.微内核结构

- ◆现代操作系统的一个趋势是将这种把代码移到更高层次的思想进一步发展，从操作系统中去掉尽可能多的东西，而只留一个最小的核心。
- ◆通常的方法是将大多数操作系统功能由用户进程来实现。过去已成为操作系统传统的许多服务，现在成了与微内核交互的外部子系统。

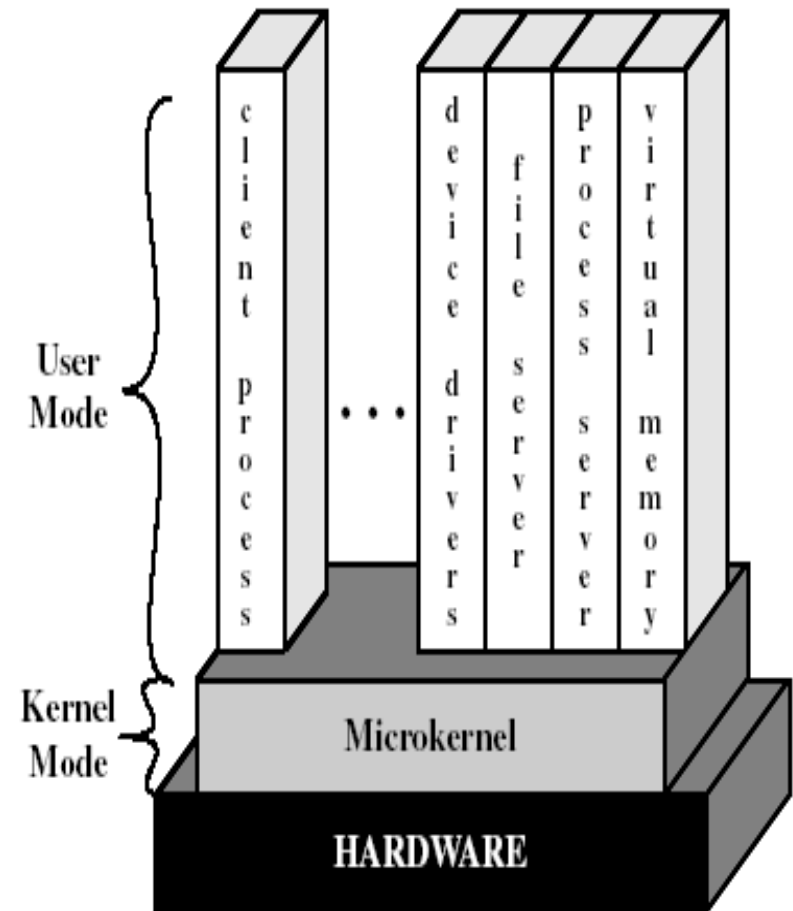
微内核思想

- ◆ 微内核用水平型代替传统的垂直型结构操作系统。
- ◆ 微内核**基本思想是**：内核中仅存放那些最基本的核心操作系统功能。其它服务和应用则建立在微内核之外，在用户模式下运行。

Layered vs. Microkernel Architecture

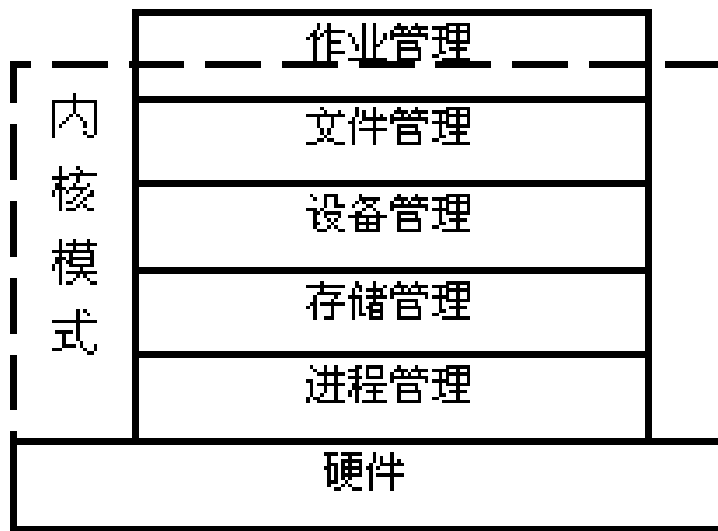


(a) Layered kernel

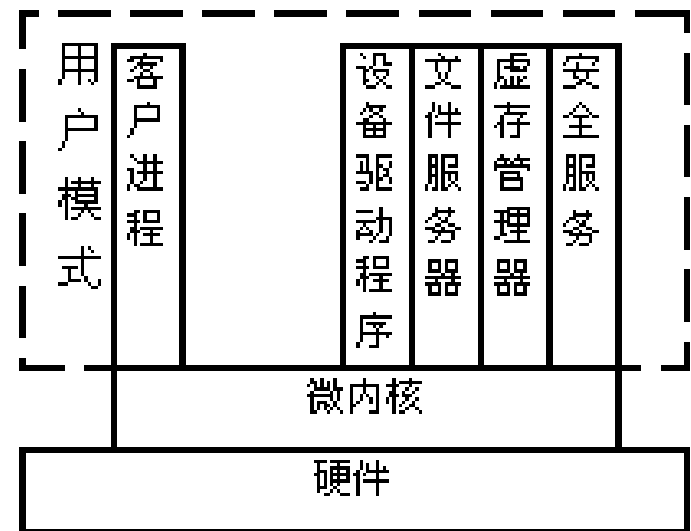


(b) Microkernel

分层结构内核和微内核结构对比



1) 分层结构内核

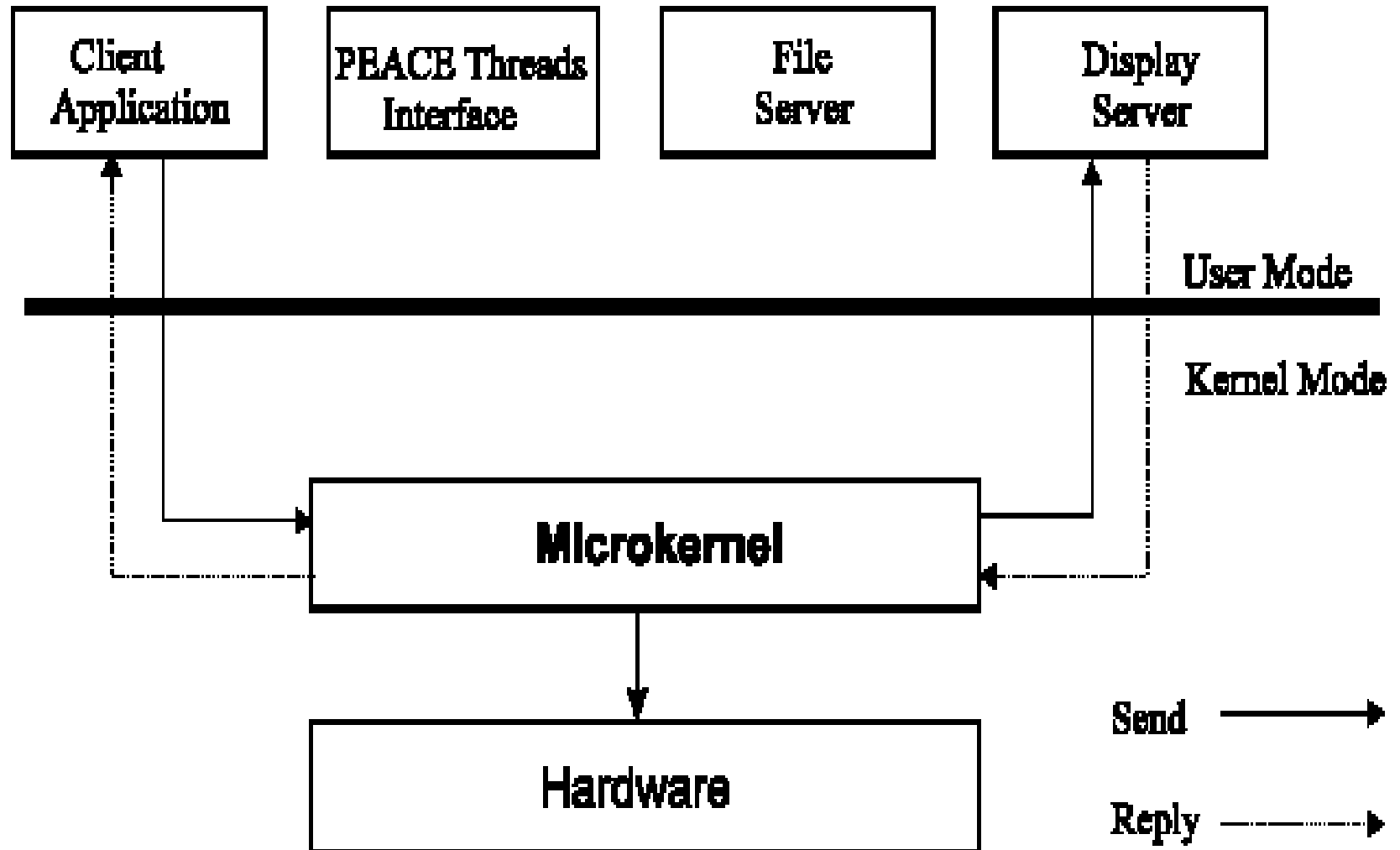


2) 微内核

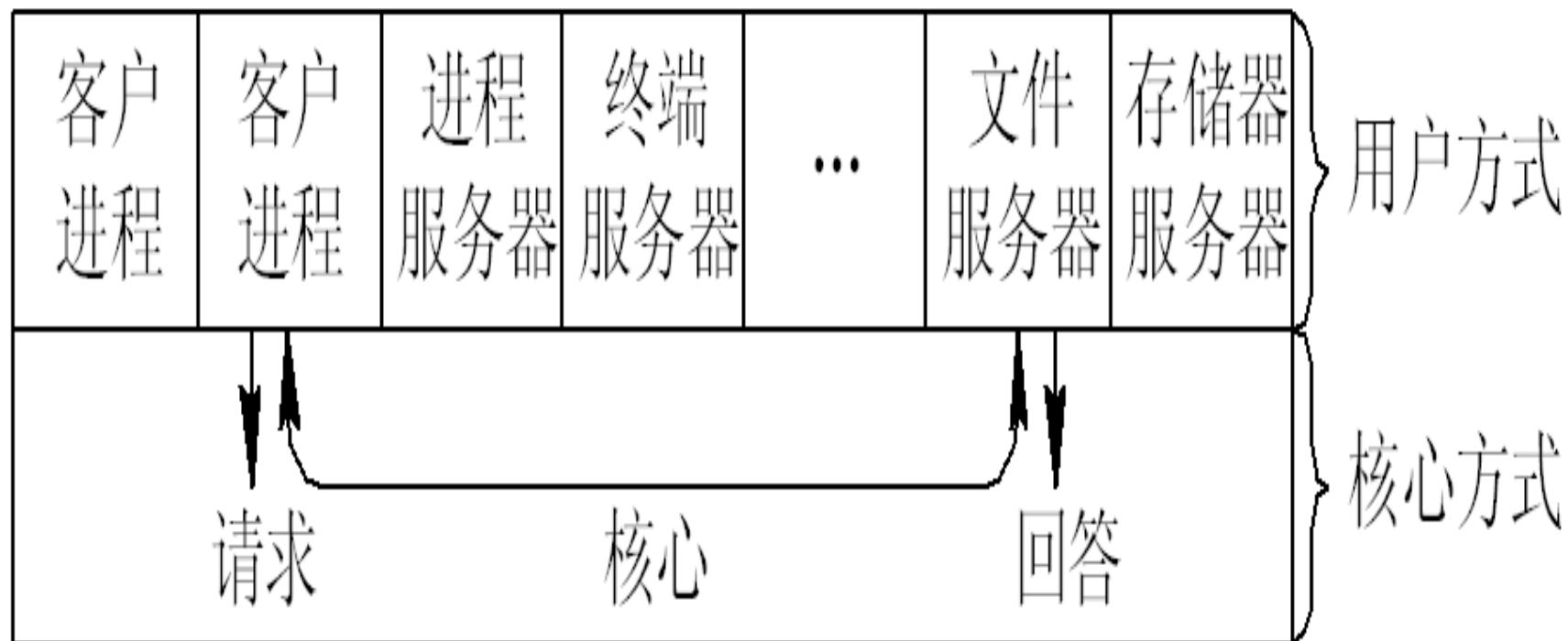
分层结构内核和微内核结构对比

- ◆ 分层结构操作系统的内核很大，互相之间调用关系复杂。
- ◆ 微内核结构操作系统则把大量的操作系统功能放到内核外实现，这些外部的操作系统构件是作为服务过程来实现的，它们之间的信息相互均借助微内核提供的消息传送机制实现。
- ◆ 这样，微内核起消息交换功能，它验证消息，在构件之间传送消息，并授权存取硬件。

Microkernel Operating System



单机环境下的客户/服务器模式



微内核 客户/服务器模式的优点

(1) 提高了系统的灵活性和可扩充性。

- modular design.
- easy to add services.

(2) 提高了OS的可靠性:

-small microkernel can be rigorously tested.

(3) 可运行于分布式系统中。

- message are sent without knowing what the target machine is.

微内核 客户/服务器模式的优点

◆(4)支持面向对象的操作系统（OOOS）

- components are objects with clearly defined interfaces that can be interconnected to form software.

The slide features a light blue dashed grid background. A solid blue horizontal line spans the width of the slide, with a small blue circle at its left end. Another solid blue horizontal line is positioned below the main text, with a small blue circle at its right end. A vertical blue line runs down the right side of the slide, intersecting the lower horizontal line.

本讲完毕