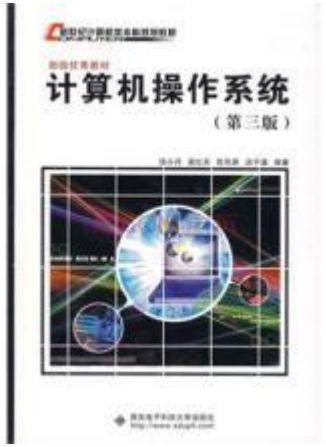
操作系统(OS, Operating System)

主讲教师: 杨志豪

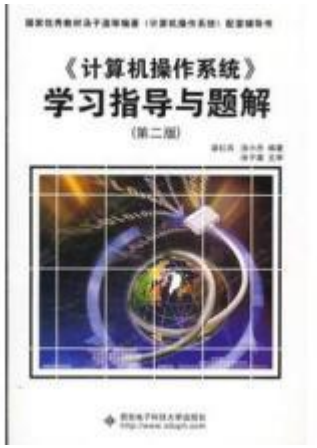
电话：13190114398
信箱：yangzh@dlut.edu.cn

教材及参考书

教材：
- 计算机操作系统（第三版），汤小丹等， 西安电子科技大学出版社
- 《计算机操作系统》学习指导与题解(第二版），梁红兵，汤小丹等，西安电子科技大学出版社



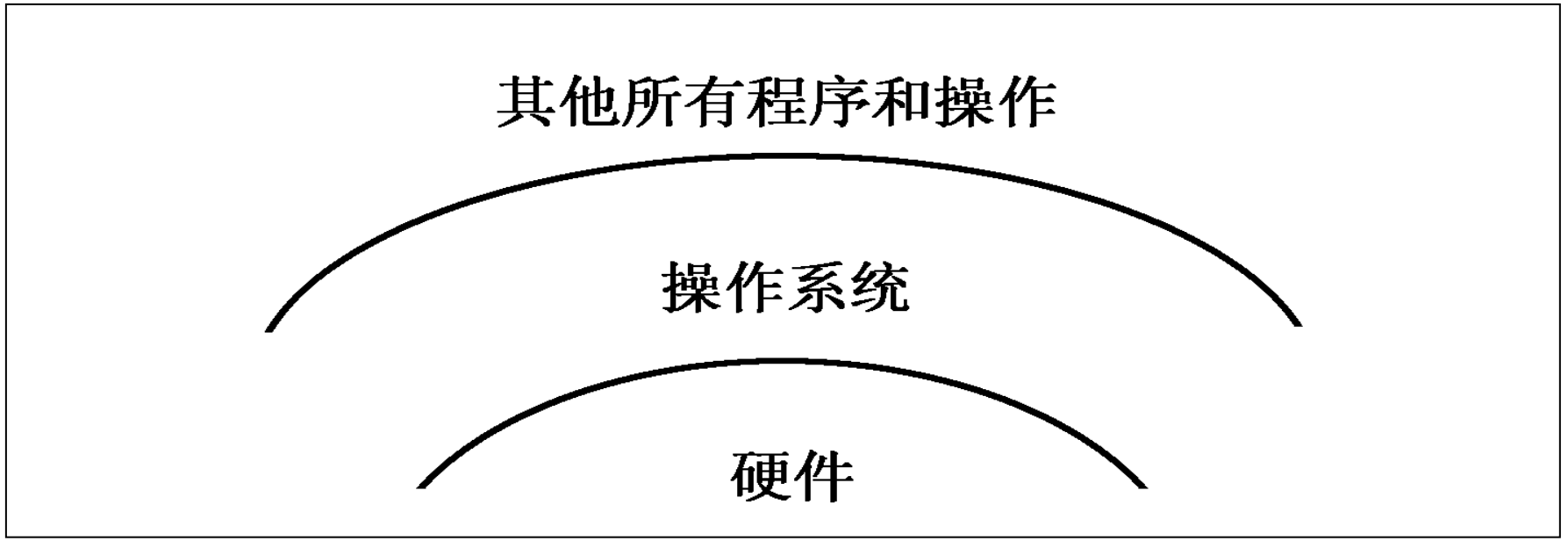
参考书：
- 计算机操作系统教程，张尧学，史美林 (第二版)，清华大学出版社
- 现代操作系统，陈向群等译 ，机械工业出版社



课程成绩评定

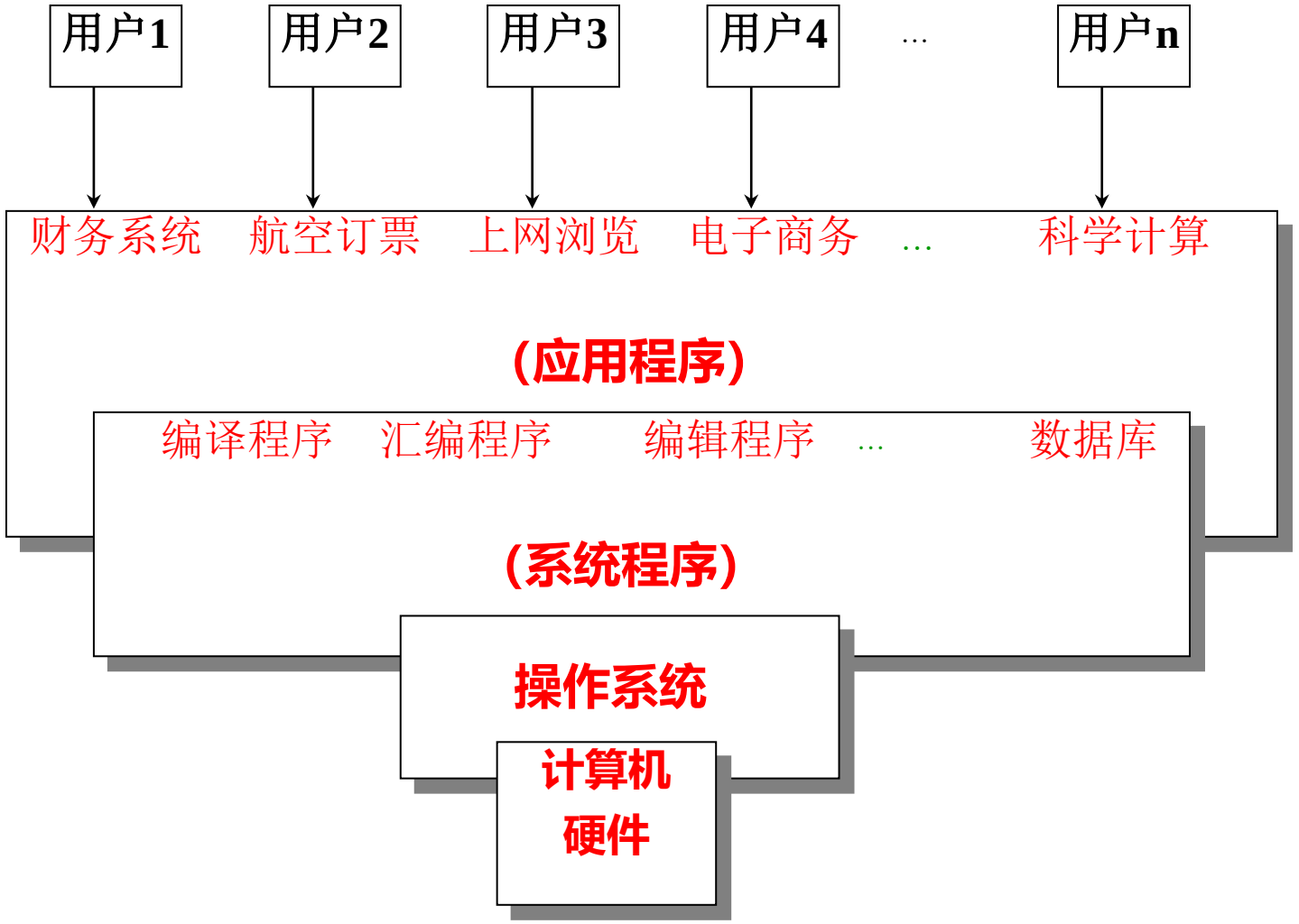
- **平时成绩：提问+作业（30分）**
- **期末考试（70分）**

操作系统在计算机系统中的地位（1）



- 计算机系统由硬件和软件组成
- 操作系统：
  - 是在硬件基础上的第一层软件
  - 是其它软件和硬件之间的接口

操作系统在计算机系统中的地位（2）



引入操作系统的目的

- 方便用户（提供接口，控制计算机工作流程，减轻用户编程负担，用户专心于应用本身）
- 提高资源利用率（并发执行程序，资源全面管理，统一分配，调度）
- 提高计算机工作效率（并发、共享）
- 安全（提供安全控制机制）

第一章 操作系统引论

- 1.1 操作系统的定义、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 操作系统结构设计
- 1.6 openEuler简介

操作系统的定义

资源管理的观点

1

●操作系统是控制和管理计算机的软、硬件资源，合理地组织计算机的工作流程，以及方便用户的程序集合。

2

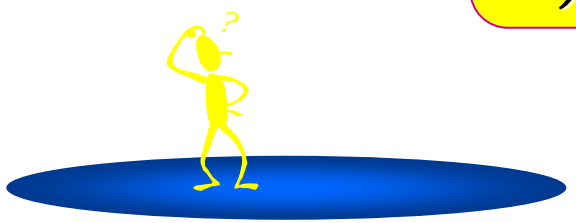
3

本质

用户的观点

●操作系统是配置在计算机硬件上的第一层软件，是对硬件系统的第一次扩充。

机器扩充的观点



1.1 操作系统的目标和作用

- 1.1.1 操作系统的目标

  通常在计算机硬件上配置的OS，其目标有以下几点：

  - 有效性
  - 方便性
  - 可扩充性
  - 开放性

1.1.1 操作系统的目标

1．有效性

在早期(20世纪50～60年代)，由于计算机系统非常昂贵，操作系统最重要的目标无疑是有效性。操作系统的有效性可包含如下两方面的含意：

- 提高系统资源利用率。诸如CPU、I/O设备、内存及外存。
- 提高系统的吞吐量。操作系统还可以通过合理地组织计算机的工作流程，而进一步改善资源的利用率，加速程序的运行，缩短程序的运行周期，从而提高系统的吞吐量。

1.1.1 操作系统的目标

2．方便性

一个未配置OS的计算机系统是极难使用的，因为计算机硬件只能识别0和1这样的机器代码。用户要直接在计算机硬件上运行自己所编写的程序，就必须用机器语言书写程序。

如果我们在计算机硬件上配置了OS，用户便可通过OS所提供的各种命令来使用计算机系统。

3．可扩充性

OS必须具有很好的可扩充性，才能适应计算机硬件、体系结构（多处理器、计算机网络）以及应用发展的要求：微内核结构、客户服务器模式。

1.1.1 操作系统的目标

4．开放性

开放性是指系统能遵循世界标准规范，特别是遵循开放系统互连(OSI)国际标准。凡遵循国际标准所开发的硬件和软件，均能彼此兼容，可方便地实现互连。

自20世纪80年代以来，由于计算机网络的迅速发展，为使来自不同厂家的计算机和设备能通过网络加以集成化，并能正确、有效地协同工作，实现应用的可移植性和互操作性，要求操作系统必须提供统一的开放环境，具有开放性。

1.1.2 操作系统的作用

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

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

OS处于用户与计算机硬件系统之间，用户通过OS，能够方便、快捷、安全、可靠地操纵计算机硬件和运行自己的程序。

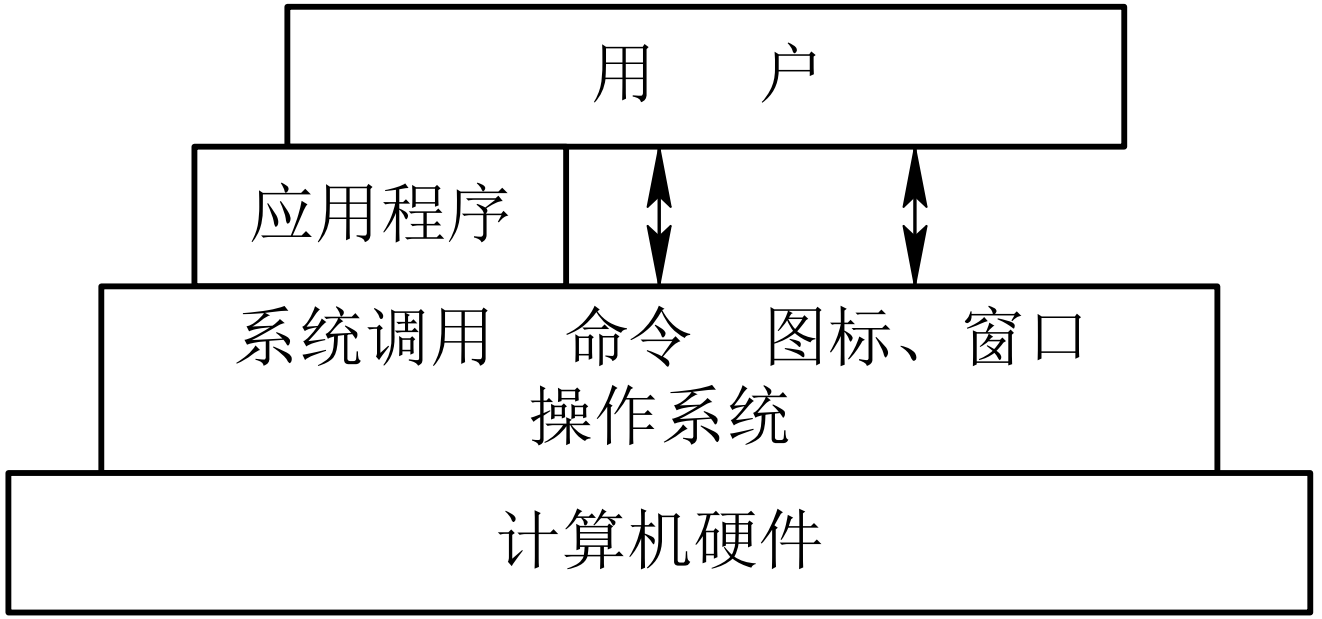


图1-1　OS作为接口的示意图

1.1.2 操作系统的作用

接口类型

- 命令方式。这是指由OS提供了一组联机命令接口，以允许用户通过键盘输入有关命令来取得操作系统的服务，并控制用户程序的运行。
- 图形、窗口方式。这是当前使用最为方便、最为广泛的接口，它允许用户通过屏幕上的窗口和图标来实现与操作系统的通信，并取得它的服务。
- 系统调用方式。OS提供了一组系统调用，用户可在自己的应用程序中通过相应的系统调用，取得它的服务。

1.1.2 操作系统的作用

2．OS作为计算机系统资源的管理者

在一个计算机系统中，可将资源分为四类：处理器、存储器、I/O设备以及信息(数据和程序)。相应地，OS的主要功能也正是针对这四类资源进行有效的管理：

- 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。

1.1.2 操作系统的作用

- 处理器上可执行的指令分为：
  - 特权指令：只能由操作系统使用的指令，如果允许用户随便使用，有可能使系统系统陷入混乱
  - 非特权指令：用户只能使用非特权指令

- 处理器状态划分为：
  - 管态：（管理态）操作系统管理程序运行的状态，可以使用各种指令(特权和非特权指令）；可使用所有资源；并具有改变处理器状态的能力。
  - 目态：（用户态）用户程序运行的状态,只能使用非特权指令

处理机状态保证了特权指令的正确使用，把OS与用户程序区别开来

1.1.2 操作系统的作用

3．OS实现了对计算机资源的抽象

人们在裸机上覆盖上一层I/O设备管理软件，由它来实现对I/O设备操作的细节，并向上提供一组I/O操作命令，用户可利用它来进行数据输入或输出，而无需关心I/O是如何实现的。

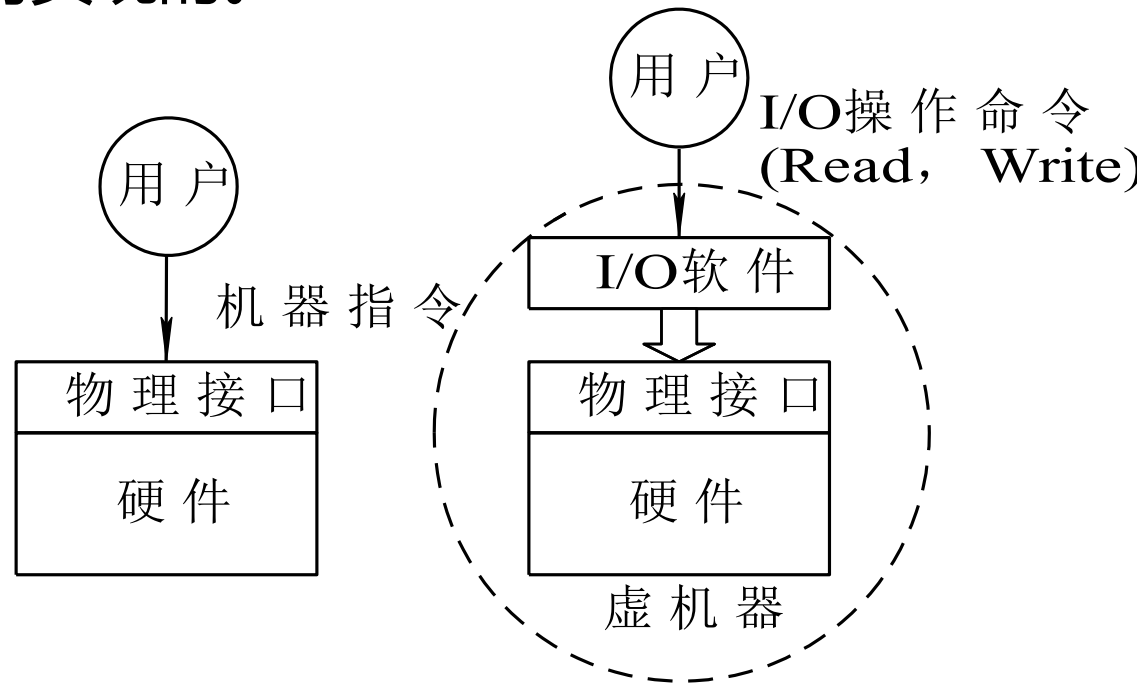


图1-2 I/O软件隐藏了I/O操作实现的细节

1.1.2 操作系统的作用

- 通常把覆盖了上述软件的机器称为扩充机器或虚机器,由该层软件实现了对计算机硬件操作的第一个层次的抽象。
- 为了方便用户使用文件系统，人们又在第一层软件上再覆盖上一层用于文件的管理软件，同样由它来实现对文件操作的细节，并向上提供一组对文件进行存取操作的命令。该层软件实现了对硬件资源操作的第二个层次的抽象。
- 而当人们又在文件管理软件上再覆盖一层面向用户的窗口软件后，用户便可在窗口环境下方便地使用计算机，形成一台功能更强的虚机器。

1.1.3 推动操作系统发展的主要动力

四个动力：

- 不断提高计算机资源的利用率：多道批处理系统、SPOOLing系统、虚拟存储器技术
- 方便用户：人机交互的分时系统、图形用户界面
- 器件的不断更新换代：CPU、外设
- 计算机体系结构的不断发展：多处理器、计算机网络

1.2 操作系统的发展过程

1946年 第一代计算机上没有操作系统 —— 手工操作

1958年 第二代计算机上有了监控系统 —— 单道批处理

1964年 第三代计算机上操作系统得到极大发展 —— 多道批处理
分时OS
实时OS
通用OS

1974年 第四代计算机操作系统向多元化方向发展 —— 单用户OS
网络OS
分布式OS
嵌入式OS
多处理机OS
… …

1.2.1　无操作系统的计算机系统

1．人工操作方式

从第一台计算机诞生(1945年)到20世纪50年代中期的计算机，属于第一代计算机。

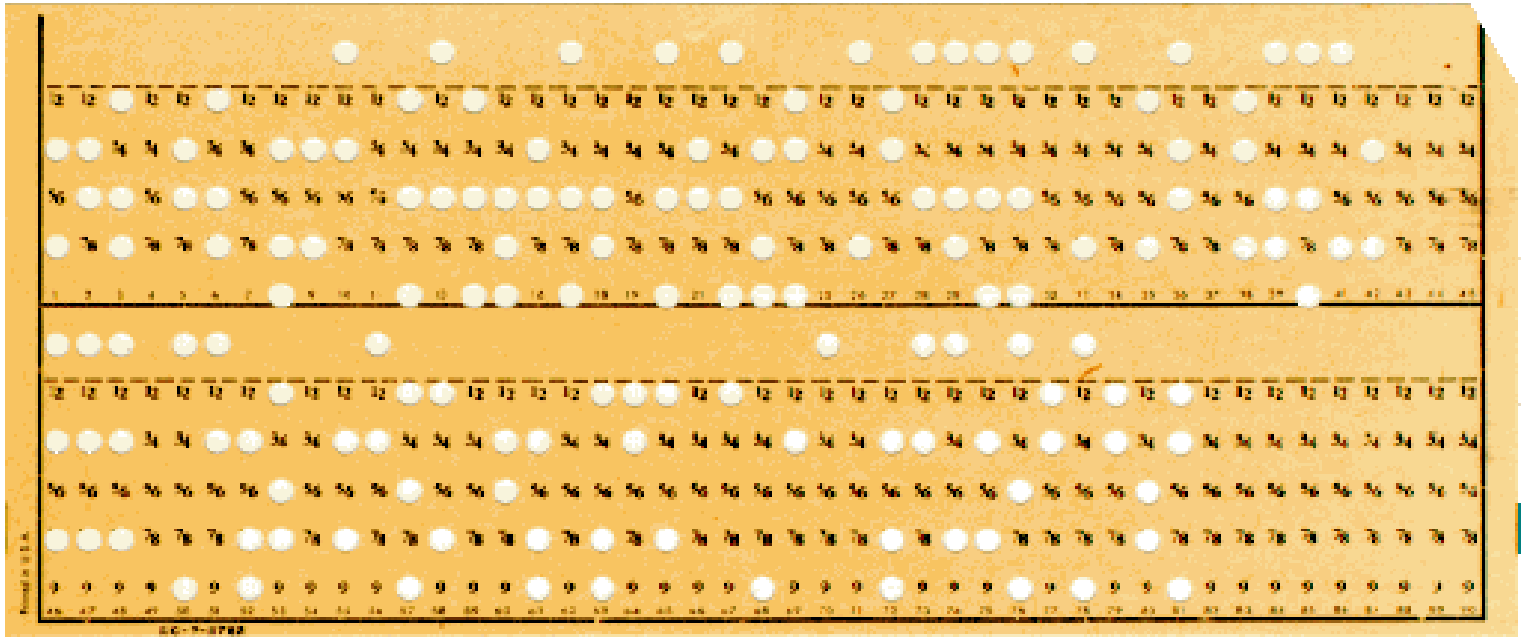
计算机操作是由用户(即程序员)采用人工操作方式直接使用计算机硬件系统，即由程序员将事先已穿孔(对应于程序和数据)的纸带(或卡片)装入纸带输入机(或卡片输入机)，再启动它们将程序和数据输入计算机，然后启动计算机运行。

1.2.1 无操作系统的计算机系统

第一台电子计算机-- ENIAC (1946)

1.2.1 无操作系统的计算机系统

穿孔卡片（Punched Card）



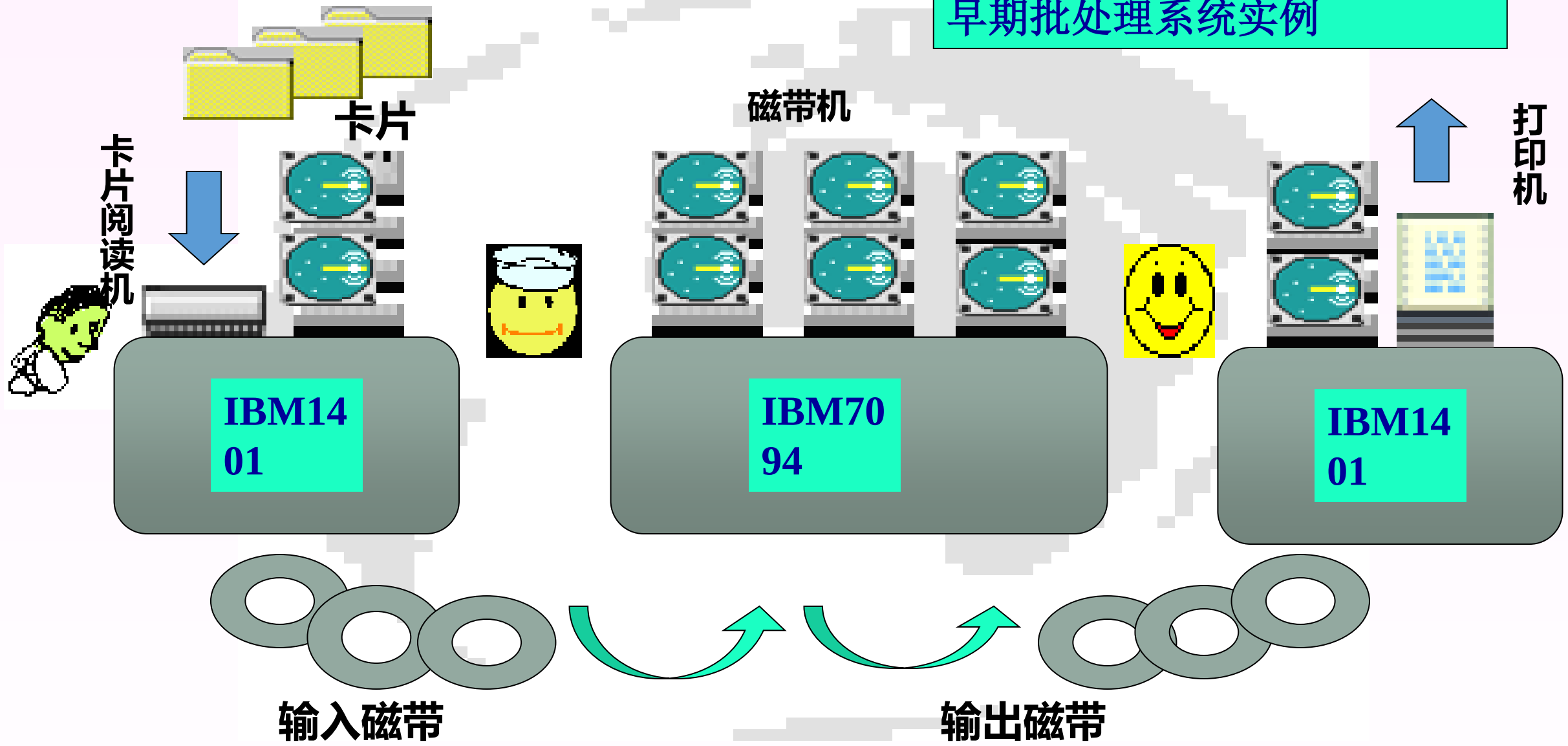
1.2.1 无操作系统的计算机系统

2．脱机输入/输出方式

为了解决人机矛盾及CPU和I/O设备之间速度不匹配的矛盾，20世纪50年代末出现了脱机输入/输出(Off-Line I/O)技术。

该技术是事先将装有用户程序和数据的纸带(或卡片)装入纸带输入机(或卡片机)，在一台外围机的控制下，把纸带(卡片)上的数据(程序)输入到磁带上。当CPU需要这些程序和数据时，再从磁带上将其高速地调入内存。

早期批处理系统实例



1.2.1 无操作系统的计算机系统

脱机I/O方式的主要优点：

- 减少了CPU的空闲时间。装带(卡)、卸带(卡)以及将数据从低速I/O设备送到高速磁带(或盘)上，都是在脱机情况下进行的，并不占用主机时间，从而有效地减少了CPU的空闲时间，缓和了人机矛盾。
- 提高了I/O速度。当CPU在运行中需要数据时，是直接从高速的磁带或磁盘上将数据调入内存的，不再是从低速I/O设备上输入，极大地提高了I/O速度，从而缓和了CPU和I/O设备速度不匹配的矛盾，进一步减少了CPU的空闲时间。

1.2.2 单道批处理系统

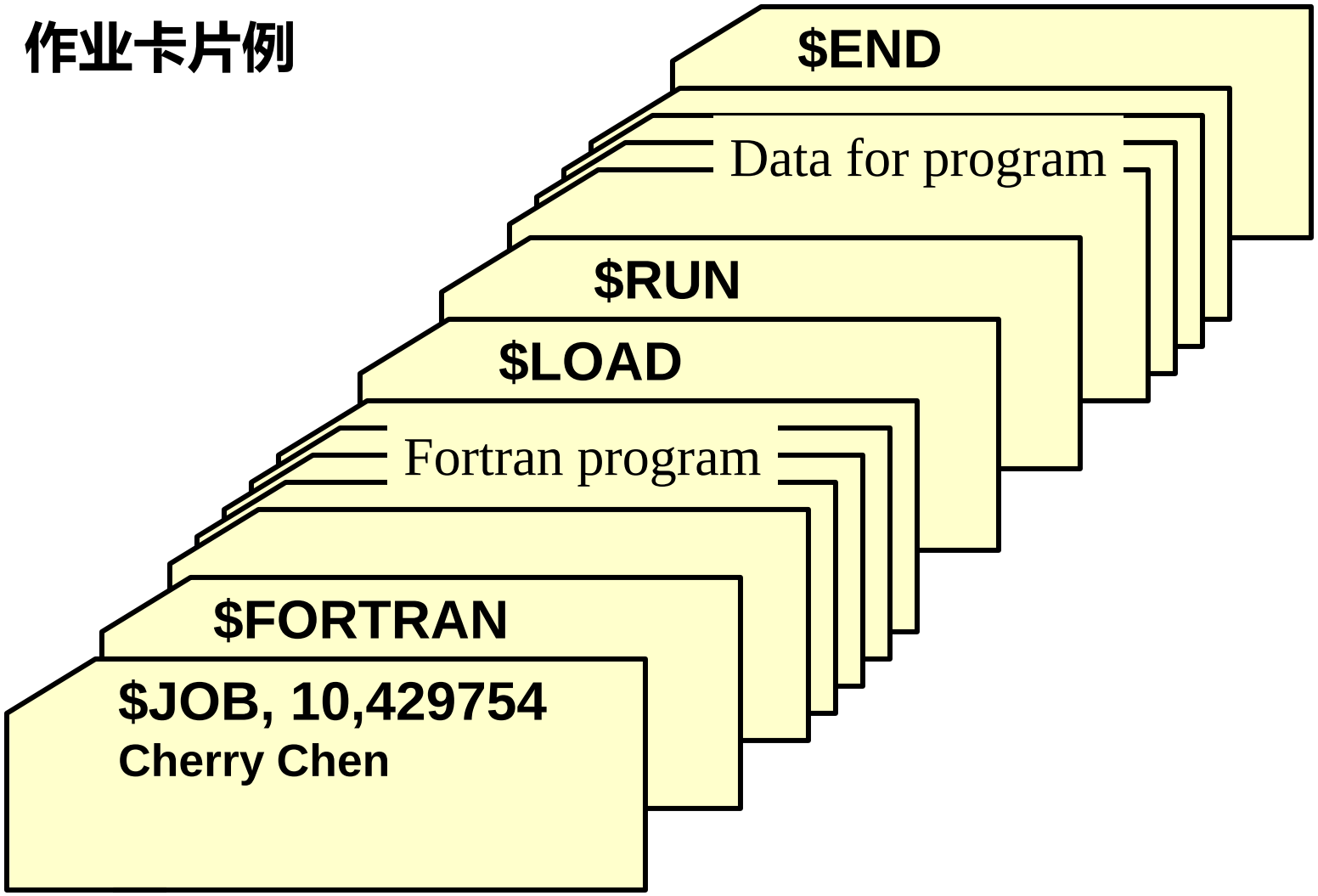
1．单道批处理系统的处理过程

上世纪50年代中期发明了晶体管，使计算机的体积大大减小，功耗显著降低，同时可靠性也得到大幅度提高，但计算机系统仍非常昂贵。为了能充分地利用它，应尽量让该系统连续运行，以减少空闲时间。

为此，通常是把一批作业以脱机方式输入到磁带上，并在系统中配上监督程序(Monitor)，在它的控制下使这批作业能一个接一个地连续处理。

1.2.2 单道批处理系统

作业卡片例



1.2.2 单道批处理系统

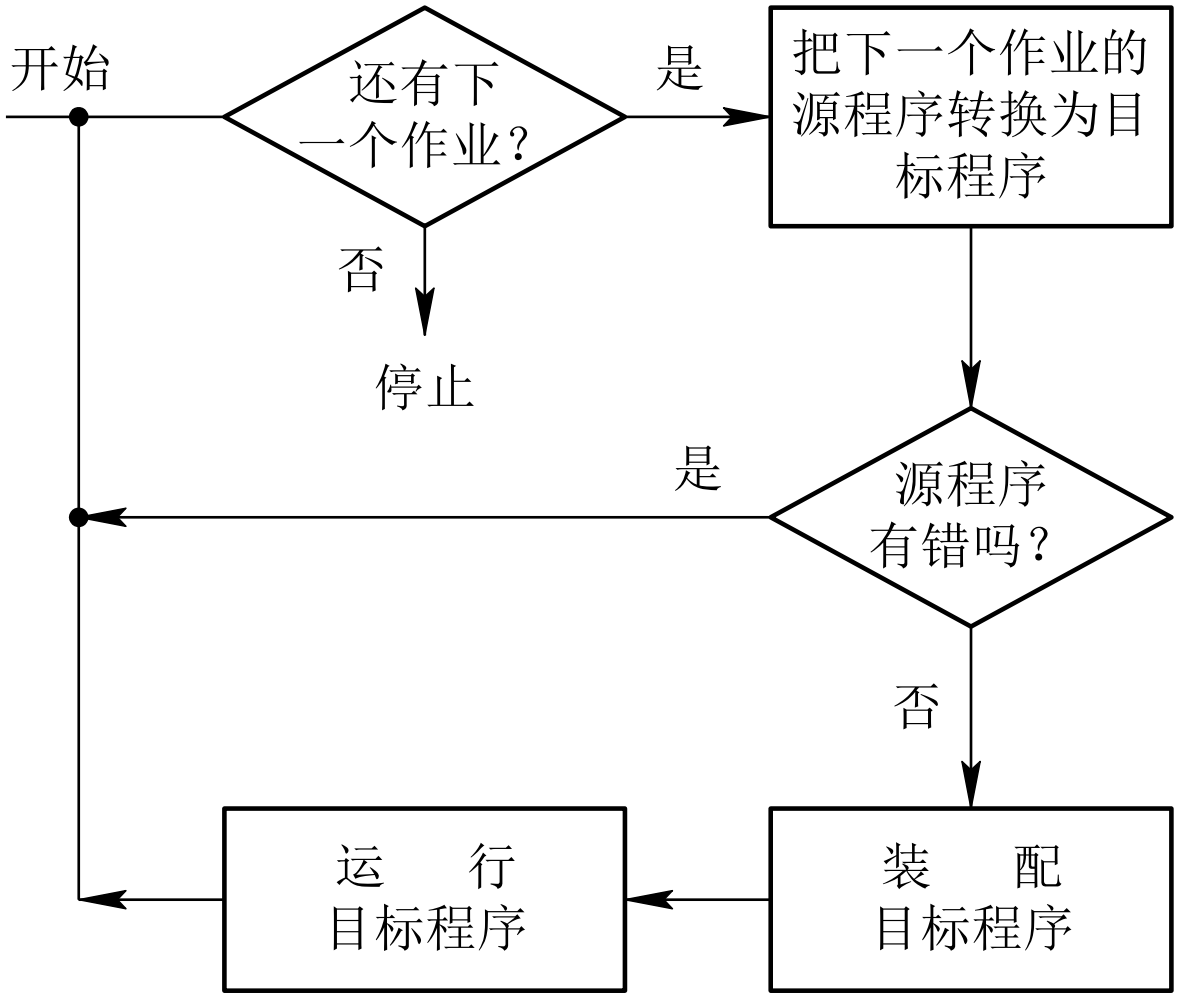


图1-4 单道批处理系统的处理流程

1.2.2 单道批处理系统

2．单道批处理系统的特征

单道批处理系统是最早出现的一种OS。该系统的主要特征如下：

- 自动性。在顺利情况下，在磁带上的一批作业能自动地逐个地依次运行，而无需人工干预。
- 顺序性。磁带上的各道作业是顺序地进入内存，各道作业的完成顺序与它们进入内存的顺序，在正常情况下应完全相同，亦即先调入内存的作业先完成。
- 单道性。内存中仅有一道程序运行，即监督程序每次从磁带上只调入一道程序进入内存运行，当该程序完成或发生异常情况时，才换入其后继程序进入内存运行。

1.2.3 多道批处理系统

1．多道程序设计的基本概念

- 为了进一步提高资源的利用率和系统吞吐量，在20世纪60年代中期又引入了多道程序设计技术，由此而形成了多道批处理系统(Multiprogrammed Batch Processing System)。
- 在该系统中，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。

1.2.3 多道批处理系统

2．多道程序设计的优点

1) 提高CPU的利用率。当内存中仅有一道程序时，每逢该程序在运行中发出I/O请求后，CPU空闲，必须在其I/O完成后CPU才继续运行；尤其因I/O设备的低速性，更使CPU的利用率显著降低。

在引入多道程序设计技术后，由于同时在内存中装有若干道程序，并使它们交替地运行，这样，当正在运行的程序因I/O而暂停执行时，系统可调度另一道程序运行，从而保持了CPU处于忙碌状态。

1.2.3 多道批处理系统

2) 可提高内存和I/O设备利用率。允许在内存中装入多道程序，并允许它们并发执行，则无疑会大大提高内存和I/O设备的利用率。

3) 增加系统吞吐量。在保持CPU、I/O设备不断忙碌的同时，也必然会大幅度地提高系统的吞吐量。

1.2.3 多道批处理系统

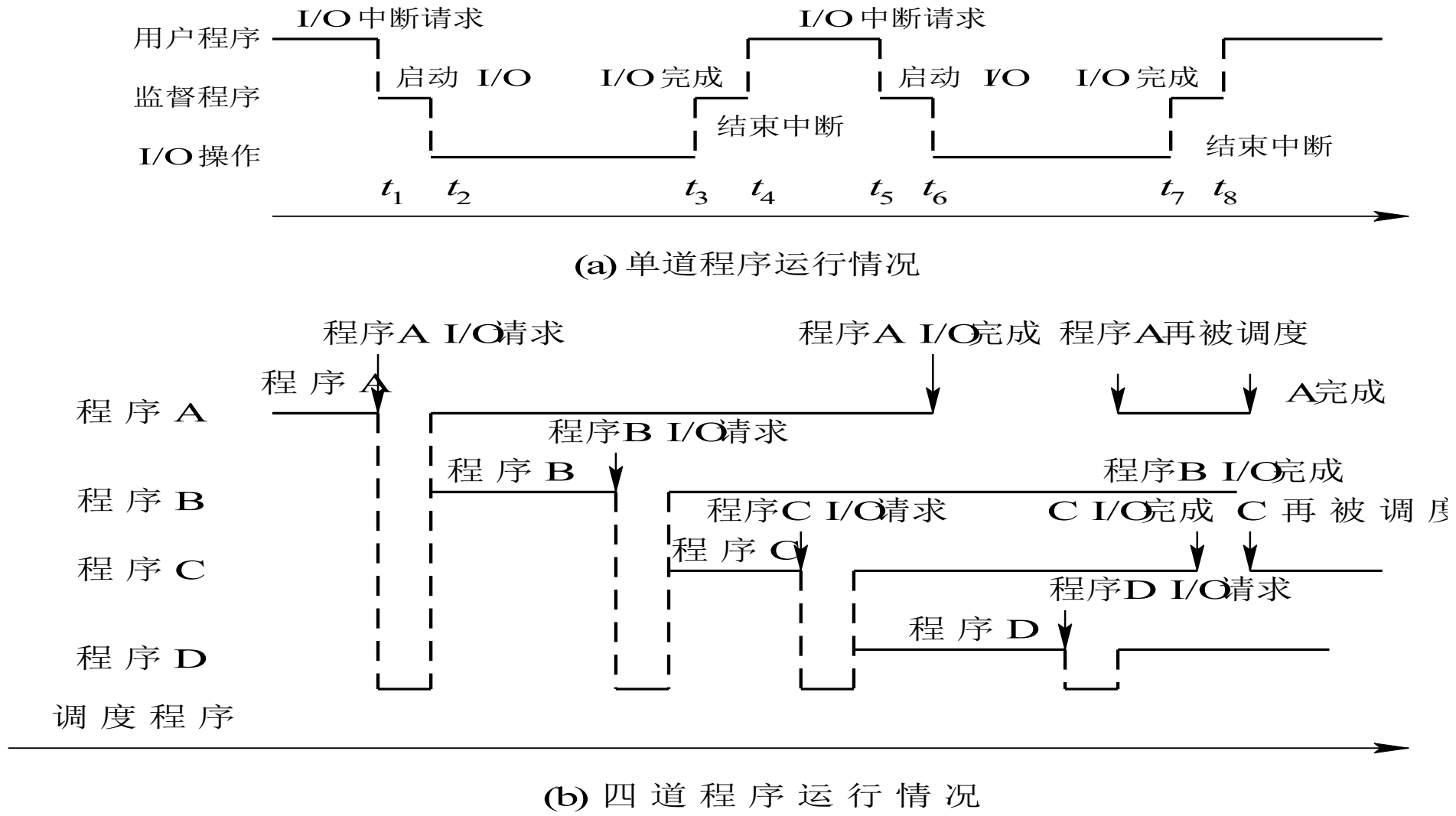


图1-5 单道和多道程序运行情况

1.2.3 多道批处理系统

3. 多道程序设计的缺点：

- 平均周转时间长。
- 依然无交互能力（同单道批处理系统）。

1.2.3 多道批处理系统

4．多道批处理系统需要解决的问题

多道批处理系统是一种更高效、但更复杂的系统。为使系统中的多道程序间能协调地运行，必须解决下述一系列问题。

- 处理机管理问题。
- 内存管理问题。
- I/O设备管理问题。
- 文件管理问题。
- 作业管理问题。

1.2.4 分时系统

1．分时系统的产生

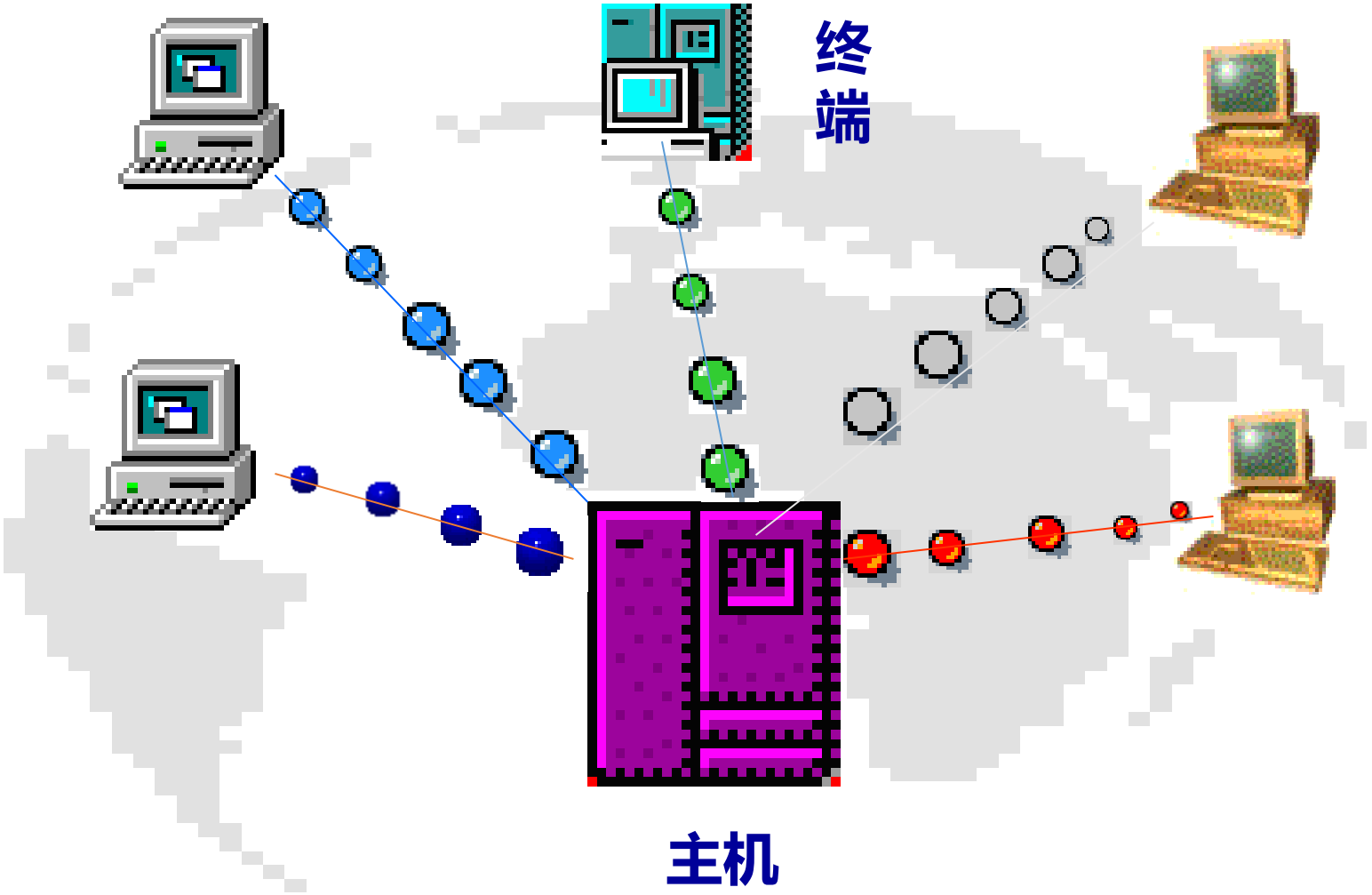
分时系统(Time Sharing System)与多道批处理系统之间有着截然不同的性能差别，它能很好地将一台计算机提供给多个用户同时使用，提高计算机的利用率。用户的需求具体表现在以下几个方面：

- 人-机交互。每当程序员写好一个新程序时，都需要上机进行调试。由于新编程序难免有些错误或不当之处需要修改，因而希望能像早期使用计算机时一样对它进行直接控制，并能以边运行边修改的方式，对程序中的错误进行修改，亦即，希望能进行人-机交互。

1.2.4 分时系统

- 共享主机。在20世纪60年代计算机非常昂贵，不可能像现在这样每人独占一台微机，而只能是由多个用户共享一台计算机，但用户在使用机器时应能够像自己独占计算机一样，不仅可以随时与计算机交互，而且应感觉不到其他用户也在使用该计算机。
- 便于用户上机。在多道批处理系统中，用户上机前必须把自己的作业邮寄或亲自送到机房。这对于用户尤其是远地用户来说是十分不便的。用户希望能通过自己的终端直接将作业传送到机器上进行处理，并能对自己的作业进行控制。

1.2.4 分时系统



分时系统工作示意图

1.2.4 分时系统

2．分时系统实现中的关键问题

- 及时接收。需在系统中配置一个多路卡。多路卡的作用是使主机能同时接收各用户从终端上输入的数据。此外，还须为每个终端配置一个缓冲区，用来暂存用户键入的命令(或数据)。
- 及时处理。人机交互的关键，是使用户键入命令后能及时地控制自己作业的运行，或修改自己的作业。为此，各个用户的作业都必须在内存中，且应能频繁地获得处理机而运行。

1.2.4 分时系统

3．分时系统的特征

分时系统与多道批处理系统相比，具有非常明显的不同特征：

- 多路性。允许在一台主机上同时联接多台联机终端，系统按分时原则为每个用户服务。宏观上，是多个用户同时工作，共享系统资源；而微观上，则是每个用户作业轮流运行一个时间片。
- 独立性。每个用户各占一个终端，彼此独立操作，互不干扰。因此，用户所感觉到的，就像是他一人独占主机。
- 及时性。用户的请求能在很短的时间内获得响应。此时间间隔是以人们所能接受的等待时间来确定的，通常仅为1～3秒钟。
- 交互性。用户可通过终端与系统进行广泛的人机对话。其广泛性表现在：用户可以请求系统提供多方面的服务，如文件编辑、数据处理和资源共享等。

1.2.4 分时系统

3．分时系统的特征

分时系统与多道批处理系统相比，具有非常明显的不同特征：

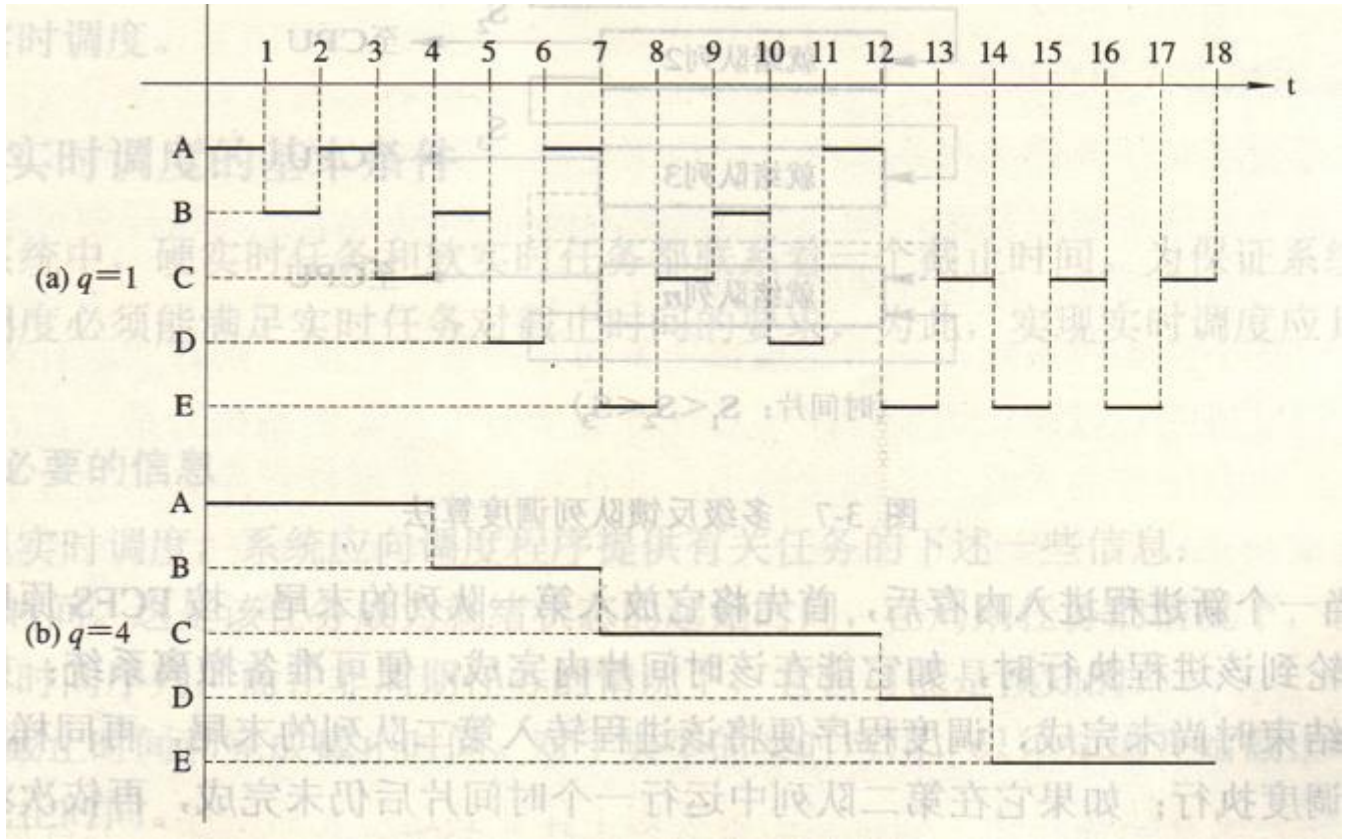
- 多路性。允许在一台主机上同时联接多台联机终端，系统按分时原则为每个用户服务。宏观上，是多个用户同时工作，共享系统资源；而微观上，则是每个用户作业轮流运行一个时间片。
- 独立性。每个用户各占一个终端，彼此独立操作，互不干扰。因此，用户所感觉到的，就像是他一人独占主机。
- 及时性。用户的请求能在很短的时间内获得响应。此时间间隔是以人们所能接受的等待时间来确定的，通常仅为1～3秒钟。
- 交互性。用户可通过终端与系统进行广泛的人机对话。其广泛性表现在：用户可以请求系统提供多方面的服务，如文件编辑、数据处理和资源共享等。

1.2.4 分时系统

分时系统的几个概念：

- 分时(Time Sharing)： CPU时间分段处理
- 时间片(time slice)：CPU的时间段
- 响应时间(response time)：从终端发出请求到系统给予回答所经历的时间

1.2.4 分时系统



q=1和q=4时的进程运行情况

1.2.5 实时系统

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

- 实时控制。生产过程的控制以及武器的控制，如火炮的自动控制系统、飞机的自动驾驶系统，以及导弹的制导系统等。
- 实时信息处理。计算机接收从远程终端上发来的服务请求，根据用户提出的请求对信息进行检索和处理，并在很短的时间内为用户做出正确的响应。典型的实时信息处理系统有早期的飞机或火车的订票系统、情报检索系统等。

1.2.5 实时系统

2．实时任务

1) 按任务执行时是否呈现周期性来划分

- 周期性实时任务。外部设备周期性地发出激励信号给计算机，要求它按指定周期循环执行，以便周期性地控制某外部设备。
- 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)。它又可分为开始截止时间(某任务在某时间以前必须开始执行)和完成截止时间(某任务在某时间以前必须完成)两部分。

1.2.5 实时系统

2) 根据对截止时间的要求来划分

- 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。
- 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。

1.2.5 实时系统

3．实时系统与分时系统特征的比较

- 多路性。实时信息处理系统也按分时原则为多个终端用户服务。实时控制系统的多路性则主要表现在系统周期性地对多路现场信息进行采集，以及对多个对象或多个执行机构进行控制。而分时系统中的多路性则与用户情况有关，时多时少。
- 独立性。实时信息处理系统中的每个终端用户在向实时系统提出服务请求时，是彼此独立地操作，互不干扰；而实时控制系统中，对信息的采集和对对象的控制也都是彼此互不干扰。
- 交互性。实时信息处理系统虽然也具有交互性，但这里人与系统的交互仅限于访问系统中某些特定的专用服务程序。它不像分时系统那样能向终端用户提供数据处理和资源共享等服务。

1.2.5 实时系统

- 及时性。实时信息处理系统对实时性的要求与分时系统类似，都是以人所能接受的等待时间来确定的；而实时控制系统的及时性，则是以控制对象所要求的开始截止时间或完成截止时间来确定的，一般为秒级到毫秒级，甚至有的要低于100微秒。
- 可靠性。分时系统虽然也要求系统可靠，但相比之下，实时系统则要求系统具有高度的可靠性。因为任何差错都可能带来巨大的经济损失，甚至是无法预料的灾难性后果，所以在实时系统中，往往都采取了多级容错措施来保障系统的安全性及数据的安全性。

1.2.5 批处理系统、分时系统、实时系统的比较

批处理系统、分时系统、实时系统是最常见的操作系统。

	交互性	及时性	可靠性	效率
批处理系统	弱	弱	中	最高
分时系统	强	强	中	低
实时系统	特定交互强	最强	最强	低

1.2.6 微机操作系统的发展

1．单用户单任务操作系统

单用户单任务操作系统的含义是，只允许一个用户上机，且只允许用户程序作为一个任务运行。这是最简单的微机操作系统，主要配置在8位和16位微机上。最有代表性的单用户单任务微机操作系统是CP/M和MS-DOS。

1.2.6 微机操作系统的发展

1）CP/M

1974年第一代通用8位微处理机芯片Intel 8080出现后的第二年，Digital Research公司就开发出带有软盘系统的8位微机操作系统。1977年Digital Research公司对CP/M进行了重写，使其可配置在以Intel 8080、8085、Z80等8位芯片为基础的多种微机上。1979年又推出带有硬盘管理功能的CP/M 2.2版本。由于CP/M具有较好的体系结构，可适应性强，且具有可移植性以及易学易用等优点，使之在8位微机中占据了统治地位。

2）MS-DOS

1981年IBM公司首次推出了IBM-PC个人计算机(16位微机)，在微机中采用了微软公司开发的MS-DOS(Disk Operating System)操作系统，该操作系统在CP/M的基础上进行了较大的扩充，使其在功能上有很大的增强。从20世纪80年代到90年代初，由于MS-DOS性能优越而受到当时用户的广泛欢迎，成为事实上的16位单用户单任务操作系统标准。

1.2.6 微机操作系统的发展

2．单用户多任务操作系统

单用户多任务操作系统的含义是，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，从而有效地改善了系统的性能。目前在32位微机上配置的操作系统基本上都是单用户多任务操作系统，其中最有代表性的是由微软公司推出的Windows。

1.2.6 微机操作系统的发展

3．多用户多任务操作系统

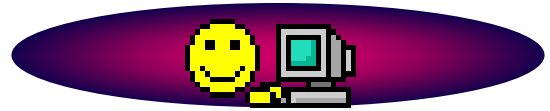
多用户多任务操作系统的含义是，允许多个用户通过各自的终端使用同一台机器，共享主机系统中的各种资源，而每个用户程序又可进一步分为几个任务，使它们能并发执行，从而可进一步提高资源利用率和系统吞吐量。在大、中和小型机中所配置的大多是多用户多任务操作系统，其中最有代表性的是UNIX OS。

1.2.6 微机操作系统的发展

UNIX OS是美国电报电话公司的Bell实验室在1969～1970年期间开发的，1979年推出来的UNIX V.7已被广泛应用于多种中、小型机上。随着微机性能的提高，人们又将UNIX移植到微机上。在1980年前后，将UNIX第7版本移植到Motorola公司的MC 680xx微机上，后来又将UNIX V7.0版本进行简化后移植到Intel 8080上，把它称为Xenix。现在最有影响的两个能运行在微机上的UNIX操作系统的变型是Solaris OS和Linux OS。

网络操作系统

•特点



互连（独立OS)
自治性
网络协议
可靠的数据传输
网络服务

• 功能

网络通信

资源共享管理
网络服务
网络管理
互操作能力

适应作业：异地通信、资源共享

分布式操作系统

分布式系统是由若干个计算机经互连网络连接而成的，这些计算机既可以独立工作，又能协同工作，它们对于用户来说就像一个系统。

• 分布式OS特点

- 多机
- 统一的操作系统界面
- 健壮性
- 透明性

适应作业：任务协同处理，协同计算、通信、资源共享

●嵌入式操作系统

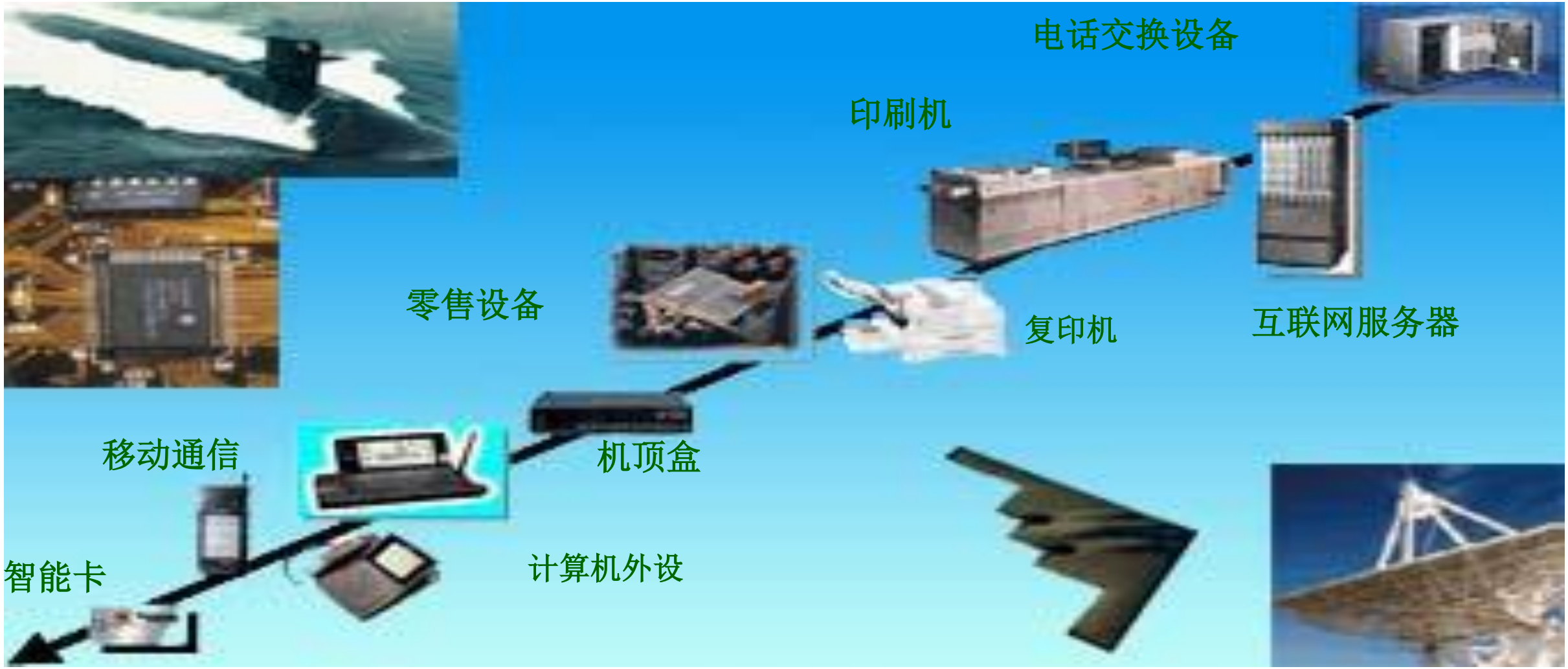
●嵌入式系统

嵌入式计算机，顾名思义即将计算机嵌入到其他设备上，这些设备无处不在，大到汽车发动机、机器人，小到电视机、微波炉、移动电话。运行在其上的操作系统比较简单，只实现所要求的控制功能。

●特点

1.专用。完成某一项或有限项功能
2. 实时。在性能和实时性方面有严格的限制， 能源、成本和可靠性通常是影响设计的重要因素
3.资源受限： 占有资源少、易于连接

嵌入式操作系统的应用



1.3 操作系统的特征

并发

两个或两个以上的事件在同一时间间隔内发生

共享

系统中的软、硬件资源供多个程序共同享用

- 互斥共享——某时间段内，只允许一个进程访问，该进程访问完了，别人才可访问。如：打印机
- 同时访问——某时间段内，允许多个进程同时访问。如：磁盘

虚拟

通过某种技术手段把一个物理上的实体，变成多个逻辑上的对应物

异步

也称随机性，是指系统中各种事件的发生顺序是不确定的。
进程是以人们不可预知的速度向前推进，走走停停。

1.3.1 并发性

1．并行与并发

并行性和并发性是既相似又有区别的两个概念，并行性是指两个或多个事件在同一时刻发生；而并发性是指两个或多个事件在同一时间间隔内发生。

在多道程序环境下，并发性是指在一段时间内，宏观上有多个程序在同时运行，但在单处理机系统中，每一时刻却仅能有一道程序执行，故微观上这些程序只能是分时地交替执行。倘若在计算机系统中有多个处理机，则这些可以并发执行的程序便可被分配到多个处理机上，实现并行执行，多个程序可同时执行。

1.3.1 并发性

2．引入进程

程序是静态实体(Passive Entity)，在多道程序系统中，它们是不能独立运行的，更不能和其它程序并发执行。在操作系统中引入进程的目的，就是为了使多个程序能并发执行，从而有效地提高了系统资源的利用率和系统吞吐量，并改善了系统的性能。

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，它是由一组机器指令、数据和堆栈等组成的，是一个能独立运行的活动实体。多个进程之间可以并发执行和交换信息。一个进程在运行时需要一定的资源，如CPU、存储空间及I/O设备等。

1.3.1 并发性

3．引入线程

通常在一个进程中可以包含若干个线程，它们可以利用进程所拥有的资源。在引入线程的OS中，通常都是把进程作为分配资源的基本单位，而把线程作为独立运行和独立调度的基本单位。由于线程比进程更小，基本上不拥有系统资源，故对它的调度所付出的开销就会小得多，能更高效地提高系统内多个程序间并发执行的程度。

1.3.2 共享性

1．互斥共享方式

系统中的某些资源，如打印机、磁带机，虽然它们可以提供给多个进程(线程)使用，但为使所打印或记录的结果不致造成混淆，应规定在一段时间内只允许一个进程(线程)访问该资源。为此，系统中应建立一种机制，以保证对这类资源的互斥访问。

2．同时访问方式

系统中还有另一类资源，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”，在单处理机环境下往往是宏观上的，而在微观上，这些进程可能是交替地对该资源进行访问。典型的可供多个进程“同时”访问的资源是磁盘设备。

1.3.2 共享性

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。

- 一方面，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；
- 另一方面，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。

1.3.3 虚拟技术

1．时分复用技术

时分复用，亦即分时使用方式，在计算机领域中，广泛利用该技术来实现虚拟处理机、虚拟设备等，以提高资源的利用率。

1) 虚拟处理机技术

在虚拟处理机技术中，利用多道程序设计技术，为每道程序建立一个进程，让多道程序并发地执行，以此来分时使用一台处理机。此时，虽然系统中只有一台处理机，但它却能同时为多个用户服务，使每个终端用户都认为是有一个处理机在专门为他服务。我们把用户所感觉到的处理机称为虚拟处理器。

2) 虚拟设备技术

我们还可以通过虚拟设备技术，将一台物理I/O设备虚拟为多台逻辑上的I/O设备，并允许每个用户占用一台逻辑上的I/O设备，这样便可使原来仅允许在一段时间内由一个用户访问的设备(即临界资源)，变为在一段时间内允许多个用户同时访问的共享设备。

1.3.3 虚拟技术

2．空分复用技术

它是将一个频率范围非常宽的信道，划分成多个频率范围较窄的信道，其中的任何一个频带都只供一对用户通话。在计算机中也使用了空分复用技术来提高存储空间的利用率。

1) 虚拟磁盘技术

通常在一台机器上只配置一台硬盘。我们可以通过虚拟磁盘技术将一台硬盘虚拟为多台虚拟磁盘，这样使用起来既方便又安全。虚拟磁盘技术也是采用了空分复用方式，即它将硬盘划分为若干个卷，例如1、2、3、4四个卷，再通过安装程序将它们分别安装在C、D、E、F四个逻辑驱动器上。

2) 虚拟存储器技术

空分复用可以利用存储器的空闲空间来存放其它的程序，以提高内存的利用率。

但是，单纯的空分复用存储器只能提高内存的利用率，并不能实现在逻辑上扩大存储器容量的功能，必须引入虚拟存储技术才能达到此目地。而虚拟存储技术在本质上就是使内存分时复用。它可以使一道程序通过时分复用方式，在远小于它的内存空间中运行。 例如，一个100 MB的应用程序可以运行在20 MB的内存空间。

1.3.4 异步性

在多道程序环境下允许多个进程并发执行，但只有进程在获得所需的资源后才能执行。在单处理机环境下，由于系统中只有一台处理机，因而每次只允许一个进程执行，其余进程只能等待。由于资源等因素的限制，使进程的执行通常都不是“一气呵成”，而是以“停停走走”的方式运行。

1.4 操作系统的五大功能

- 处理机管理
- 存储管理
- 设备管理
- 文件管理
- 用户接口

1.4.1 处理机管理

任务

- 对处理机的分配和运行实施有效管理。
- 在多道程序环境下，处理机的分配和运行以进程为单位，因此，对处理机的管理即对进程的管理。

功能

- 进程控制
- 进程同步
- 进程通信
- 进程调度

1.4.2 存储器管理

任务

- 方便用户使用内存
- 提高内存的利用率
- 从逻辑上扩充内存

功能

- 内存分配
- 内存保护
- 地址映射
- 内存扩充

1.4.3 设备管理

任务

- 完成用户程序请求的I/O操作，为用户程序分配I/O设备
- 提高外部设备的利用率
- 尽可能地提高输入/输出的速度
- 方便用户使用外部设备

功能

- 缓冲管理
- 设备分配
- 设备处理
- I/O控制
- 磁盘调度

1.4.4 文件管理

任务

- 大量的信息以文件的形式放在外存，对信息的管理也就是对文件的管理

功能

- 文件存储空间的管理
- 文件结构
- 目录管理
- 文件的读、写管理和保护

1.4.5 操作系统与用户接口

命令接口

- 联机命令接口
- 脱机命令接口
- 图形用户界面（命令接口的升级版）

程序接口

- 也称**系统调用**

1.5 操作系统结构设计

- OS是一个庞大的系统软件，如何设计与实现是一项艰难而复杂的系统工程，往往用人年来衡量。
- 设计目标：正确性、性能、功能、方便性
- 设计方法：
  - 无结构操作系统
  - 模块化结构OS
  - 分层式结构OS
  - 面向对象设计
  - 微内核OS结构

1.5.1 传统的操作系统结构

1.无结构操作系统

在早期开发操作系统时，设计者只是把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。此时的OS是为数众多的一组过程的集合，每个过程可以任意地相互调用其它过程，致使操作系统内部既复杂又混乱。因此，这种OS是无结构的。

ESSAYS ON SOFTWARE ENGINEERING

1.5.1 传统的操作系统结构

2．模块化结构OS

1) 模块化程序设计技术的基本概念

模块化程序设计技术是20世纪60年代出现的一种结构化程序设计技术。该技术是基于“分解”和“模块化”原则来控制大型软件的复杂度。为使OS具有较清晰的结构，OS不再是由众多的过程直接构成，而是将OS按其功能精心地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，如进程管理模块、存储器管理模块、I/O设备管理模块等，并仔细地规定好各模块间的接口，使各模块之间能通过该接口实现交互。

1.5.1 传统的操作系统结构

然后，再进一步将各模块细分为若干个具有一定功能的子模块，如把进程管理模块又分为进程控制、进程同步等子模块，同样也要规定好各子模块之间的接口。若子模块较大时，可再进一步将它细分。我们把这种设计方法称为模块-接口法，由此构成的操作系统就是具有模块化结构的操作系统。

1.5.1 传统的操作系统结构

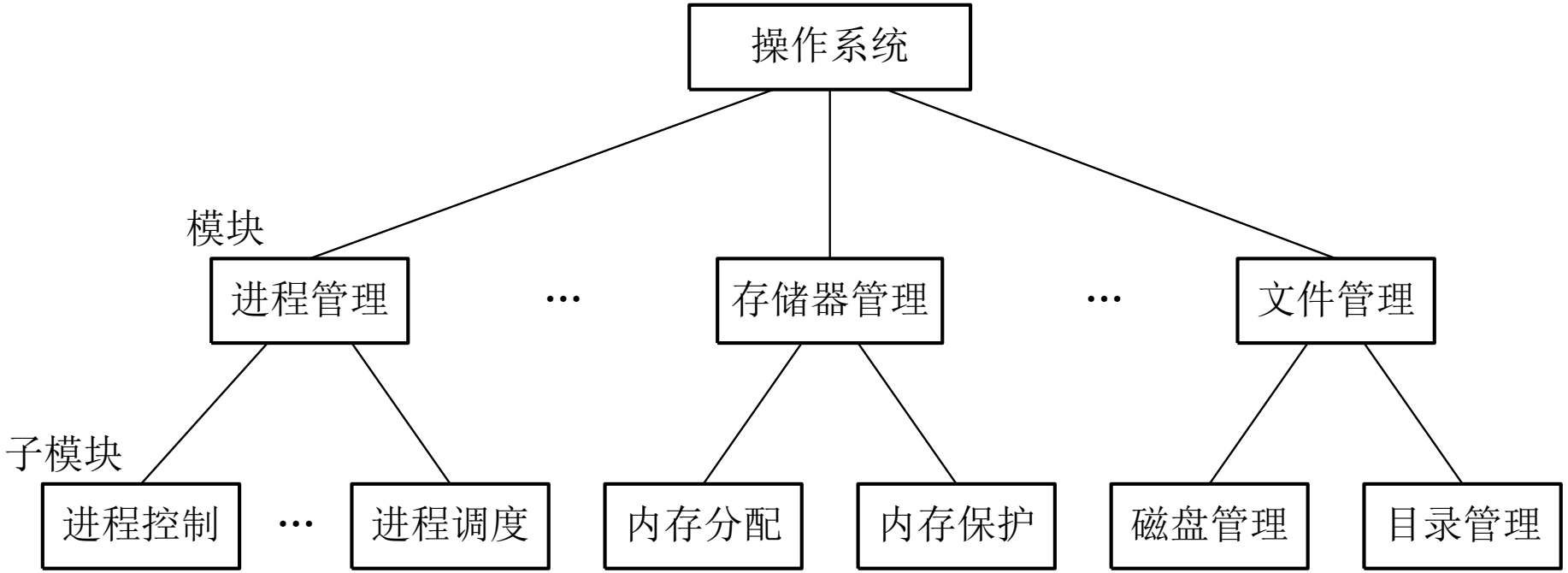


图1-6 模块化结构的操作系统

1.5.1 传统的操作系统结构

2) 模块的独立性

在模块-接口法设计方法中，关键问题是模块的划分和规定好模块之间的接口。如果将模块划分得太小，虽然可以降低模块本身的复杂性，但会引起模块之间的联系过多，而会造成系统比较混乱；如果将模块划分得过大，又会增加模块内部的复杂性，使内部的联系增加。因此，在划分模块时，应在两者间进行权衡。

另外，在划分模块时，必须充分注意模块的独立性问题。因为模块的独立性越高，各模块间的交互就越少，系统的结构也就越清晰。衡量模块的独立性有以下两个标准：

- 内聚性，指模块内部各部分间联系的紧密程度。内聚性越高，模块的独立性越强。
- 耦合度，指模块间相互联系和相互影响的程度。显然，耦合度越低，模块的独立性越好。

1.5.1 传统的操作系统结构

3) 模块接口法的优缺点

利用模块-接口法开发的OS，较之无结构OS具有以下明显的优点：

- 提高OS设计的正确性、可理解性和可维护性；
- 增强OS的适应性；
- 加速OS的开发过程。

模块化结构设计仍存在下述问题：

- 在OS设计时，对各模块间的接口规定很难满足在模块完成后对接口的实际需求。
- 在OS设计阶段，设计者必须做出一系列的决定(决策)，每一个决定必须建立在上一个决定的基础上。但在模块化结构设计中，各模块的设计齐头并进，无法寻找到一个可靠的决定顺序，造成各种决定的“无序性”，这将使程序设计人员很难做到“设计中的每一步决定都是建立在可靠的基础上”，因此，“模块-接口法”又被称为“无序模块法”。

1.5.1 传统的操作系统结构

3．分层式结构OS

1) 分层式结构的基本概念

为了将模块-接口法中“决定顺序”的无序性变为有序性，引入了有序分层法。分层法的设计任务是，在目标系统A_n和裸机系统(又称宿主系统)A_0之间，铺设若干个层次的软件A_1、A_2、A_3、…、A_{n-1}，使A_n通过A_{n-1}、A_{n-2}、…、A_2、A_1层，最终能在A_0上运行。在操作系统中，常采用自底向上法来铺设这些中间层。

自底向上的分层设计的基本原则是：每一步设计都是建立在可靠的基础上。为此规定，每一层仅能使用其底层所提供的功能和服务，这样可使系统的调试和验证都变得更容易。

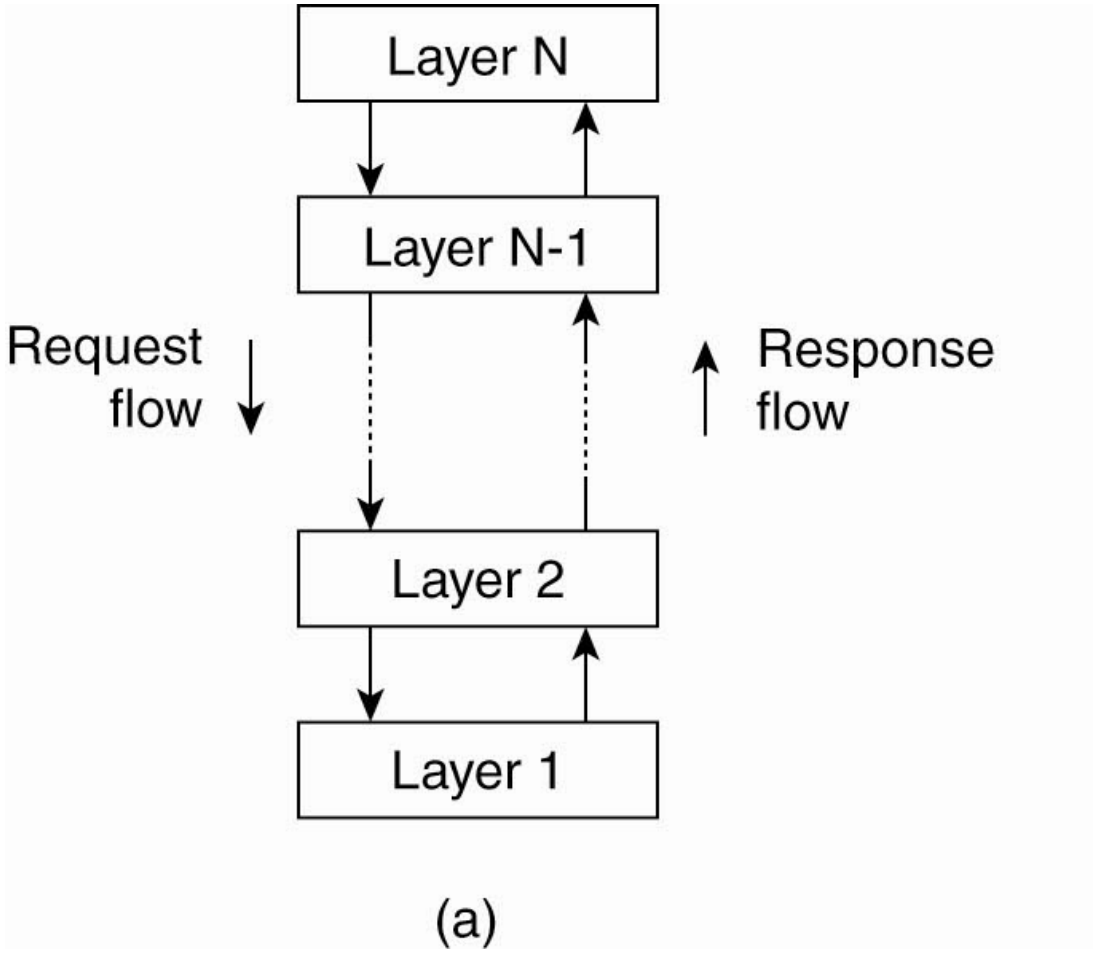
2) 分层结构的优缺点

分层结构的主要优点有：

- 易保证系统的正确性。自下而上的设计方式，使所有设计中的决定都是有序的，或者说是建立在较为可靠的基础上的，这样比较容易保证整个系统的正确性。
- 易扩充和易维护性。在系统中增加、修改或替换一个层次中的模块或整个层次，只要不改变相应层次间的接口，就不会影响其它层次，这必将使系统维护和扩充变得更加容易。

分层结构的主要缺点是系统效率会降低。

1.5.1 传统的操作系统结构



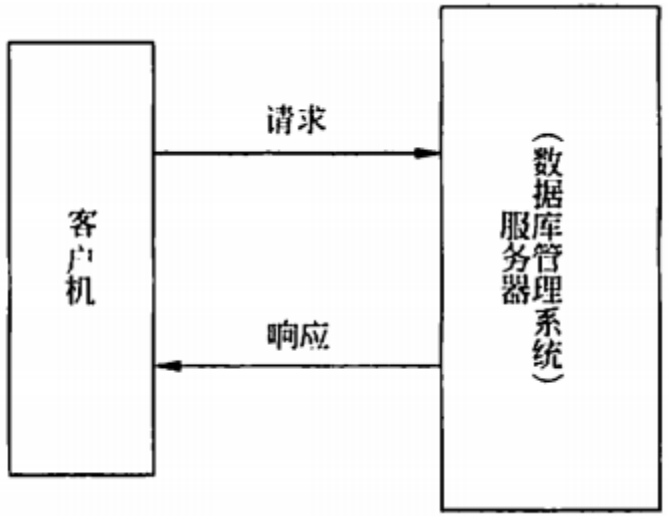
分层式结构

1.5.2 客户/服务器模式

客户/服务器(Client/Server)模式可简称为C/S模式。

1．客户/服务器模式的组成

- 客户机：通常在一个LAN网络上连接有多台网络工作站(简称客户机)，每台客户机都是一个自主计算机，具有一定的处理能力。
- 服务器：通常是一台规模较大的机器，在其上驻留有网络文件系统或数据库系统等，它应能为网上所有的用户提供一种或多种服务。
- 网络系统：用于连接所有客户机和服务器，实现它们之间通信和网络资源共享的系统。



1.5.2 客户/服务器模式

2．客户/服务器之间的交互

在采用客户/服务器的系统中，在客户机和服务器之间的交互过程可分成以下四步：

- 客户发送请求消息。当客户机上的用户要请求服务器进行应用处理时，应输入相应的命令和有关参数。由客户机上的发送进程先把这些信息装配成请求消息，然后把它发往服务器；客户机上的接收进程则等待接收从服务器发回来的响应消息。
- 服务器接收消息。服务器中的接收进程平时处于等待状态，一旦有客户机发来请求，接收进程便被激活，并根据请求信息的内容，将之提供给服务器上的相应软件进行处理。
- 服务器回送消息。服务器的软件根据请求进行处理，在完成指定的处理后，把处理结果装配成一个响应消息，由服务器中的发送进程将之发往客户机。
- 客户机接收消息。客户机中的接收进程把收到的响应消息转交给客户机软件，再由后者做出适当处理后提交给发送该请求的客户。

1.5.2 客户/服务器模式

3．客户/服务器模式的优点

C/S模式之所以能成为当前分布式系统和网络环境下软件的主要工作模式，是由于该模式具有传统集中模式所无法比拟的一系列优点。

- 数据的分布处理和存储。由于客户机具有相当强的处理和存储能力，可进行本地处理和数据的分布存储，从而摆脱了由于把一切数据都存放在主机中而造成的既不可靠又容易产生瓶颈现象的困局。
- 便于集中管理。尽管C/S模式具有分布处理功能，但公司(单位)中的有关全局的重要信息、机密资料、重要设备以及网络管理等，仍可采取集中管理方式。这样可较好地保障系统的“可靠”与“安全”。
- 灵活性和可扩充性。C/S模式非常灵活，极易扩充。理论上，客户机和服务器的数量不受限制。其灵活性还表现在可以配置多种类型的客户机和服务器。
- 易于改编应用软件。在客户/服务器模式中，对于客户机程序的修改和增删，比传统集中模式要容易得多，必要时也允许由客户进行修改。

1.5.3 面向对象的程序设计

1．面向对象技术的基本概念

面向对象技术是20世纪80年代初提出并很快流行起来的。该技术是基于“抽象”和“隐蔽”原则来控制大型软件的复杂度的。

2．面向对象技术的优点

在设计操作系统时，将计算机中的实体作为对象来处理，可带来如下好处：

(1) 通过“重用”提高产品质量和生产率。

在面向对象技术中可通过“重用”以前项目中经过精心测试的对象，或由其他人编写、测试和维护的对象类，来构建新的系统，这不仅可大大降低开发成本，而且能获得更好的系统质量。

1.5.3 面向对象的程序设计

(2) 使系统具有更好的易修改性和易扩展性。

通过封装，可隐蔽对象中的变量和方法，因而当改变对象中的变量和方法时，不会影响到其它部分，从而可方便地修改老的对象类。另外，继承是面向对象技术的重要特性，在创建一个新对象类时，通过利用继承特性，可显著地减少开发的时空开销，使系统具有更好的易扩展性和灵活性。

(3) 更易于保证系统的“正确性”和“可靠性”。

对象是构成操作系统的基本单元，由于可以独立地对它进行测试，易于保证每个对象的正确性和可靠性，因此也就比较容易保证整个系统的正确性和可靠性。此外，封装对对象类中的信息进行了隐蔽，这样又可有效地防止未经授权者的访问和用户不正确的使用，有助于构建更为安全的系统。

1.5.4 微内核OS结构

1．微内核操作系统的基本概念

为了提高操作系统的“正确性”、“灵活性”、“易维护性”和”可扩充性”，在进行现代操作系统结构设计时，即使在单处理机环境下，大多也采用基于客户/服务器模式的微内核结构，将操作系统划分为两大部分：微内核和多个服务器。关于什么是微内核操作系统结构，现在尚无一致公认的定义，但可以从下面四个方面，对微内核结构的操作系统进行描述。

1) 足够小的内核

在微内核操作系统中，内核是指精心设计的、能实现现代OS最基本的核心功能的部分。微内核并非是一个完整的OS，而只是操作系统中最基本的部分，它通常用于：① 实现与硬件紧密相关的处理；② 实现一些较基本的功能；③ 负责客户和服务器之间的通信。它们只是为构建通用OS提供一个重要基础，这样就可以确保把操作系统内核做得很小。

1.5.4 微内核OS结构

2) 基于客户/服务器模式

由于客户/服务器模式具有非常多的优点，故在单机微内核操作系统中几乎无一例外地都采用客户/服务器模式，将操作系统中最基本的部分放入内核中，而把操作系统的绝大部分功能都放在微内核外面的一组服务器(进程)中实现。例如，用于提供对进程(线程)进行管理的进程(线程)服务器，提供虚拟存储器管理功能的虚拟存储器服务器，提供I/O设备管理的I/O设备管理服务器等，它们都是被作为进程来实现的，运行在用户态，客户与服务器之间是借助微内核提供的消息传递机制来实现信息交互的。图1-10示出了在单机环境下的客户/服务器模式。

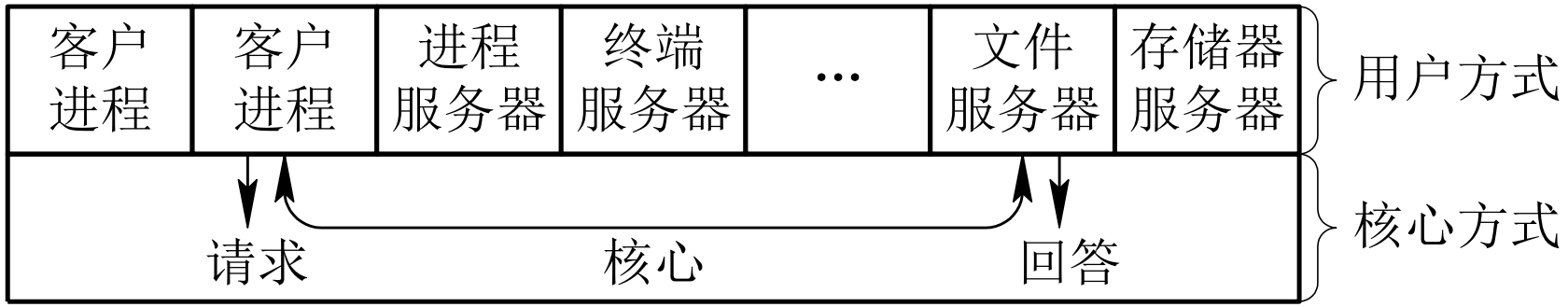


图1-10 在单机环境下的客户/服务器模式

1.5.4 微内核OS结构

3) 应用“机制与策略分离”原理

在现代操作系统的结构设计中，经常利用“机制与策略分离”的原理来构造OS结构。所谓机制，是指实现某一功能的具体执行机构。而策略，则是在机制的基础上，借助于某些参数和算法来实现该功能的优化，或达到不同的功能目标。通常，机制处于一个系统的基层，而策略则处于系统的高层。在传统的OS中，将机制放在OS的内核的较低层，把策略放在内核的较高层次中。而在微内核操作系统中，通常将机制放在OS的微内核中。正因为如此，才有可能将内核做得很小。

4) 采用面向对象技术

操作系统是一个极其复杂的大型软件系统，我们不仅可以通过结构设计来分解操作系统的复杂度，还可以基于面向对象技术中的“抽象”和“隐蔽”原则控制系统的复杂性，再进一步利用“对象”、“封装”和“继承”等概念来确保操作系统的“正确性”、“可靠性”、“易修改性”、“易扩展性”等，并提高操作系统的设计速度。

1.5.4 微内核OS结构

2．微内核的基本功能

1) 进程(线程)管理

为实现进程(线程)调度功能，须在进程管理中设置一个或多个进程(线程)优先级队列；能将指定优先级进程(线程)从所在队列中取出，并将其投入执行。如何确定每类用户(进程)的优先级，以及应如何修改它们的优先级等，都属于策略问题，可将它们放入微内核外的进程(线程)管理服务器中。

2) 低级存储器管理

通常在微内核中，只配置最基本的低级存储器管理机制。如用于实现将用户空间的逻辑地址变换为内存空间的物理地址的页表机制和地址变换机制，这一部分是依赖于机器的，因此放入微内核。

而实现虚拟存储器管理的策略，则包含应采取何种页面置换算法，采用何种内存分配与回收策略等，应将这部分放在微内核外的存储器管理服务器中去实现。

1.5.4 微内核OS结构

3) 中断和陷入处理

大多数微内核操作系统都是将与硬件紧密相关的一小部分放入微内核中处理。此时微内核的主要功能，是捕获所发生的中断和陷入事件，并进行相应的前期处理。如进行中断现场保护，识别中断和陷入的类型，然后将有关事件的信息转换成消息后，把它发送给相关的服务器。

在微内核OS中，是将进程管理、存储器管理以及I/O管理这些功能一分为二，属于机制的很小一部分放入微内核中，另外绝大部分放在微内核外的各种服务器中来实现。

1.5.4 微内核OS结构

3．微内核操作系统的优点

1) 提高了系统的可扩展性

由于微内核OS的许多功能是由相对独立的服务器软件来实现的，当开发了新的硬件和软件时，微内核OS只须在相应的服务器中增加新的功能，或再增加一个专门的服务器。与此同时，也必然改善系统的灵活性，不仅可在操作系统中增加新的功能，还可修改原有功能，以及删除已过时的功能，以形成一个更为精干有效的操作系统。

2) 增强了系统的可靠性

这一方面是由于微内核是出于精心设计和严格测试的，容易保证其正确性；另一方面是它提供了规范而精简的应用程序接口(API)，为微内核外部的程序编制高质量的代码创造了条件。此外，由于所有服务器都是运行在用户态，服务器与服务器之间采用的是消息传递通信机制，因此，当某个服务器出现错误时，不会影响内核，也不会影响其它服务器。

1.5.4 微内核OS结构

3) 可移植性

作为一个好的操作系统，必须具备可移植性，使其能较容易地运行在不同的计算机硬件平台上。在微内核结构的操作系统中，所有与特定CPU和I/O设备硬件有关的代码，均放在内核和内核下面的硬件隐藏层中，而操作系统其它绝大部分(即各种服务器)均与硬件平台无关，因而，把操作系统移植到另一个计算机硬件平台上所需作的修改是比较小的。

4) 提供了对分布式系统的支持

由于在微内核OS中，客户和服务器之间以及服务器和服务器之间的通信，是采用消息传递通信机制进行的，致使微内核OS能很好地支持分布式系统和网络系统。

5) 融入了面向对象技术

在设计微内核OS时，采用了面向对象的技术，其中的“封装”、“继承”、“对象类”和“多态性”，以及在对象之间采用消息传递机制等，都十分有利于提高系统的“正确性”、“可靠性”、“易修改性”、“易扩展性”等，而且还能显著地减少开发系统所付出的开销。

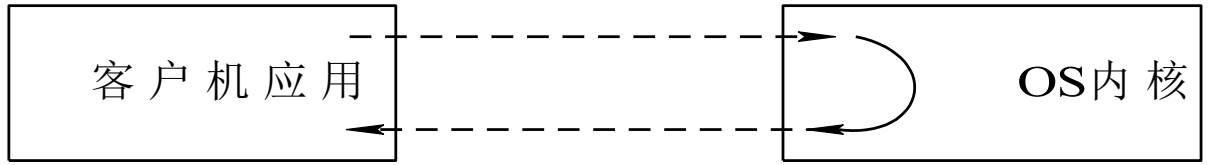
1.5.4 微内核OS结构

4．微内核操作系统存在的问题

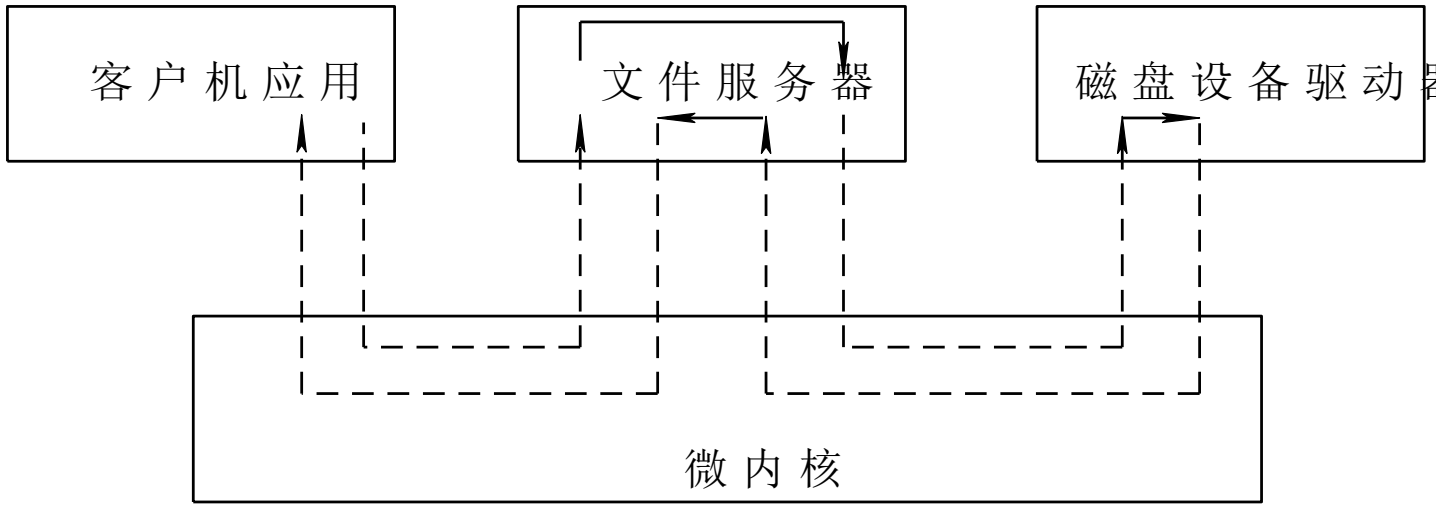
在微内核OS中，由于采用了非常小的内核，以及客户/服务器模式和消息传递机制，这些虽给微内核OS带来了许多优点，但由此也使微内核OS存在着潜在的缺点。其中最主要的是，较之早期OS，微内核OS的运行效率有所降低。

效率降低的最主要的原因是，在完成一次客户对OS提出的服务请求时，需要利用消息实现多次交互和进行用户/内核模式及上下文的多次切换。例如，当某个服务器自身尚无能力完成客户请求，而需要其它服务器的帮助时，如图1-11中所示，其中的文件服务器还需要磁盘服务器的帮助，这时就需要进行八次上下文的切换。

1.5.4 微内核OS结构



(a) 在整体式内核文件操作中的上下文切换



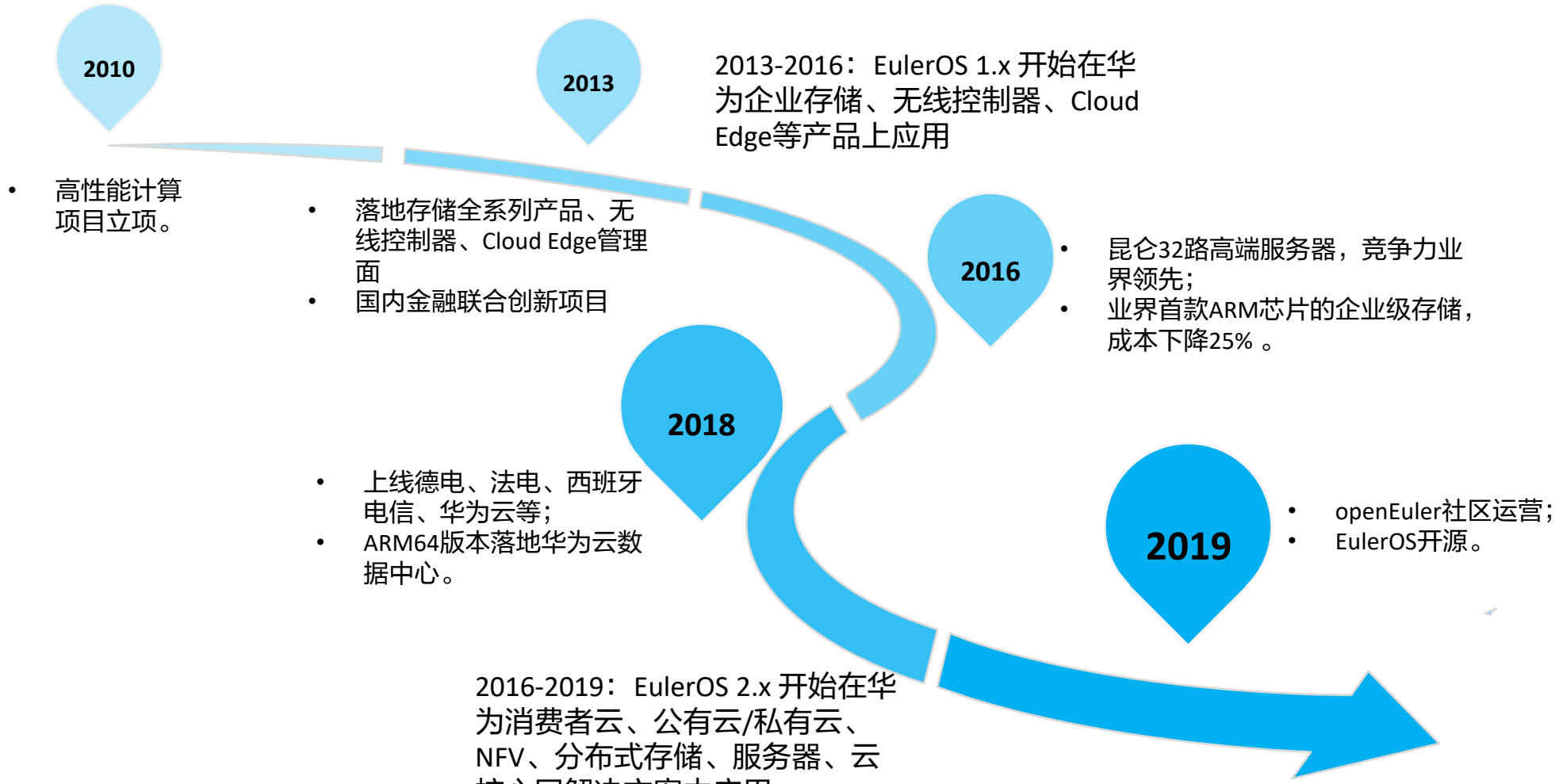
(b) 在微内核中等价操作的上下文切换

图1-11 在传统OS和微内核OS中的上下文切换

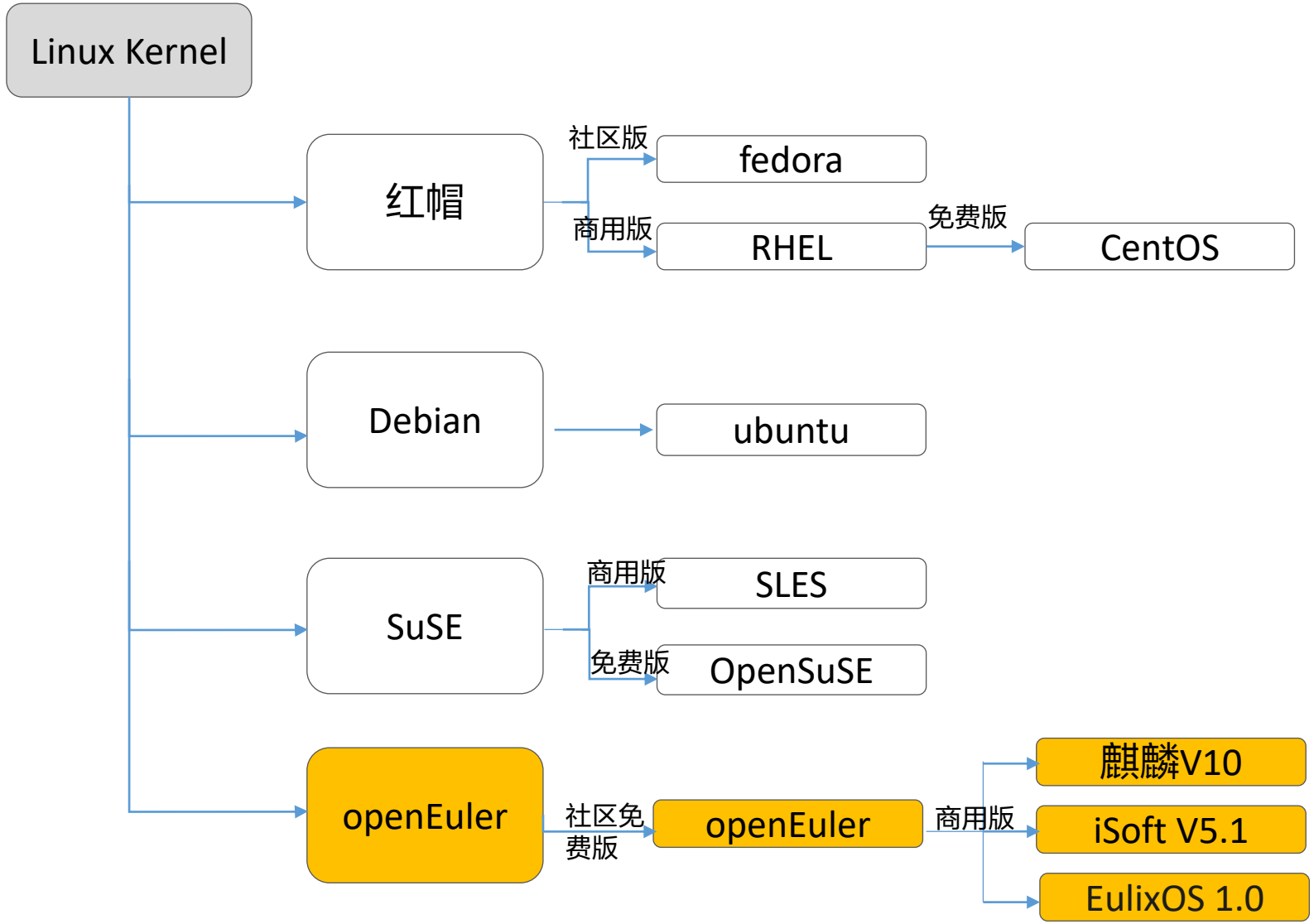
1.6 openEuler简介

- EulerOS是华为公司开发的一款基于Linux内核的服务器操作系统，支持x86和ARM（Advanced RISC Machine）等多种处理器架构，适用于数据库、大数据、云计算、人工智能等应用场景；
- 在近10年的发展中，EulerOS成功支持了华为各种产品解决方案，以安全、稳定、高效被业界认可；
- 随着云计算的兴起和鲲鹏芯片的发展，EulerOS成为与鲲鹏芯片配套最合适的软件基础设施；
- 为推动鲲鹏生态的发展，2019年底EulerOS被正式推送开源社区，命名为openEuler。所有开发者、企业、商业组织都可以使用openEuler社区版本，也可以基于社区版本发布自己二次开发的操作系统版本。
- https://openeuler.org/
- https://gitee.com/openeuler/

openEuler发展历程一览

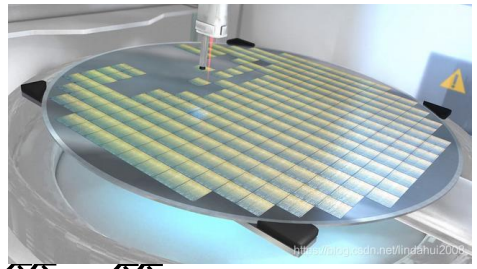


openEuler和主流OS系的关系



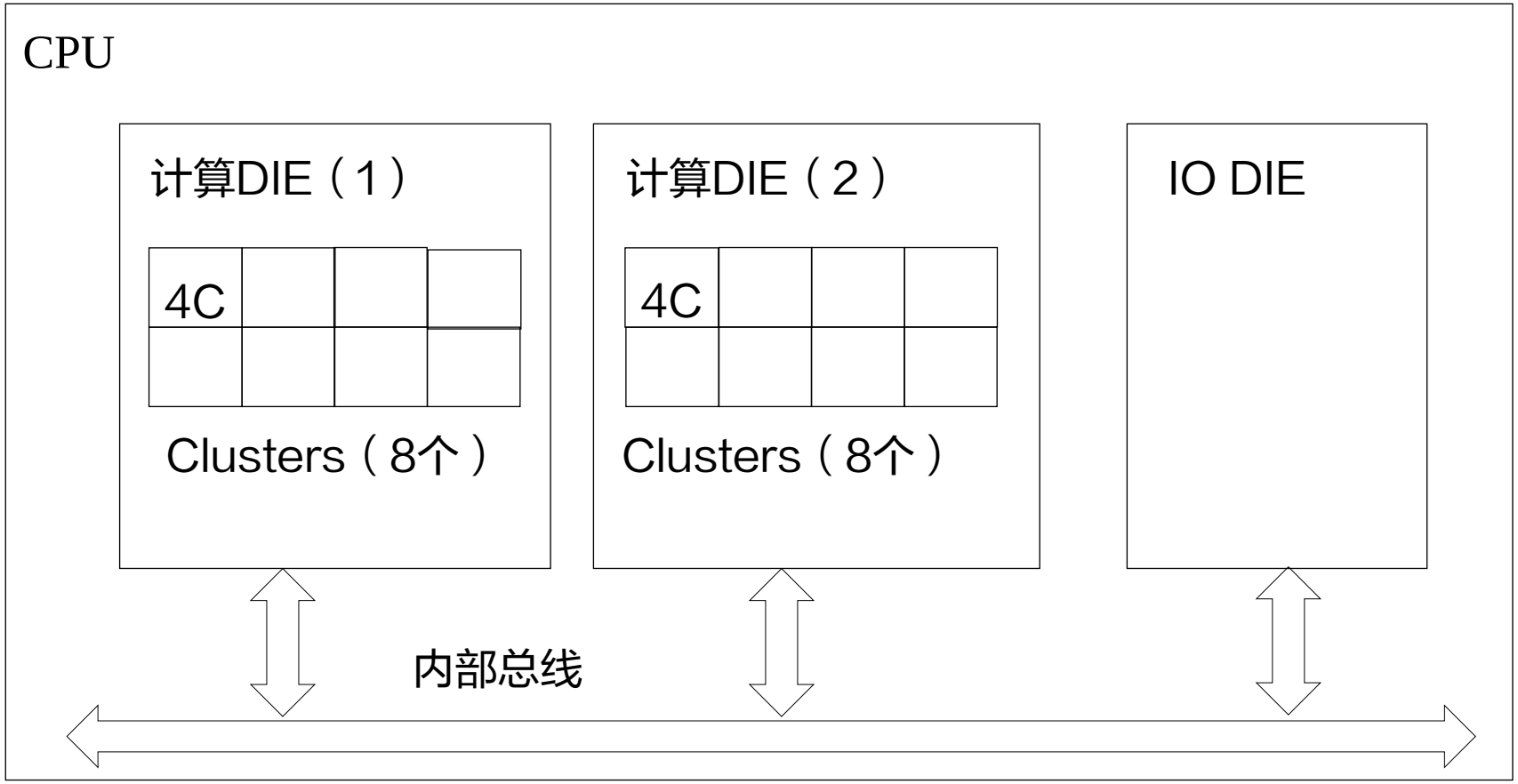
鲲鹏处理器

- 鲲鹏处理器是基于ARMv8-64指令集开发的通用处理器；CPU有很多指令集架构，最流行的就是Intel 的x86，在手机端占据统治地位的ARM等。从历史的发展来看，ARM的算力从2000年的不到5%， 发展到现在82%，非常迅速。



- 鲲鹏处理器的相关概念：
  - DIE - 芯片的最小物理单元。Kunpeng 920封装了3个DIE，两个用来做计算，第三个用来做IO；
  - Cluster – 若干个核(core)的集合。Kunpeng 920把4个core集合成为一个cluster，而一个DIE上有8个cluster；
  - Core - 真正的计算单元，我们在操作系统侧看到的“核”。

鲲鹏920芯片架构全景图

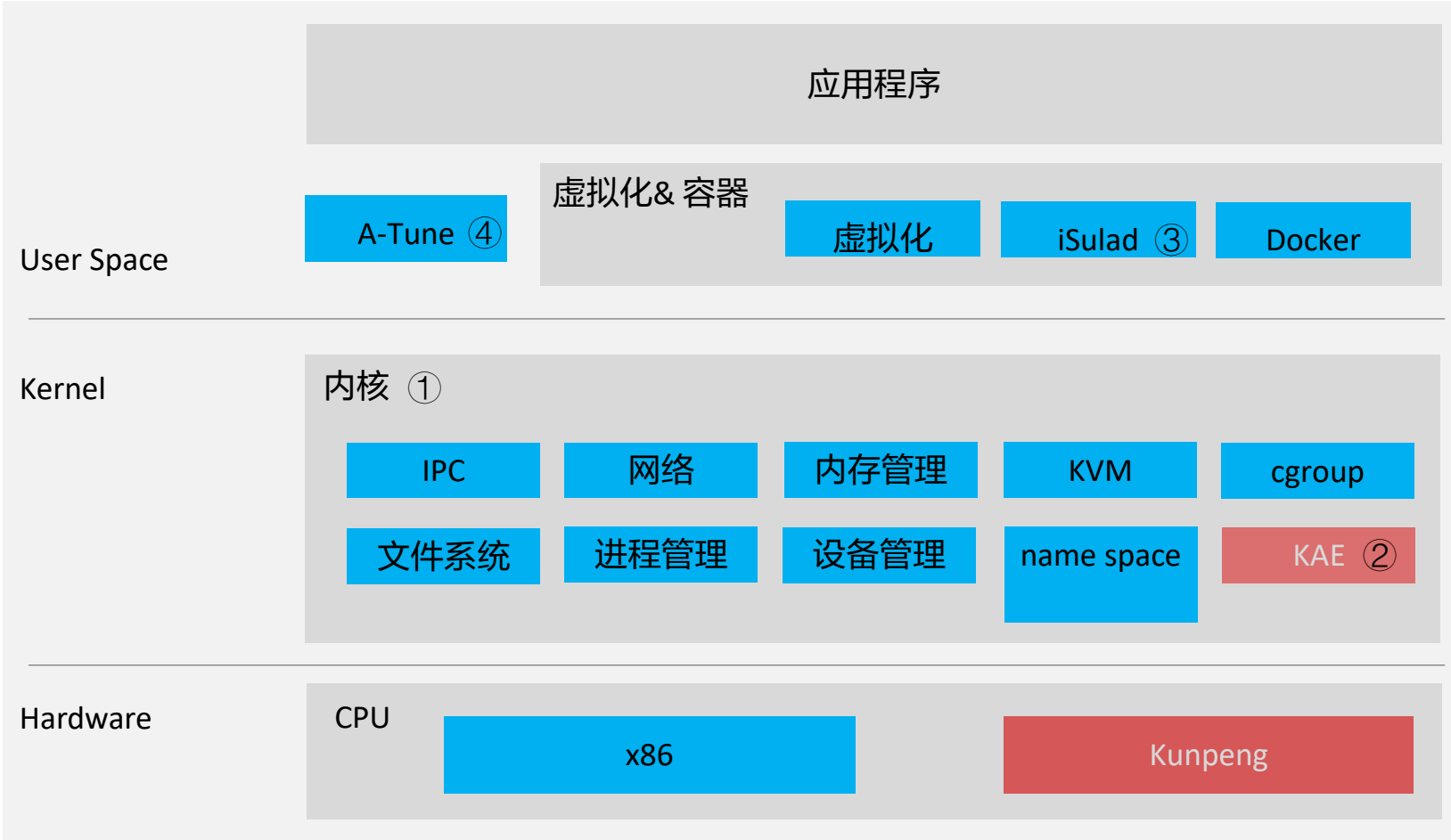


如图所示：

- 1片SoC上包含3个DIE，2个计算DIE，1个IO DIE
- 1个计算DIE中8个Cluster
- 1个Cluster中4个Core
- 因此一个鲲鹏920芯片中包含4*8*2=64个核
- 计算DIE上的每一个core具有自己的L1和L2级cache，所有的core共享L3级cache
- IO DIE上集成有网络模块、PCIe模块
- 这些DIE在芯片内部通过高速内部总线进行连接

鲲鹏处理器和openEuler

鲲鹏处理器在高性能、高吞吐、高集成、高能效方面有创新突破，作为华为公司的软件平台，openEuler天然支持鲲鹏处理器，并充分发挥处理器的各种特性，为此，openEuler对通用Linux操作系统作了增强。



目前，openEuler所做的增强包含了如下几个方面：

1. 多核调度技术
2. 软硬件协同：提供KAE（Kunpeng Accelerator Engine）引擎插件，加速硬件：加密/解密
3. 轻量级虚拟化：提供iSulad轻量级容器解决方案
4. 智能优化引擎：A-Tune：自动识别业务场景，使应用跑在最佳系统配置下

本章小结

- OS 定义、位置、作用、目的
- OS 类型及各自的特点
- OS 五大功能
- OS 特征
- OS 设计
- openEuler简介
- 概念：多道、虚拟、并发、并行、效率（吞吐量）、时间片、进程、批处理、脱机、交互性、响应时间、分时、透明、终端、接口、系统调用、系统开销、处理机状态、特权指令、中断