

本课的目的



《操作系统》是计算机科学与技术专业的**主要专业基础课和主干课**。

通过本课程的学习，使学生了解操作系统的**内部结构**，掌握操作系统的**设计方法、原理和实现技术**，熟悉操作系统的**操作和使用**，从而使学生能方便、灵活地使用计算机，能更好地在操作系统的基础上**建立自己的应用系统、开发自己的应用软件**。

为什么要学习操作系统



- 操作系统的功能在很多领域使用。

并发程序的开发 分布式系统 网络

等等都大量使用了操作系统的概念和技术。

- 操作系统的技巧也在很多领域使用。

如抽象、缓存、并发等。如程序设计中的抽象数据类型，抽象类。

- 学习操作系统是一件有趣的事情。

参考书



- 《计算机操作系统教程》 张尧学 史美林
清华大学出版社
- 《计算机的心智 操作系统之哲学原理》
邹恒明 机械工业出版社
- 《操作系统——内核与设计原理》
Operating Systems Internals and
Design Principles (Fourth Edition)
William Stallings

你用过的操作系统能做什么？

你知道操作系统不能做什么？

不做天气预报
不做房屋设计
不是编译程序

总之，操作系统不直接解决最终具体应用问题，也不负责编译源程序...

操作系统

为你完成所有

“硬件相关、应用无关”的工作，
以给你方便、效率、安全。

第一章 绪论



本章首先介绍操作系统基本概念及在学习操作系统中涉及的计算机基本概念;接着介绍操作系统的发展过程,旨在通过操作系统的演变过程对操作系统的基本概念有进一步的揭示;接下来介绍操作系统的基本功能、基本特征以及体系结构。

1.1 什么是操作系统

1.2 操作系统运行环境

1.3 操作系统的形成与发展

1.4 操作系统的功能和特性

1.5 操作系统的结构

1.6 Linux操作系统



1.1 什么是操作系统

程序是如何运行的

- 1.加电启动
- 2.装载操作系统
- 3.运行程序

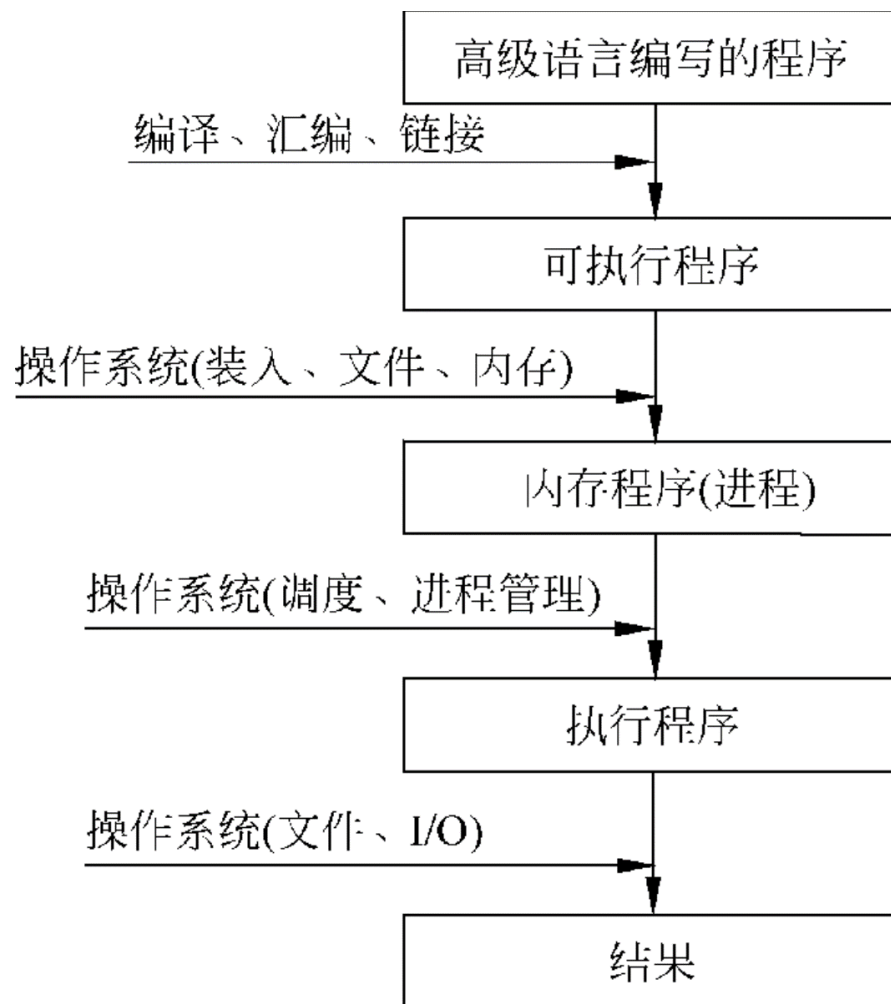


图1-1 程序的执行过程

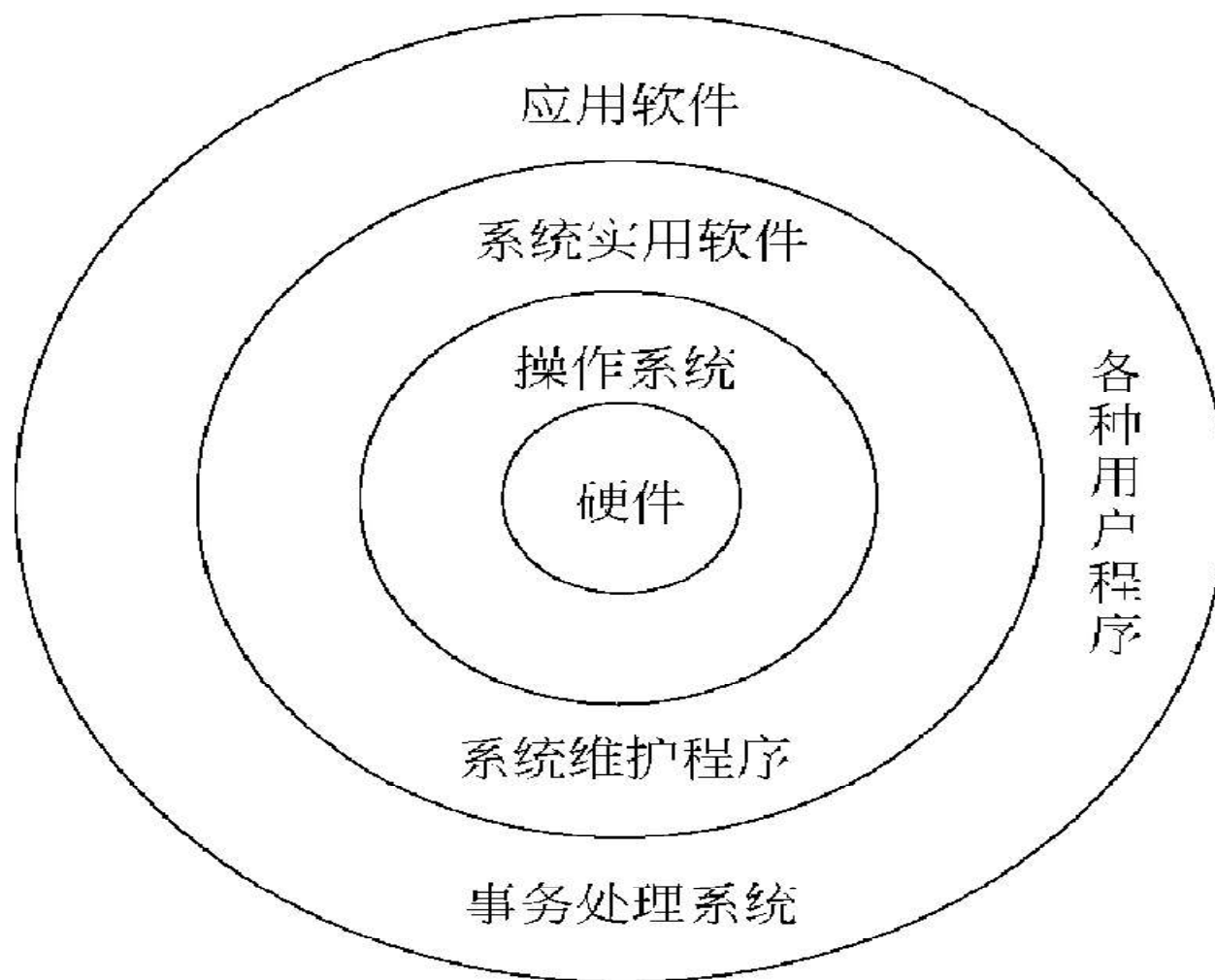


图1-2 计算机系统层次结构

没有装OS的计算机能做什么？（裸机）



计算机硬件+OS（WINDOWS），可操作计算机（虚拟机）

计算机硬件+OS（WINDOWS）+工具、应用软件（计算机系统）

OS的主要作用是：（介于用户和计算机之间）

- （1）管理系统资源：CPU、主存、I/O设备、文件。
- （2）为用户提供一个良好的接口。
- （3）最大限度提高资源使用效率。

操作系统：操作系统是直接控制和管理计算机软硬件资源的最基本的系统软件,它可以合理地组织计算机的工作流程,以方便用户充分、有效地利用这些资源,并增强整个计算机的处理能力。

1.2 操作系统运行环境

1.2.1 计算机的基本硬件元素

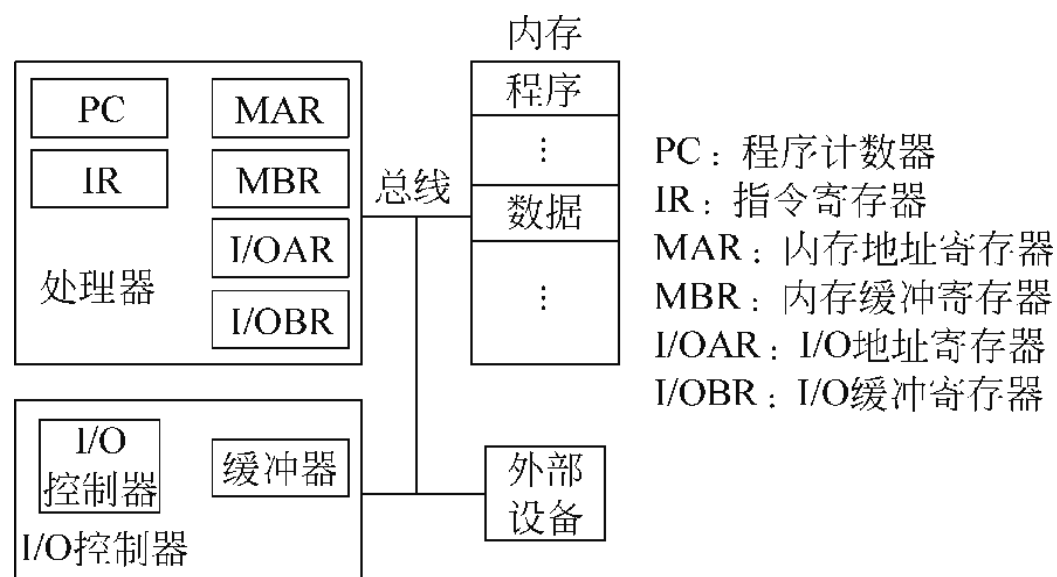


图1-3 计算机的基本硬件元素

1.2.2 与操作系统相关的几种主要寄存器

数据寄存器、地址寄存器、条件码寄存器、程序计数器PC、指令寄存器IR、程序状态字PSW、中断现场保护寄存器、过程调用堆栈

1.2.3 指令的执行

每个CPU都有一个**指令集合**。任何应用程序的运行都是通过指令的执行才得以实现的。指令的读入和执行过程称为一个执行周期。

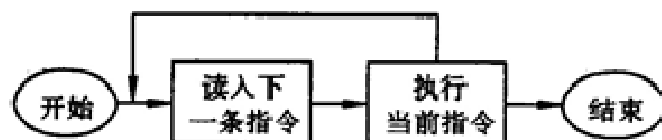
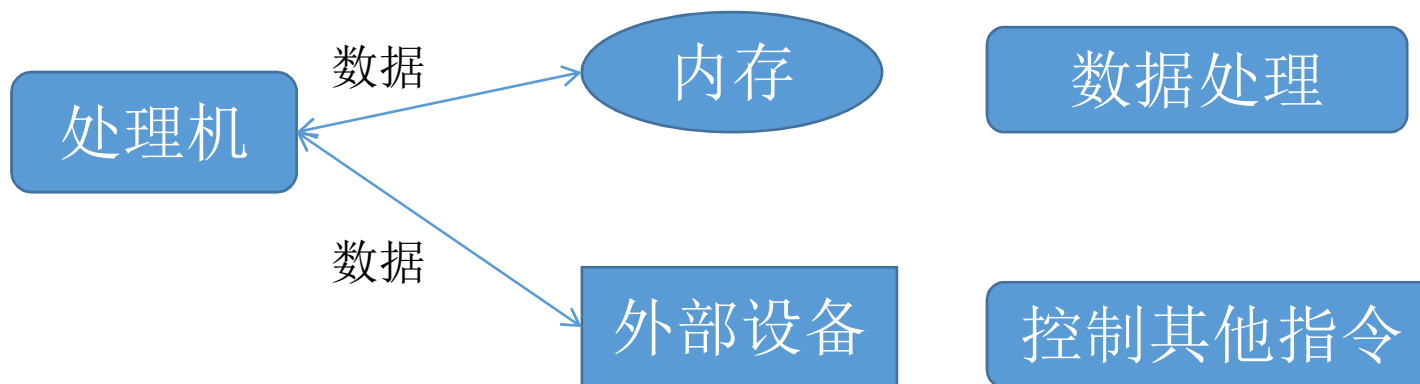


图1-4 指令的执行周期



1.2.4 中断

1. 中断概念

是指某个事件（例如，电源掉电、浮点运算溢出、外部设备传输完成或出错等）发生时，系统中止现运行程序的执行，引出处理事件程序对相应事件进行处理，处理完毕后返回断点继续执行。

引起中断的事件称**中断源**，如打印完成中断，其中断源是打印机。

发生中断时正在运行的程序被暂时停止，程序的暂停点称为断点。例如，某程序正在执行0200地址的指令被中断，那么，0200地址就是断点，在中断返回时就执行0200的下一条指令。

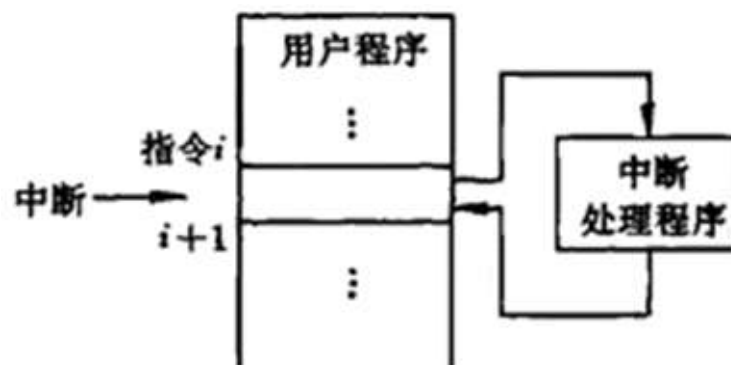


图1—5 中断的执行过程

2. 中断类型与优先级

内中断与外中断；不同的中断赋予不同的优先级。

3. 中断响应与中断处理程序

处理机发现有中断请求时，中止现运行程序的执行并自动引出中断处理程序的过程。

对中断事件进行处理的程序叫中断处理程序。如时钟中断处理、打印机完成中断处理、打印机缺纸中断处理等等。它是操作系统中与硬件最接近的一部分，是操作系统与硬件的接口。

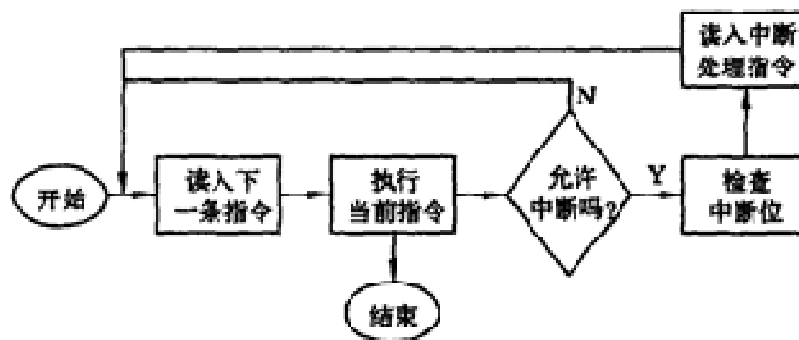


图1—6 中断处理时的指令执行周期

1.2.5 处理机状态及特权指令



1. 处理机的状态

核态 (Kernel Mode):

CPU执行操作系统程序时所处的状态。

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。

管态 (supervisor mode)

比核态的权限低，在此状态下允许使用一些用户态下不能使用的资源，但不能使用修改CPU状态的指令。无核态时，管态执行核态的全部功能。

用户态 (User Mode):

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。

当CPU执行用户程序时，CPU处于用户态，在执行的过程中出现了中断（或自陷）时，系统转去处理与中断有关的事件，这时CPU由用户态转换成核态（或管态），处理完中断后，返回断点继续执行用户程序，这时CPU由核态转换成用户态。

2. 特权指令集



在核态下操作系统可以使用所有指令，包括一组特权指令。

- 允许和禁止中断；
- 在进程之间切换处理机；
- 存取用于内存保护的寄存器；
- 执行输入和输出操作；
- 停止一个中央处理机的工作。

在下列情况下，由用户态转向核态：

- 用户程序要求操作系统的服务，系统调用；
- 发生一次中断；
- 在用户程序中产生了一个错误的状态；
- 在用户程序中企图执行一条特权指令；

从核态转回用户态用一条指令实现，这条指令也是特权指令。
一般情况下是中断返回指令。

1.3 操作系统的形成与发展

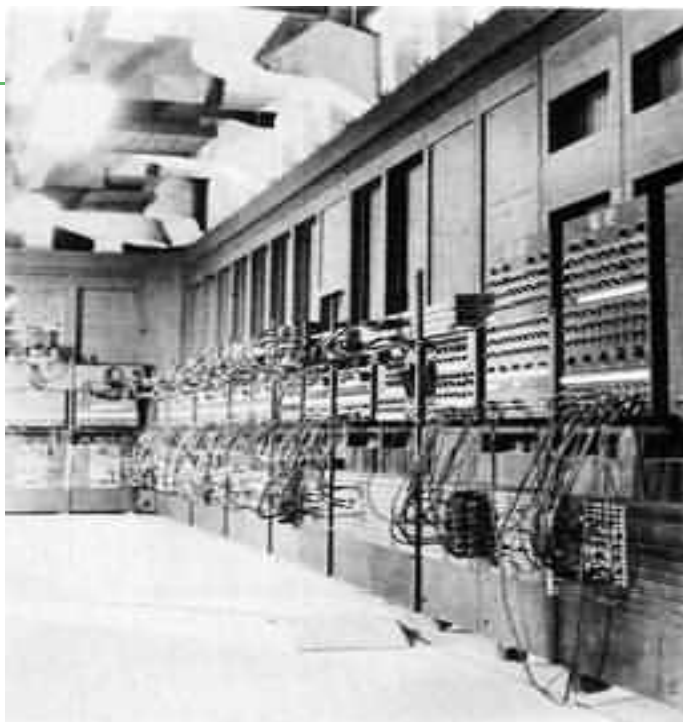
对一个事物，了解它的过去，才能懂得它的现在，也才能预见它的将来。是什么推动了操作系统的发展呢？操作系统在其发展过程中有什么变化呢？

1.3.1 操作系统发展的基础

1. 体系结构 简单—复杂 单机—多机 集中—分散
2. 软件资源 高级语言、编译系统、服务程序等等
3. 用户需求 越来越高

1.3.2 手工操作阶段 脱机/联机I/O

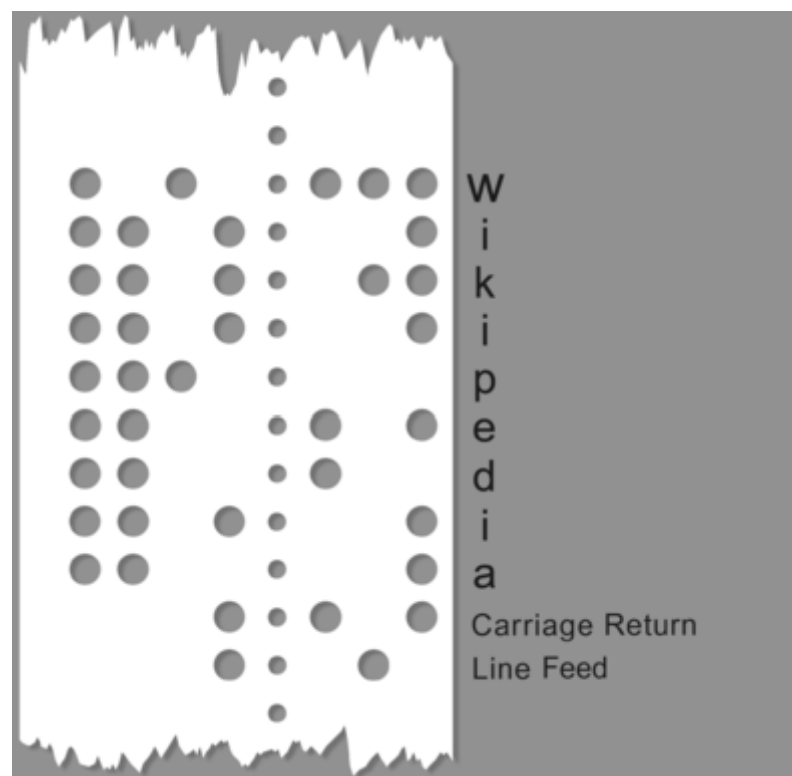
从第一台计算机诞生(1945年)到20世纪50年代中期的计算机，属于第一代计算机。此时的计算机是利用成千上万个真空管做成的，它的运行速度仅为每秒数千次，但体积却十分庞大，且功耗也非常高。这时还未出现OS。



ENIAC

长30.48米，宽1米，占地面积约63平方米，30个操作台，约相当于10间普通房间的大小，重达30吨，耗电量150千瓦，造价48万美元。它包含了17,468 真空管7,200水晶 二极管, 1,500 中转, 70,000 电阻器, 10,000 电容器, 1500继电器, 6000多个开关，每秒执行5000次加法或400次乘法，是继电器计算机的1000倍、手工计算的20万倍。

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。当程序运行完毕并取走计算结果之后，才让下一个用户上机。



(1) 用户独占全机。此时，计算机及其全部资源只能由上机用户独占。

(2) CPU等待人工操作。当用户进行装带(卡)、卸带(卡)等人工操作时，CPU及内存等资源是空闲的。

可见，人工操作方式严重降低了计算机资源的利用率，此即所谓的人机矛盾。随着**CPU**速度的提高和系统规模的扩大，人机矛盾变得日趋严重。此外，随着**CPU**速度的迅速提高而**I/O**设备的速度却提高缓慢，这又使**CPU与I/O设备之间速度不匹配的矛盾更加突出**。为了缓和此矛盾，曾先后出现了**通道技术、缓冲技术**，但都未能很好地解决上述矛盾，直至后来又引入了脱机输入/输出技术，才获得了较为令人满意的结果。

1.3.3 批处理系统

1. 单道批处理系统

(1) 处理过程

上世纪50年代中期发明了晶体管，晶体管替代真空管来制作计算机，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。

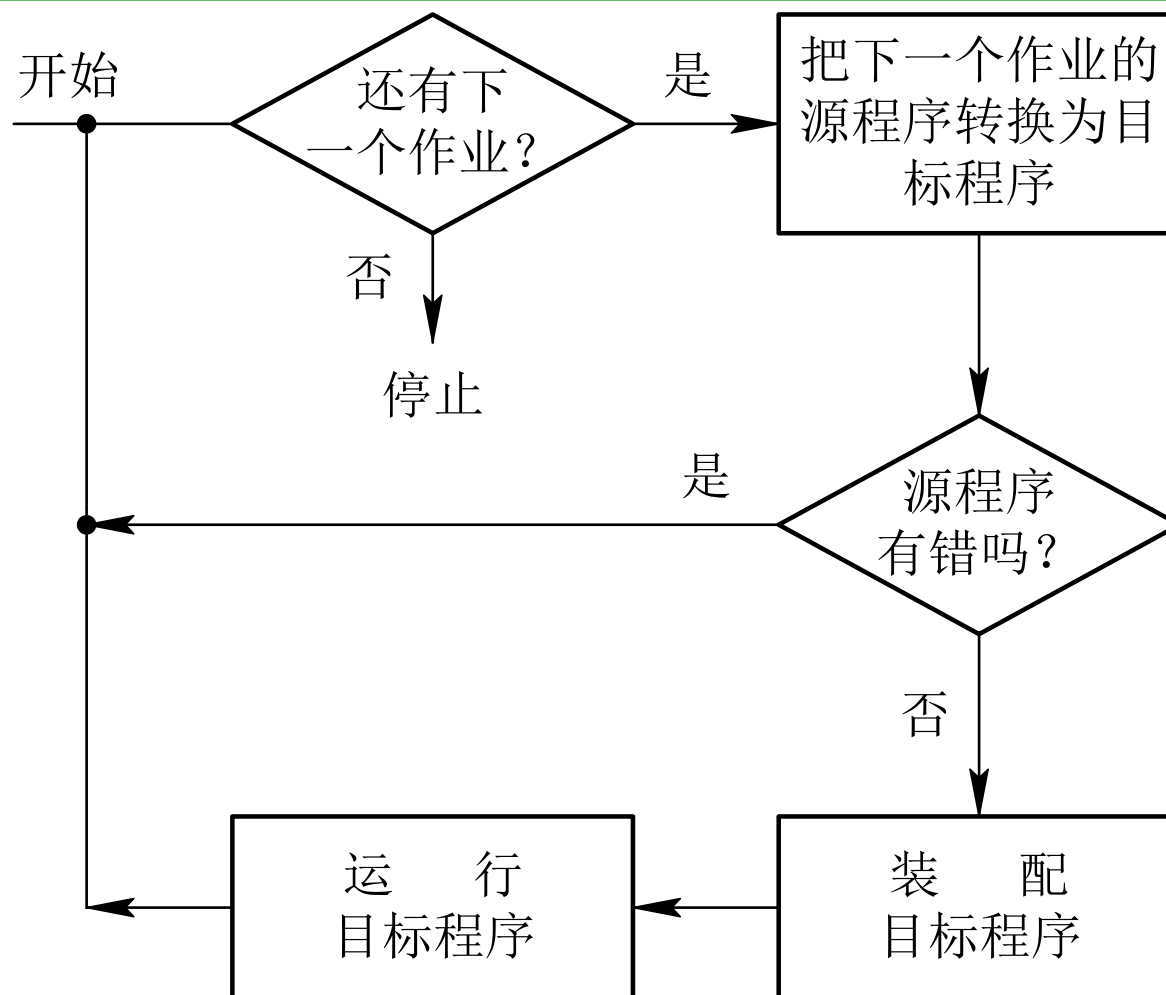


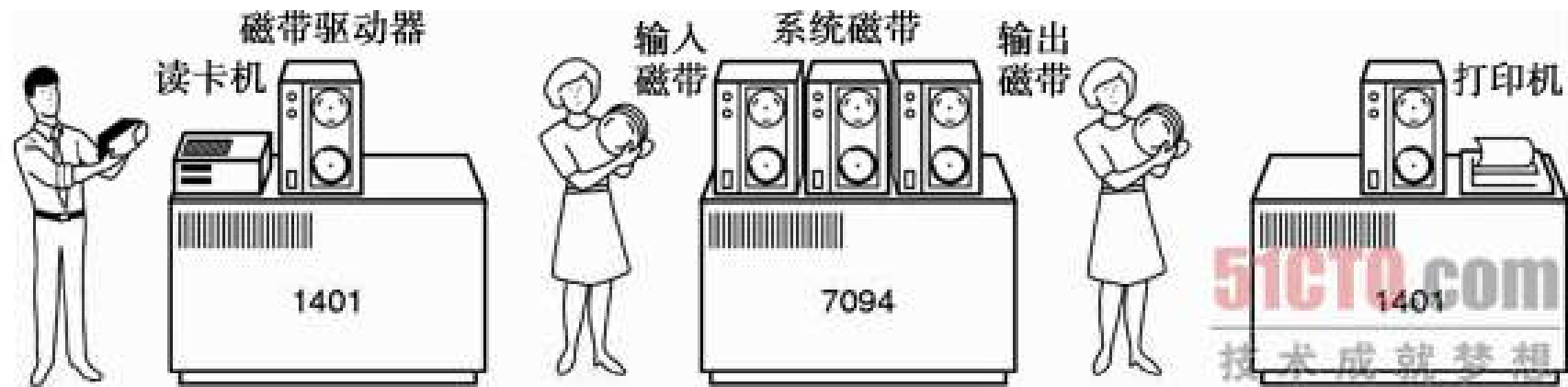
图1-4 单道批处理系统的处理流程

(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。严格地说，它只能算作是OS的前身而并非是现在人们所理解的OS。该系统的主要特征如下：

自动性、顺序性、单道性。

批处理系统示意



重要实例



- IBM开发的FORTRAN监视系统FMS，用于IBM7094；
- IBM 开发的基于磁带的工作监控系统，用于IBM7090和7094；
- 密歇根大学1959年开发的MAD/UMES（密歇根大学算法译码器和密歇根大学执行系统），用于IBM7094。

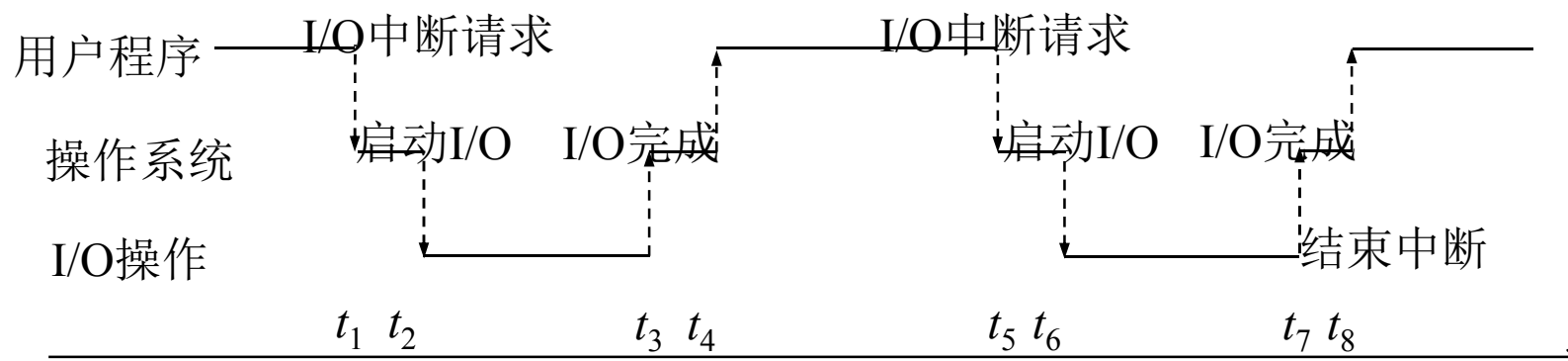
2. 多道程序系统

多道程序设计：把一个以上的程序同时存放在主存中，并且“同时”处于运行状态，这些作业共享处理机时间和外部设备等资源。

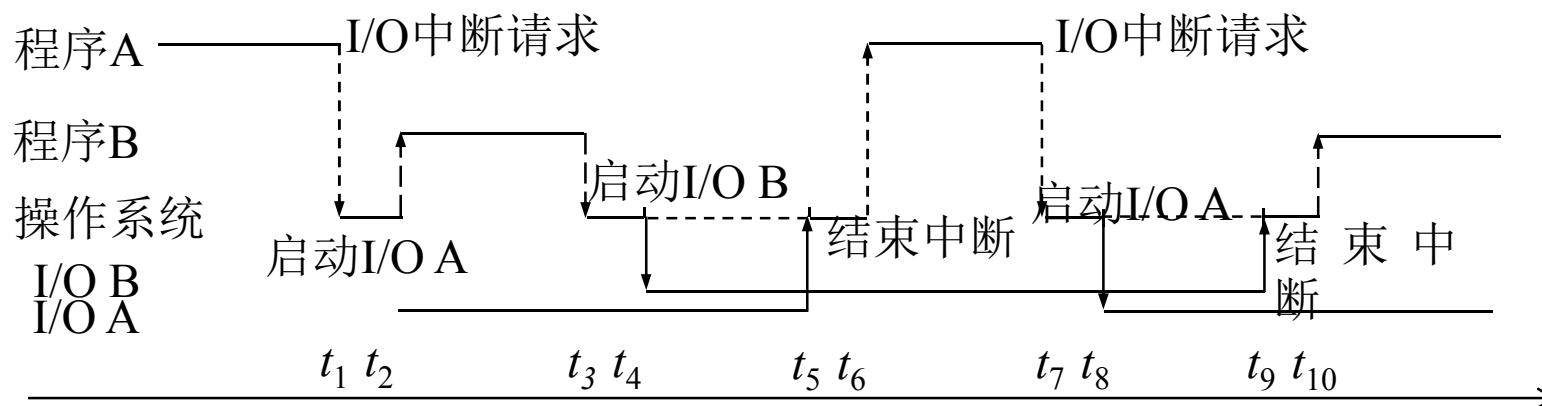
对于单处理机系统而言：

多道程序的特点：多道、宏观上并行、微观上串行。在批处理系统中采用多道程序技术，就形成了多道批处理系统。

引入多道程序设计技术的根本目的是提高CPU的利用率，充分发挥并行性。包括程序之间、设备与CPU之间均并行工作。



(a) 单道程序运行情况



(b) 两道程序运行情况

例1-1:

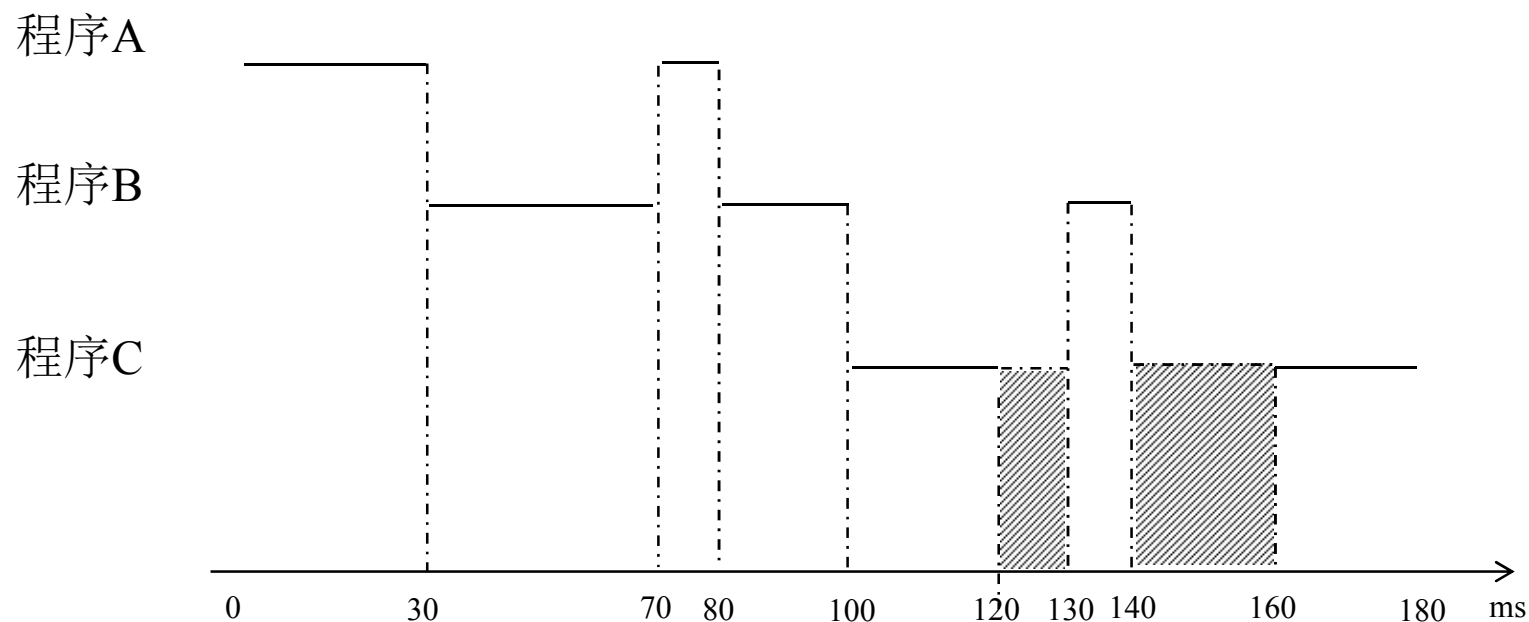
设有A、B、C三道程序，其执行过程分别如下：

A: C11=30ms I12=40ms C13=10ms

B: C21=60ms I22=30ms C23=10ms

C: C31=20ms I32=40ms C33=20ms

其中，Cij表示内部计算过程，Iij表示I/O操作，三者的优先权顺序为ABC。



3. 多道批处理系统



工作方式，将用户意图、数据、程序利用系统提供的作业控制命令提交给操作员，操作员将其输入外存，由OS控制、调度各作业的运行，最后输出结果。是一种非人工的干预方式。有单道和多道批量系统。

目标：是提高作业的吞吐量，同时兼顾作业的周转时间。

特征：“批量”，成批的完成。

优点：作业吞吐量大，资源利用率高，OS开销较小。

缺点：使用户使用起来不太方便。

(1) 交互能力较弱。

(2) 作业的平均周转时间长。

1.3.4 分时系统

分时：把CPU的运行时间划分成一个个微小的时间片，并把这些时间片依次轮流地分配给各终端用户程序。

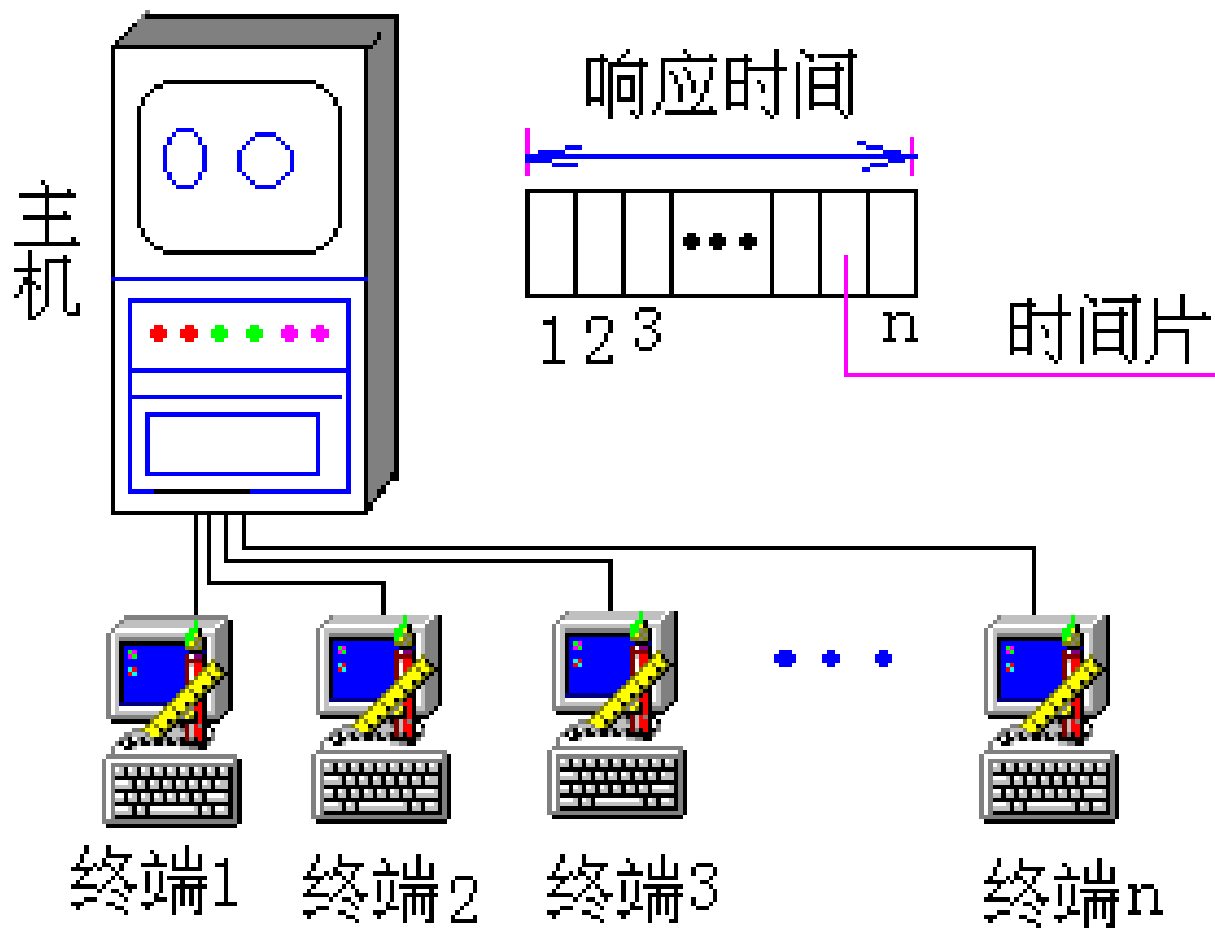
1. 分时系统：多个用户同时使用同一台计算机，系统能分时轮流地为各终端用户服务并能及时地对用户的请求予以响应。

2. 设计目标：对用户的响应的及时性，及时地响应和服务于联机用户。

3. 特点：

- (1) 交互性
- (2) 同时性（多路性）
- (3) 独立性（独占性）
- (4) 及时性

分时 操作系统(Time Sharing OS)



重要实例

最有名的是MULTICS和UNIX
部分商用领域的CTSS

傲慢的代价：MULTICS操作系统



分时操作系统里面最著名的应该是**MULTICS**和**UNIX**。前面讲过，**IBM**在其捐赠**7094**给**UM**和**MIT**时附加了一些条件，而这些附加条件使得学校非常恼火，但又不能拒绝。为此，**UM**开发了**UMES**系统以应对这个问题。**MIT**则将**UMES**移植到了自己的**7094**中。后来大家觉得只保存中间结果还不是最好的办法，毕竟频繁地保存中间结果等帆船比赛结束后再进行重载仍存在麻烦，于是就想开发一个**可支持多用户的分时操作系统**，以便一劳永逸地解决这个问题。这个时候**MIT**想到了密歇根的**R.M.Graham**，将其请来。

在**R.M.Graham**的主持下，来自贝尔实验室、**DEC**（**Digital Equipment Corporation**，美国数字仪器公司）和**MIT**的设计人员同仇敌忾（针对**IBM**的傲慢），一起努力，开始了**MULTICS**分时操作系统的研制。不过，在**MULTICS**还没有开发出来时，开发团队内部出现了分歧，贝尔实验室的几个人越来越看不惯**DEC**和**MIT**的人，觉得这伙人的思维方式跟自己不一样，做的东西不专业，觉得和他们一起做研究把自己的水平降低了，就自立门户，搞出了**UNIX**，并因此获得图灵奖。而**UNIX**的出现，使得**MULTICS**从一面世，就不能挺立，真是中国历史上“既生瑜，何生亮”在计算机操作系统历史上的完美演绎。不过，**MIT**最后还是做出了一个应用于部分商用领域的分时操作系统**CTSS**（**Compatible Time Sharing System**）。而**DEC**公司则做出了非常成功的**VMS**操作系统。

1.3.5 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。

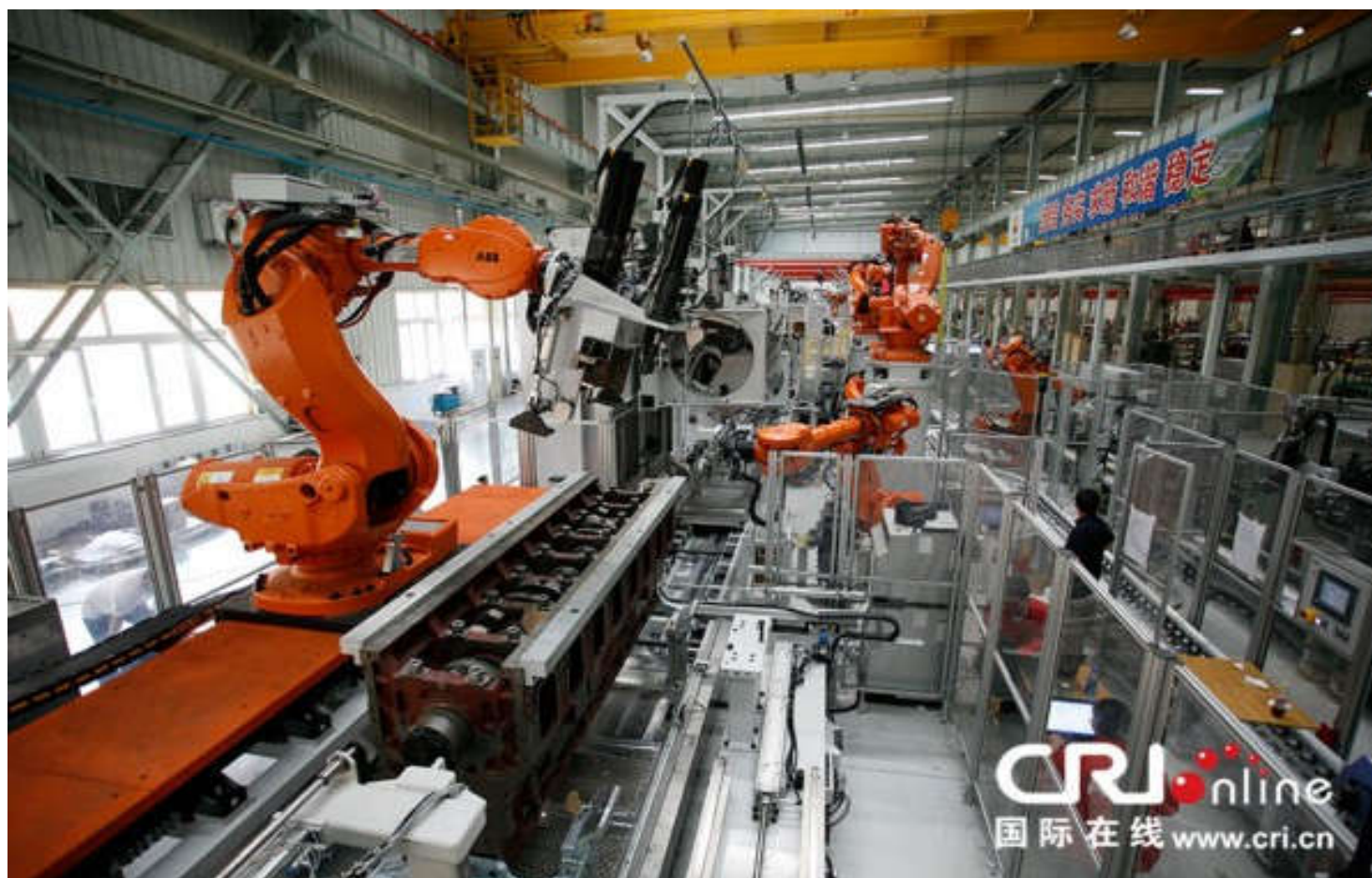
1. 应用需求

虽然多道批处理系统和分时系统已能获得较为令人满意的资源利用率和响应时间，从而使计算机的应用范围日益扩大，但它们仍然不能满足以下某些应用领域的需要。

(1) 实时控制。

当把计算机对某些工业进程进行监视，以形成以计算机为中心的控制系统时，系统要求能实时采集现场数据，并对所采集的数据进行及时处理。如工业装配线，一个部件从流水线上一个工作站流到下一个工作站时，这个操作必须在规定时间内完成，否则就可能造成流水线瘫痪，影响企业的生产和利润。





类似地，也可将计算机用于对武器的控制，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。此外，随着大规模集成电路的发展，已制作出各种类型的芯片，并可将这些芯片嵌入到各种仪器和设备中，用来对设备的工作进行实时控制，这就构成了所谓的智能仪器和设备。在这些设备中也需要配置某种类型的、能进行实时控制的系统。通常把用于进行实时控制的系统称为实时系统。

(2) 实时信息处理。通常，人们把用于对信息进行实时处理的系统称为实时信息处理系统。典型的实时信息处理系统有飞机或火车的订票系统、情报检索系统和视频点播系统等。

2. 目标：实时响应及处理的能力和高可靠性。

3. 特征

- (1) 及时性
- (2) 交互性
- (3) 安全可靠
- (4) 多路性

1.3.7 网络操作系统

特点：

- (1) 计算机网络是一个互连的计算机系统的群体。
- (2) 这些计算机是自治的，每台计算机有自己的操作系统，各自独立工作，它们在网络协议控制下协同工作。
- (3) 系统互连要通过通信设施(硬件、软件)来实现。
- (4) 系统通过通信设施执行信息交换、资源共享、互操作和协作处理，实现多种应用要求。互操作和协作处理是计算机网络应用中更高层次的要求特征。

网络操作系统与单处理机的操作系统区别：网络管理、通信、资源共享、系统很安全、多种网络应用。

1.3.8 分布式操作系统

(1) 对于各种分布式系统并没有制定标准的协议。作为计算机网络，现在已制定了明确的通信网络协议体系结构及一系列协议族。计算机网络的开发都遵循协议，当然，计算机网络也可认为是一种分布式系统。

(2) 分布式系统要求一个统一的操作系统，实现系统操作的统一性。对于各个物理资源的管理，高级操作系统和各私有操作系统之间，不允许有明显的主从管理关系。在计算机网络中，实现全网的统一管理的网络管理系统已成为越来越重要的组成部分。

(3) 系统的透明性。分布式操作系统负责全系统的资源分配和调度、任务划分、信息传输控制协调工作，并为用户提供一个统一的界面，标准的接口，用户通过这一界面实现所需要的操作和使用系统资源，至于操作定在哪一台计算机上执行或使用哪台计算机的资源则是系统的事，用户是不用知道的，也就是系统对用户是透明的。

(4) 分布式系统的基础是网络。对于多机合作和系统重构、健壮性和容错能力有更高的要求。

1.3.9 嵌入式操作系统



嵌入式系统已经成为操作系统的最新发展之一。用于PDA, 移动电话, 家用电器, 医疗设备, 机器人等等。嵌入式系统指执行专用功能并被内部计算机控制的设备或系统。

嵌入式微处理器特点:

- (1) 对实时多任务有很强的支持能力。
- (2) 具有功能很强的存储区保护功能。
- (3) 可扩展的处理器结构。
- (4) 嵌入式微处理器必须功耗很低。

嵌入式计算机系统同通用型计算机系统相比具有以下特点：

- （1）嵌入式系统通常是面向特定应用的。
- （2）嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物。这一点就决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。
- （3）嵌入式系统的硬件和软件都必须高效率地设计。
- （4）嵌入式系统和具体应用有机地结合在一起，它的升级换代也是和具体产品同步进行。
- （5）为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中，而不是存贮于磁盘等载体中。
- （6）嵌入式系统本身不具备自举开发能力，即使设计完成以后用户通常也是不能对其中的程序功能进行修改的，必须有一套开发工具和环境才能进行开发。

1.3.6 个人操作系统

个人计算机上的操作系统是一种联机的交互式的单用户操作系统。由于是个人专用，因此在多用户和分时所要求的对处理机调度、存储保护方面将会简单的多。然而，由于个人计算机的普及，对于提供更为方便友好的用户接口的要求会愈来愈迫切。随着多媒体技术的引入，要求计算机有一个具有高速数据处理能力的实时多任务操作系统。



Windows

Fun but prone to crashing



Linux

Reliable but boring



OS X

Looks nice but is useless



FreeBSD

Meek and goes unnoticed

1.4 操作系统的功能和特性

1.4.1 操作系统的功能

1. （处理机）进程管理
2. 存储管理
3. 设备管理
4. 文件管理
5. 用户接口

1.4.2 操作系统的基本特性



1. 并行性

并行性和并发性(Concurrence)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，即利用每个处理机来处理一个可并发执行的程序，这样，多个程序便可同时执行。

2. 共享性

(1) 互斥共享方式

打印机

磁带机

同一时刻只能由一个进程占用。

我们把这种资源共享方式称为互斥式共享，而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或独占资源。计算机系统的大多数物理设备，以及某些软件中所用的栈、变量和表格，都属于临界资源，它们要求被互斥地共享。为此，在系统中必需配置某种机制来保证诸进程互斥地使用独占资源。

(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。典型的可供多个进程“同时”访问的资源是磁盘设备，一些用重入码编写的文件也可以被“同时”共享，即若干个用户同时访问该文件。

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。

3. 不确定性（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方能执行。

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。当正在执行的进程提出某种资源要求时，如打印请求，而此时打印机正在为其它某进程打印，因此正在执行的进程必须等待，且放弃处理机，直到打印机空闲，并再次把处理机分配给该进程时，该进程方能继续执行。可见，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。

内存中的每个进程在何时能获得处理机运行，何时又因提出某种资源请求而暂停，以及进程以怎样的速度向前推进，每道程序总共需多少时间才能完成，等等，这些都是不可预知的。

由于各用户程序性能的不同，比如，有的侧重于计算，而有的程序I/O多，这样，很可能是先进入内存的作业后完成，而后进入内存的作业先完成。或者说，进程是以人们不可预知的速度向前推进，此即进程的不确定性或异步性(Asynchronism)。

尽管如此，但只要在操作系统中配置有完善的进程同步机制，且运行环境相同，作业经多次运行都会获得完全相同的结果。因此，异步运行方式是允许的，而且是操作系统的一个重要特征。

4. 虚拟性



(1) 时分复用技术

时分复用，亦即分时使用方式，它最早用于电信业中。为了提高信道的利用率，人们利用时分复用方式，将一条物理信道虚拟为多条逻辑信道，将每条信道供一对用户通话。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。

利用多道程序设计技术，把一台物理上的处理机虚拟为多台逻辑上的处理机，在每台逻辑处理机上运行一道程序。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。

例如，原来的打印机属于临界资源，而通过虚拟设备技术，可以把它变为多台逻辑上的打印机，供多个用户“同时”打印。

2. 空分复用技术

早在上世纪初，电信业中就使用**频分复用技术**来提高信道的利用率。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。早期的频分复用只能将一条物理信道划分为十几条到几十条话路，后来又很快发展成上万条话路，每条话路也只供一对用户通话。之后，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，这样，机器上便有了四个虚拟磁盘。当用户要访问D盘中的内容时，系统便会访问卷2中的内容。

2) 虚拟存储器技术

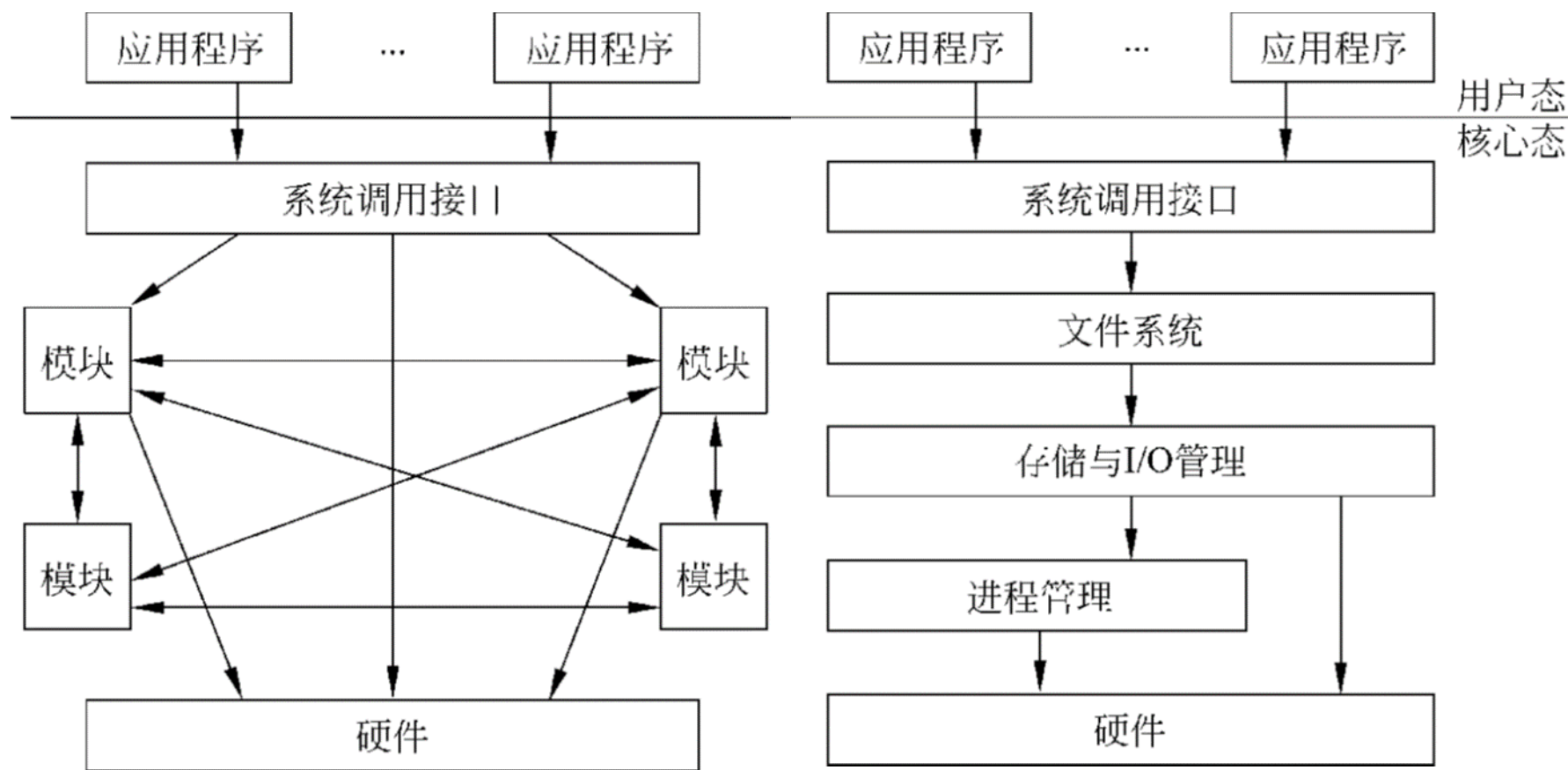
在单道程序环境下，处理机会会有很多空闲时间，内存也会有很多空闲空间，显然，这会使处理机和内存的效率低下。如果说时分复用技术是利用处理机的空闲时间来运行其它的程序，使处理机的利用率得以提高，那么空分复用则是利用存储器的空闲空间来存放其它的程序，以提高内存的利用率。

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。（调页和请求置换）

1.5 操作系统结构

1.5.1 模块组合及层次结构



1.5.2 微内核结构



1. 微内核操作系统

在微内核OS中是将很多通用操作的功能从内核中分离出来（如文件系统，设备驱动，网络协议栈等），只保留最基本的内容放入微内核中，另外绝大部分放在微内核外的各种服务器中来实现。微内核结构由一个非常简单的硬件抽象层和一组比较关键的原语或系统调用组成；这些原语，仅仅包括了建立一个系统必需的几个部分；如线程管理，地址空间和进程间通信等。

2. 微内核操作系统优点

提高了可扩展性，增强了可靠性，增强了可移植性，提供了对分布式系统的支持，融入了面向对象技术。

3. OS内核易形成系统的瓶颈

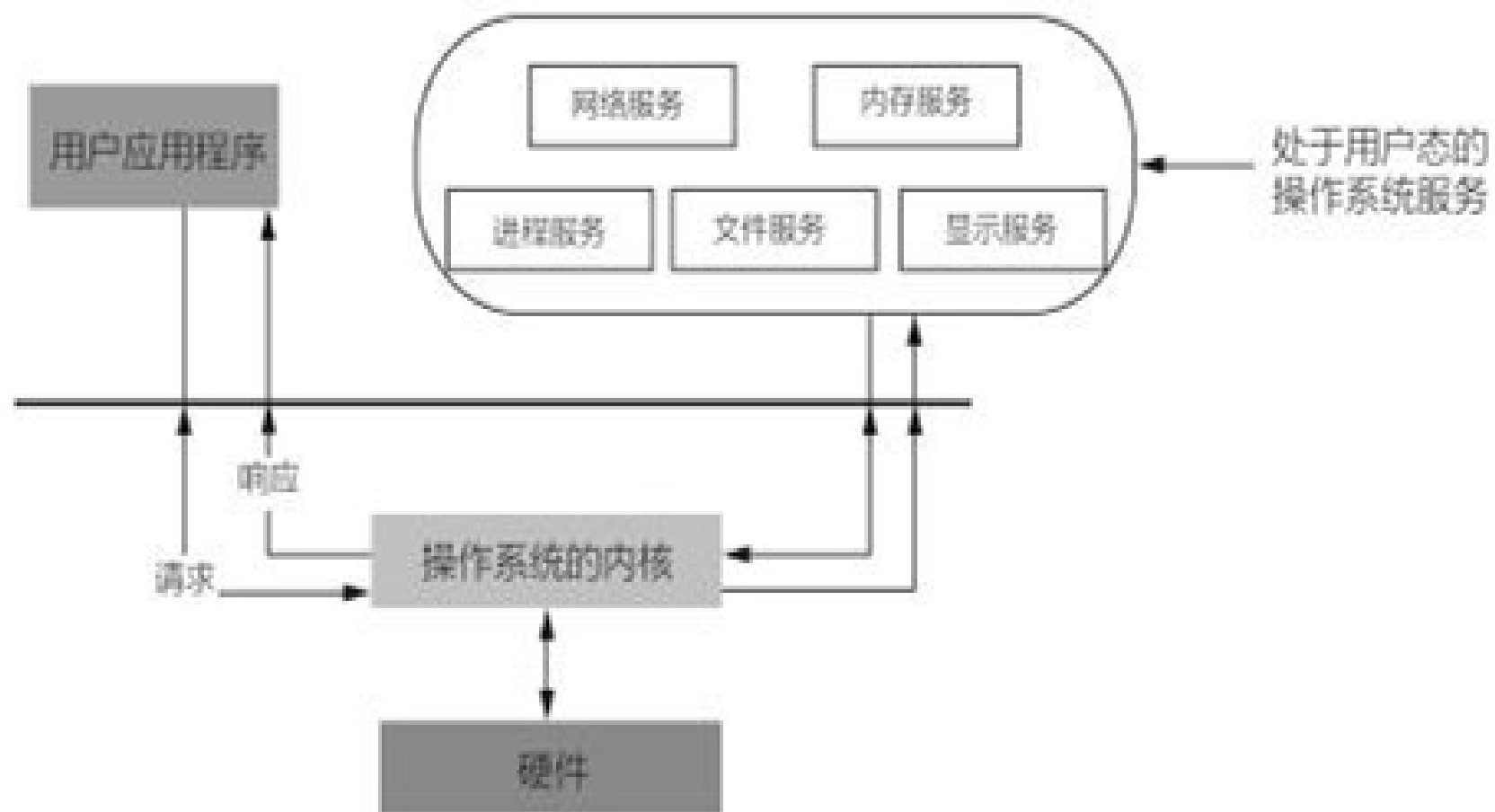
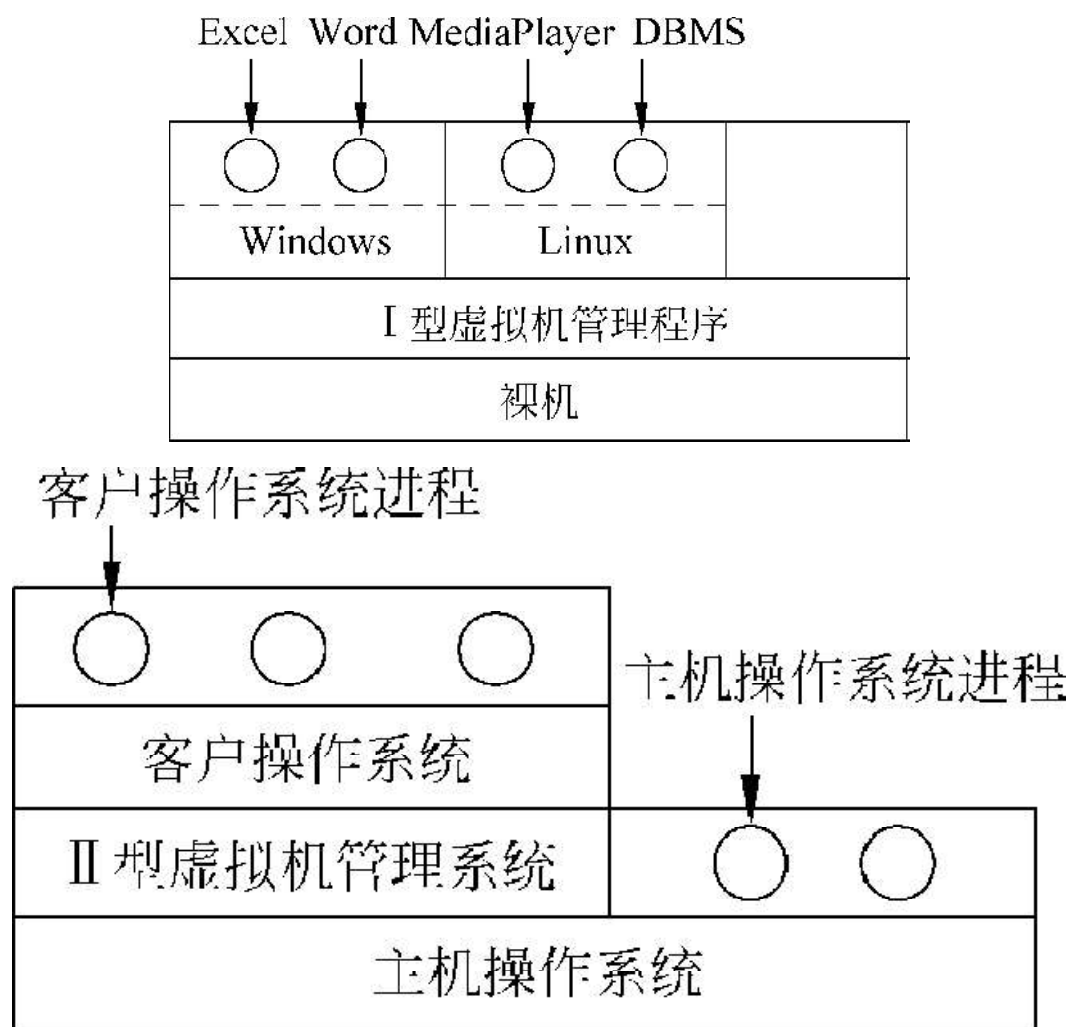


图1-11 微内核结构

1.5.3 虚拟机结构

①多道程序处理能力；②提供一个比裸机更方便扩展界面的计算机，提供这个功能的系统被称为虚拟机管理程序。



1.6 Linux操作系统

1.6.1 Linux发展历程



Linus Torvalds
李纳斯·托沃兹

1991年初，21岁的Linus Torvalds就读于芬兰的赫尔辛基大学，1991年9月17日，Torvalds将自己的开发的系统源程序完整地上传到FTP服务器上，供大家下载测试。

1991年7月3日他在一个名为“comp.os.minix”讨论组中发了一个帖子：“各位网友好！我现在正在Minix系统下做一个项目，对POSIX标准很感兴趣。有谁能向我提供一个（最好）是机器可读形式的最新的POSIX规则？能有FTP地址就更好了。”

“这花费了我大量的精力：编程——睡觉——编程——睡觉——编程——吃饭（饼干）——编程——睡觉——编程——洗澡（冲冲了事）——编程。”

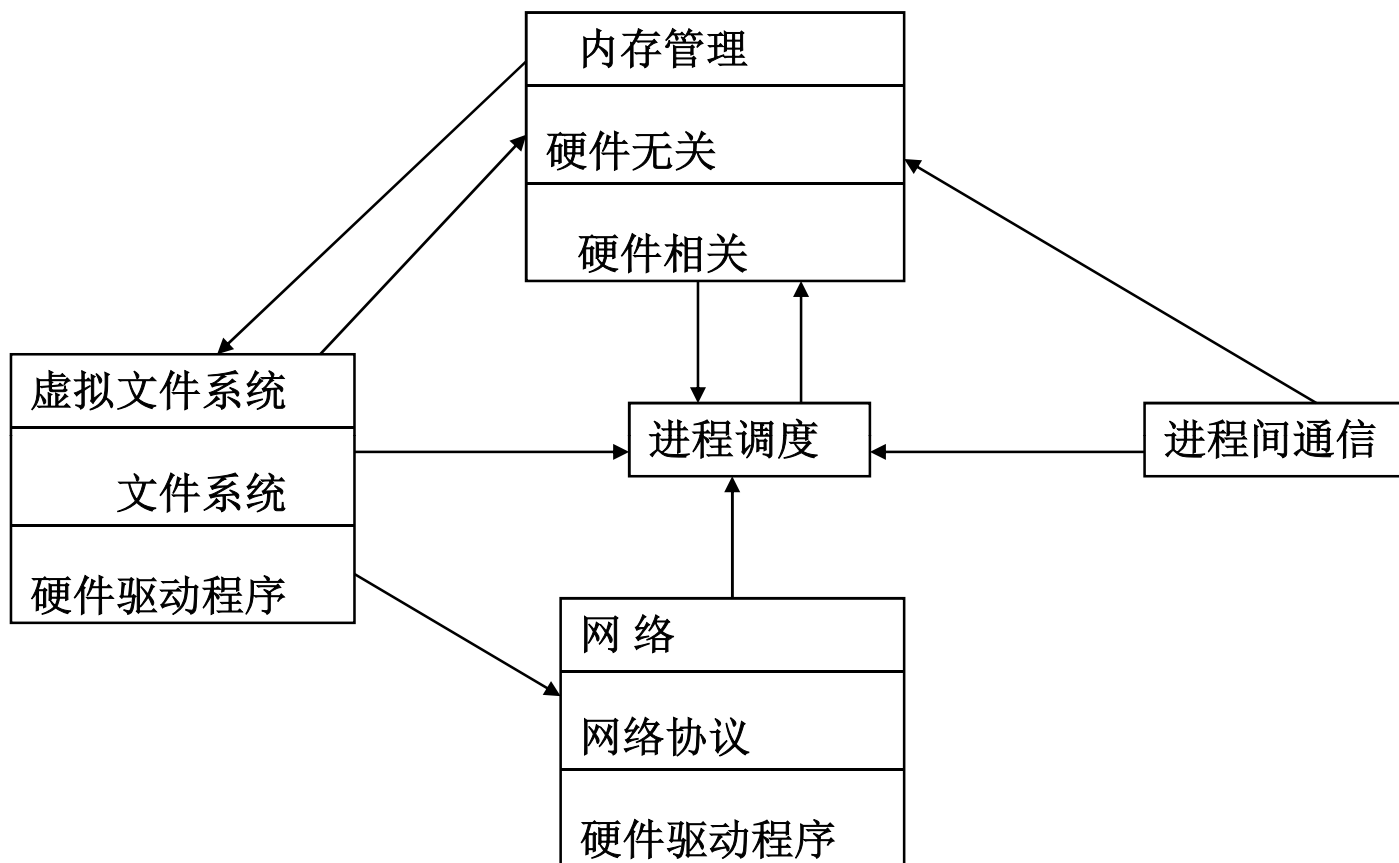
“那个夏天我除了伏在电脑面前，其他什么都没做。这么说一点也不夸张。芬兰四月到八月的日子是一年中最美好的时光。人们到布满小岛的海上航船，去海滩上晒日光浴，到夏日小木屋中消闲。但是我却在没日没夜地工作，不知哪一天是周末，哪一天是工作日。黑色窗帘遮蔽了几乎昼夜灿烂的阳光，也遮蔽了整个世界。有时候或许是夜晚？我会从床上爬起来，直接坐到离床仅几英尺远的电脑旁。毫不夸张地说，我和电脑之外的世界几乎没有任何联系。”

1.6.2 Linux的特点

1. 多用户, 多任务
2. 免费, 开源
3. 安全, 稳定
4. 丰富、安全的网络功能
5. 支持多种平台
6. Linux的不足

支持Linux的应用软件较Windows少一些; 许多硬件设备面对Linux的驱动程序也不足。

1.6.3 Linux内核结构



1-14 Linux内核子系统及其之间的关系

即将开始我们的探索之旅

