

操作系统

周建芳

电话: 17786528628

Mail:zhoujf2000@163.com

操作系统课程的特点：

实践性强（从实践总结出原理）

涉及面广（并行程序，性能问题，结构问题，程序方法论，软件工程，等等）

错综复杂：纵横交叉

如何学好操作系统？

注意学习方式的变化

教师只指出要点
要通过自学、研读参考书掌握内容

整理笔记相当重要
不能只通过PowerPoint来复习课程

善于发现问题、提出问题
要努力寻求问题的答案

课程评价方式

形成性评价60+总结性评价40

形成性评价:

1、平时表现20

课堂考勤+教学活动参与度 (问卷、作业、随堂练习等)

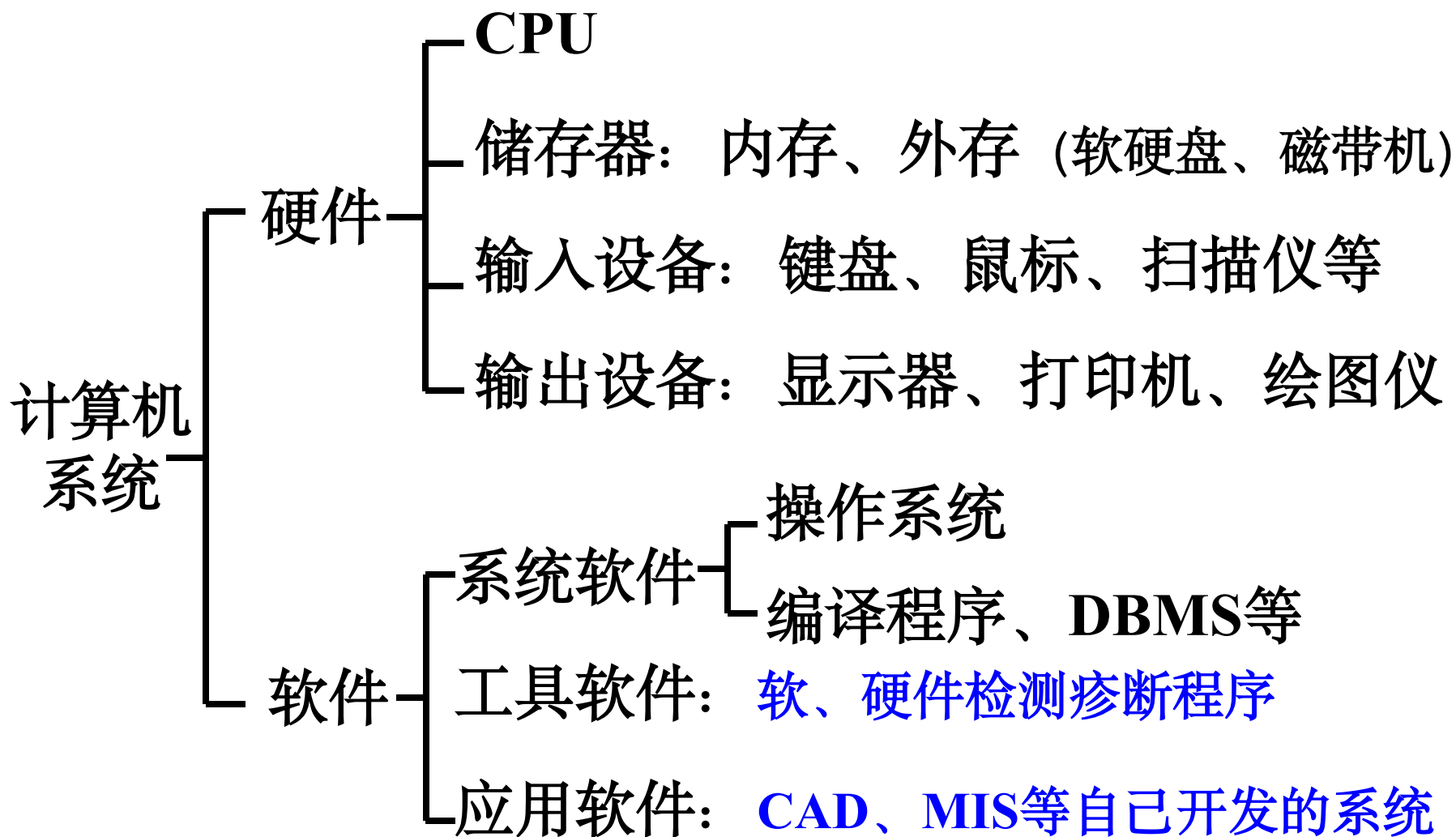
2、作业完成情况40 (正确率)

总结性评价:

期末闭卷考试

1.1 什么是操作系统

1、计算机系统的组成



概念

1、硬件：组成计算机系统的设备或机器，是“看得见，摸得着”的物理部件，它是组成计算机系统的基础。包括中央处理器（CPU）、内存存储器、外存储器、输入设备和输出设备，其中CPU与内存存储器合称为主机，外存储器、输入设备和输出设备合称为外部设备。

2、裸机：只有硬件部分，还未安装任何软件系统的电脑叫做裸机。

特点：功能有限，能够识别和执行机器指令，使用极不方便。

？ ？ ？

概念

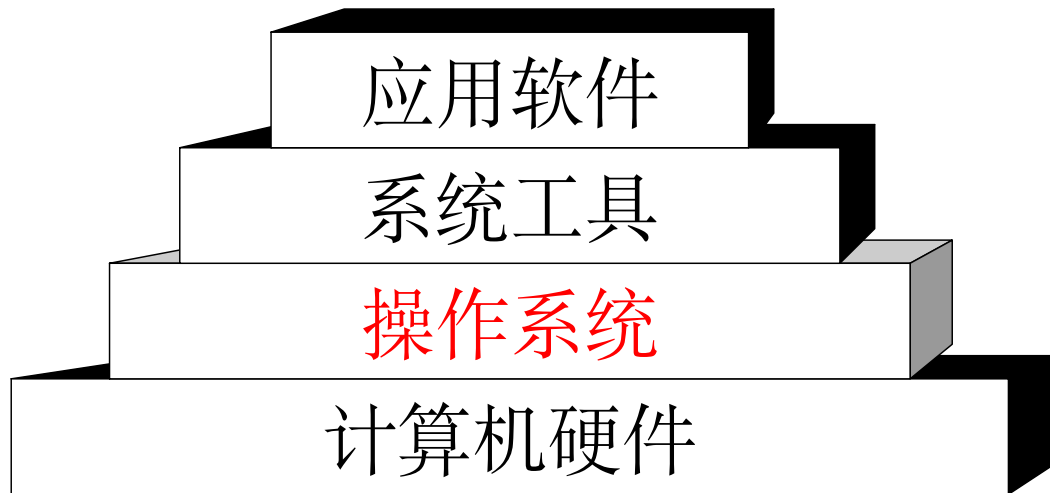
3、**软件**：组成计算机系统的**程序、数据和文档**。程序是指令的有序集合；数据是信息在计算机中的表示，是计算机处理的对象；文档是各种说明文本，是软件操作的辅助性资源。。

4、**虚拟机**：在**已有计算机系统**上覆盖了一层软件之后形成的计算机系统。

特点：使用方便，与已有计算机系统相比，功能更强大。

1.1 什么是操作系统

2、操作系统在计算机系统中的地位



操作系统在整个计算机系统中处于**软硬结合**部位。是覆盖在裸机上的**第一层**软件。

向上：为所有其他软件包括系统软件和应用软件提供软件支撑环境。

向下：统一管理和控制所有硬件，屏蔽硬件的底层细节，向用户提供方便易用的接口。

总结：操作系统在计算机系统中处于十分重要的**核心**地位，象胶水一样将计算机系统的硬件和软件粘合起来。

1.1 什么是操作系统

3、引入操作系统的目标

 **有效性**（系统管理人员的观点）：

管理和分配硬件、软件资源，合理地组织计算机的工作流程

 **方便性**（用户的观点）

提供良好的、一致的用户接口，弥补硬件系统的类型和数量差别

 **可扩充性**（开放的观点）：

硬件类型和规模、操作系统本身的功能和管理策略、多个系统间的资源共享和互操作。

1.1 什么是操作系统

4、操作系统的作用（从三个方面来看）

1、从资源管理的观点——OS作为资源管理器：

OS的首要任务是管理计算机系统中硬件与软件资源，使其得到充分而有效的利用：

- 1) 跟踪资源状态：时刻掌握计算机系统中资源的使用情况。
- 2) 分配资源：处理对资源的使用请求，协调冲突，确定资源分配算法。
- 3) 回收资源：回收用户释放的资源，以便下次重新分配。
- 4) 保护资源：负责对系统资源的保护，避免受破坏。

1.1 什么是操作系统

操作系统作为资源管理器的观点是目前人们对操作系统认识的一个主要观点，根据此观点，人们将计算机资源划分为四大类：

- 1) 处理机
- 2) 存储器
- 3) I/O设备
- 4) 信息（程序与数据）

相应可将操作系统分为四类管理器：

- 1) 处理机管理
- 2) 存储管理
- 3) 设备管理
- 4) 信息管理（文件系统）

1.1 什么是操作系统

 从软件分层、扩充机器的观点——虚拟机：

提供硬件的高层界面(虚拟机),取消硬件限制

- 1) 操作系统提供无限的内存、无限的CPU
- 2) 扩充机器,功能更强大,使用更方便

【虚拟机】：使用户和程序员在不必涉及和了解硬件工作细节的情况下能方便的使用计算机,而为用户所提供的的一个等价的扩展计算机,称为虚拟计算机。

1.1 什么是操作系统

 **从服务用户的观点——OS用户与裸机间接口：**

操作系统为方便用户使用计算机提供了二级访问接口：

- 1) **命令接口**：命令行、菜单式、命令脚本式、图形用户接口GUI；
- 2) **调用接口**：形式上类似于过程调用，在应用编程中使用。
- 3) **图形接口**：图形用户界面GUI，方便用户使用。

编程更为容易，软件可移植性增强；使用更为方便。

1.2 操作系统的发展史

操作系统发展的**主要动力是用户的需求**，并随着计算机硬件技术、软件技术的发展而不断发展完善。

1) **提高资源的利用率和系统性能**：计算机发展的初期，计算机系统昂贵，用作集中计算

2) **方便用户**：用户上机、调试程序，分散计算时的事务处理和非专业用户。

3) **硬件的发展**：CPU的位宽度（指令和数据）、快速外存、新硬件、新技术的出现等。

4) **计算机体系结构的不断发展**。

1.2 OS的发展史—手工

1、手工操作 (1946~50年代, 电子管时代)

【特点】：计算机资源昂贵，没有操作系统，软件发展相对滞后，只能用机器语言编程

【工作方式】：

- 用户：用户既是程序员、操作员，还是计算机专业人员；
- 编程语言：为机器语言；
- 输入输出介质：纸带或卡片；

【计算机的工作特点】：

- 用户独占全机：用户独占计算机所有资源，资源利用率低；
- CPU等待用户：计算前，手工装入纸带或卡片；计算完成后，手工卸取纸带或卡片；CPU利用率低；

- **【主要矛盾】：**

- 计算机处理能力的提高，手工操作的低效率
- 用户独占全机的所有资源；

$$\eta_{\text{cpu}} = T_{\text{use}} / (T_{\text{use}} + T_{\text{idle}})$$

随着器件技术的发展（遵循摩尔定律和新摩尔定律），**cpu**更快，人机矛盾冲突到不可调和得矛盾，新技术的出现将导致操作系统的发展进入新的阶段。

1.2 OS的发展史—单道 (monitor)

2、单道批处理系统 (50年代末 ~ 60年代中, 晶体管)

【现状】：硬件：①cpu更快，②出现了较为高速的存储介质如磁带，磁鼓等，③出现了专门的外围机

软件：出现了一些较为高级的程序设计语言及其编译程序，JCL (job control language) 等

【批处理中的作业的组成】：

- 用户程序
- 数据
- 作业说明书 (作业控制语言)

【批】：

- 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)

【工作原理】：

利用磁带把若干个作业分类编成作业执行序列（批），每个批作业由一个专门的**监督程序（Monitor）（操作系统的雏形！！！）**自动依次处理。可使用汇编语言开发。

处理过程见教材p7图1-4

1.2 OS的发展史—单道

批处理方式可分为联机批处理和脱机批处理。

1) 联机批处理

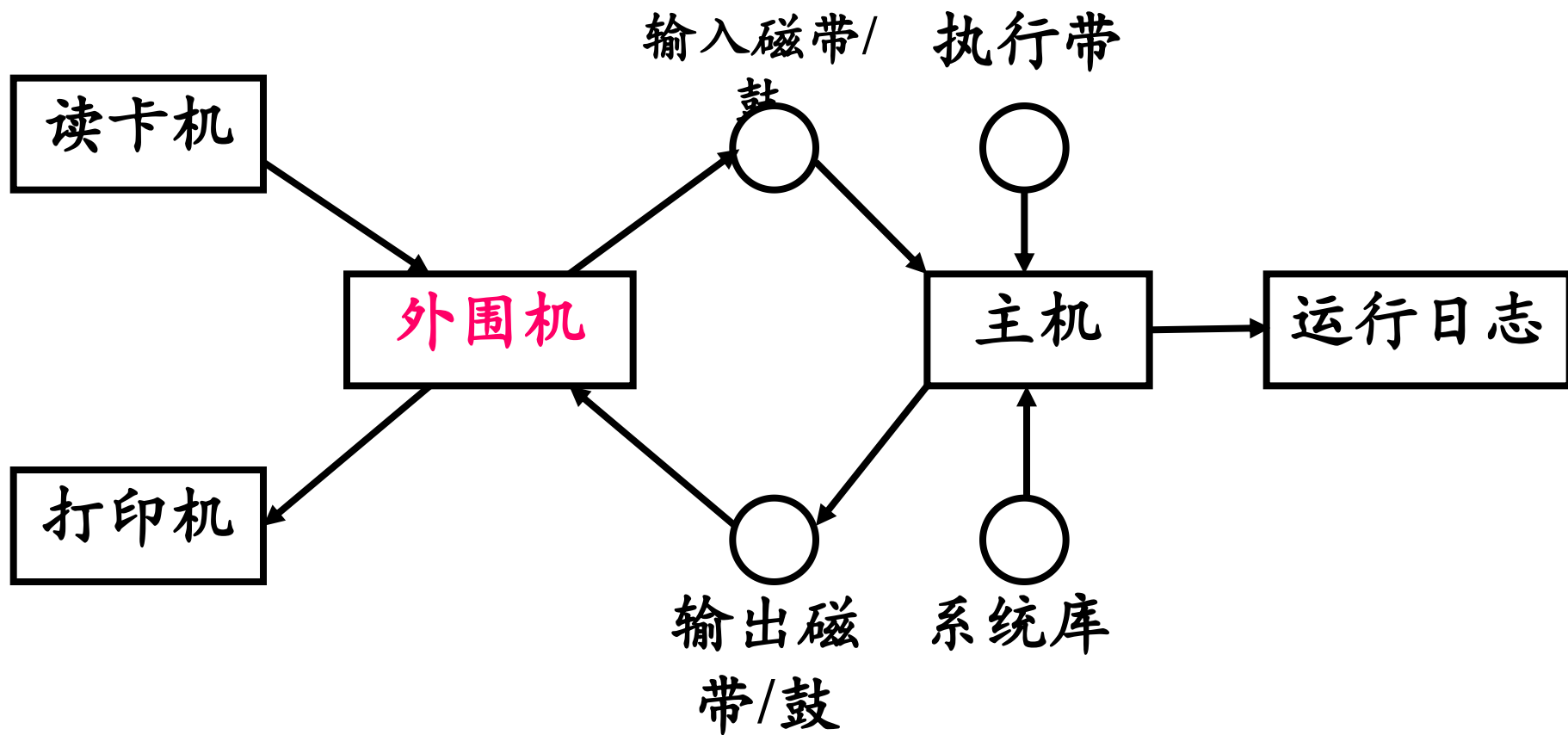
- 用户提交作业：以纸带或卡片为介质；
- 操作员合成批作业：结果为磁带介质；
- 批作业处理：对批作业中的每个作业进行相同的处理：从磁带读入用户作业和编译链接程序，编译链接用户作业，生成可执行程序；启动执行；执行结果输出。

【存在的问题】：

慢速的输入输出处理仍直接在主机的控制下完成。输入输出时，CPU处于等待状态。（CPU与慢速I/O设备之间的矛盾日益突出）

1.2 OS的发展史—单道

2) 脱机批处理



1.2 OS的发展史—单道

2) 脱机批处理

利用外围机完成输入输出功能。主机与外围机可并行工作。

- 外围机：完成面向用户的输入输出（纸带或卡片），中间结果暂存在磁带或磁盘上。
- 作业控制命令由监督程序(monitor)来执行，完成如装入程序、编译、运行等操作。

【优点】：同一批内各作业的自动依次更替，CPU和慢速I/O设备并行工作，提高了吞吐量。

【缺点】：磁带或磁盘需要人工装卸，作业需要人工分类，监督程序易遭到用户程序的破坏（由人工干预才可恢）。Cpu与高速设备仍然串行工作

1.2 OS的发展史—单道

【单道批处理的特征】：

- **自动性**:磁带上一批作业能自动逐个依次运行
- 对I/O为主的作业，CPU空闲；
- **顺序性**:作业顺序进入内存,执行与完成的顺序和调入的顺序完全相同。
- **单道性**:内存中仅有一道程序,只有该程序完成或发生异常时,才调入后继程序进入内存运行

【单道批处理的主要问题】：

一个用户独占全部资源CPU负责计算，也负责传输,CPU与外设速度不匹配。

CPU与I/O串行资源利用率仍然很低。

思考： 1、器件技术高速发展，cpu速度更快（单位时间内可以处理更多作业），内存更大（能够容纳更多的作业）更便宜，主机处理能力显著提升。

2、任意时刻，一道作业只能工作在一个资源上：或者在cpu上进行计算（I/O设备等待），或者在I/O设备上输入输出（CPU等待）。

奇思妙想：能不能一次将多个作业装入内存，错开彼此使用的资源的类型，使得多道作业共享系统的资源，共同推进，从而提高内存、cpu和设备的效率？

多道程序设计的思想！！！！

1.2 OS的发展史—多道

【多道程序设计原理】：在计算机内存中同时存放几道相互独立的程序，它们在管理程序的控制下相互穿插地运行，共享CPU和外设等资源。采用多道程序设计技术的批处理系统称为多道批处理系统。

2) 多道程序设计的实现要解决的问题

- **存储保护与重定位**。在多道程序设计环境中，几道程序共享同一内存，硬件必须提供必要的手段保证各道程序之间不互相侵犯。
- **处理机管理和调度**。多道程序共享同一个处理机，因此存在处理机的调度问题。
- **资源的管理和调度**。多道程序共享资源，同样存在着资源的分配问题。

多道程序设计思想的技术支持： 中断 技术和通道技术

60年代硬件两重大进展： **中断技术**和**通道技术**

1) **中断**：指CPU在收到外部中断信号后，停止原来工作，转去处理该中断事件，完毕后回到原来断点继续工作。

【**中断处理过程**】：中断请求，中断响应，中断点（暂停当前任务并保存现场），中断处理例程，中断返回（恢复中断点的现场并继续原有任务。

中断作用：在外部I/O设备和CPU之间建立起通讯机制

中断技术和通道技术

2) 通道(又称为I/O处理机): 实际上是一台功能单一、结构简单的I/O处理机, 它单独与CPU, 并直接控制外部设备, 与内存进行数据传输。

— 通道有专用的I/O处理器, 可与CPU并行工作。通道具有自己的指令(通道指令), 可编程实现各种复杂的I/O处理。

作用: 与脱机批处理中的外围机功能相似。

通道和中断技术结合, 使得多道程序共享系统资源可以真正实现。

(97年高级程序员试题2) 在有一台处理机CPU和两台输入输出设备I01和I02, 在多道程序环境下, 同时将P₁、P₂、P₃ 这三个作业投入运行。它们使用设备的先后顺序和占用设备时间分别是:

P₁: I02(30ms), CPU(10ms), I01(30ms), CPU(10ms)

P₂: I01(20ms), CPU(20ms), I02(40ms)

P₃: CPU(30ms), I01(20ms)

在控制程序介入时间可以忽略不计的假设下, 作业P₁、P₂、P₃ 从投入到完成所用的时间分别是多少? 三个作业从投入运行到全部完成, CPU、I01和I02的利用率分别是多少?

P_1 : IO2(30ms), CPU(10ms), IO1(30ms), CPU(10ms)

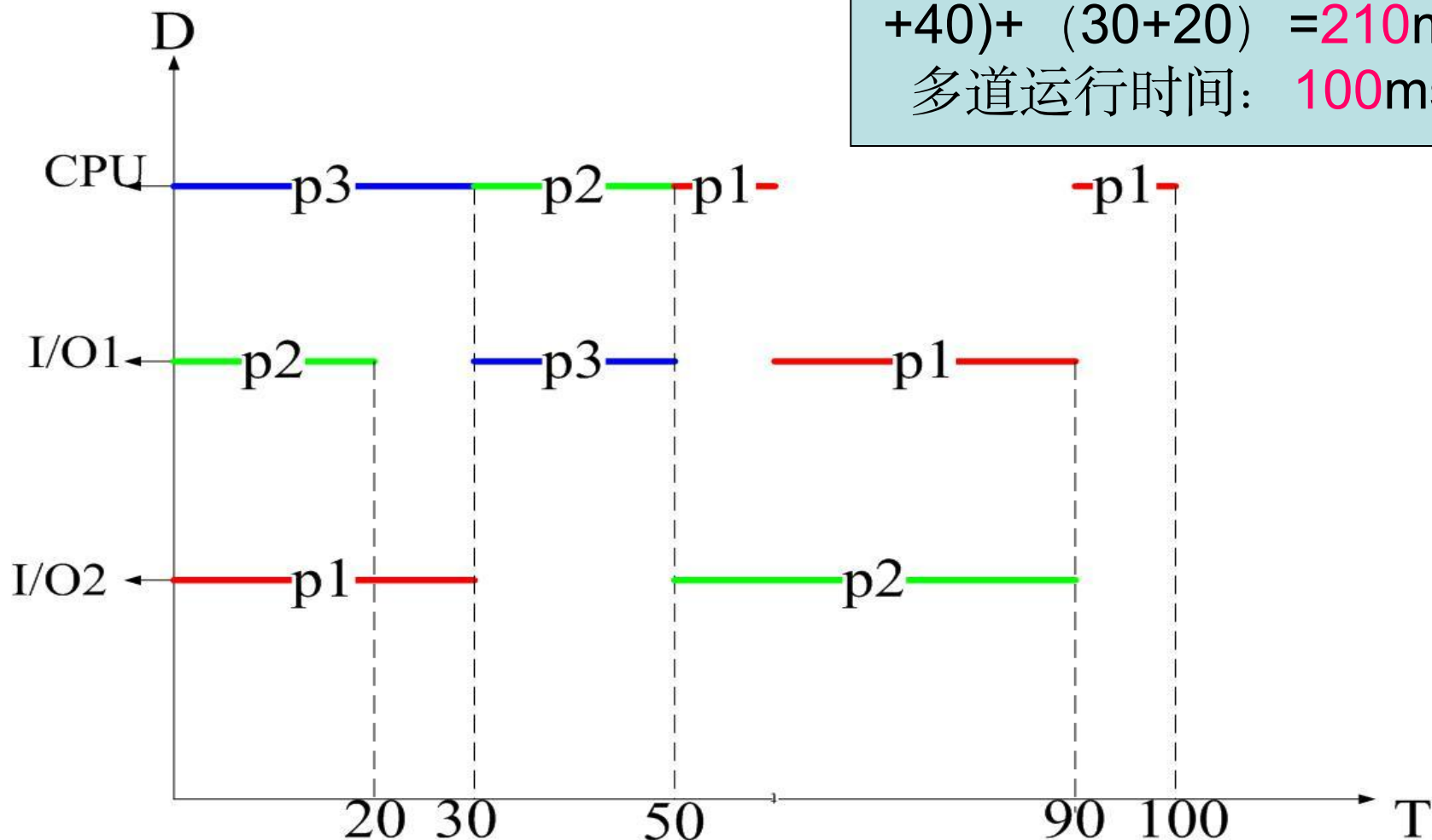
P_2 : IO1(20ms), CPU(20ms), IO2(40ms)

P_3 : CPU(30ms), IO1(20ms)

单道运行时间:

$(30+10+30+10)+(20+20+40)+(30+20) = 210\text{ms}$

多道运行时间: 100ms



1.2 OS的发展史—多道

3) 多道程序设计的特征

多道。同时存在两道或两道以上的程序处于执行的开始点和结束点中间。

宏观并行。都处于运行状态，但都未运行完；

微观串行。各作业交替使用CPU和设备。

4) 多道批处理系统的优缺点：

<优>资源利用率高：CPU、内存、设备利用率较高；

<优>作业吞吐量大：单位时间内完成工作总量大；

<缺>用户交互性差：整个作业完成后或中间出错时，才与用户交互，不利于调试和修改；

<缺>作业平均周转时间长：短作业的**周转时间**（从作业提交到完成的这段时间）显著增长；

单道和多道批处理的比较

	单道	多道
内存使用	每次一个作业	每次多个作业（充分利用内存）
作业次序	顺序，先进先出	无确定次序

1.2 OS的发展史—多道

5) 多道批处理系统的核心技术:

- 作业调度: 作业的现场保存和恢复
- 资源共享: 资源的竞争和同步——互斥机制
- 内存使用: 提高内存使用效率 (为当前由CPU执行的程序提供足够的内存) ——覆盖(overlay), 交换(swap)和虚拟存储(virtual memory)
- 内存保护: 系统存储区和各应用程序存储区不可冲突, 即存储保护
- 文件非顺序存放、随机存取。

出现了: 作业管理、处理机管理、存储管理、设备管理、文件系统管理(file system)

多道批处理系统的出现标志着现代操作系统的产生！！！！！！

操作系统的定义

操作系统是①控制和管理计算机系统的硬件和软件资源，②合理的组织计算机工作流程以及③方便用户使用的程序和数据的集合。

设置操作系统的目的就是提高计算机系统的效率，增强系统的处理能力，充分发挥系统的利用率，方便用户使用。

1.2 OS的发展史—分时

4、分时操作系统 (70年代中期至今, 大规模集成电路)

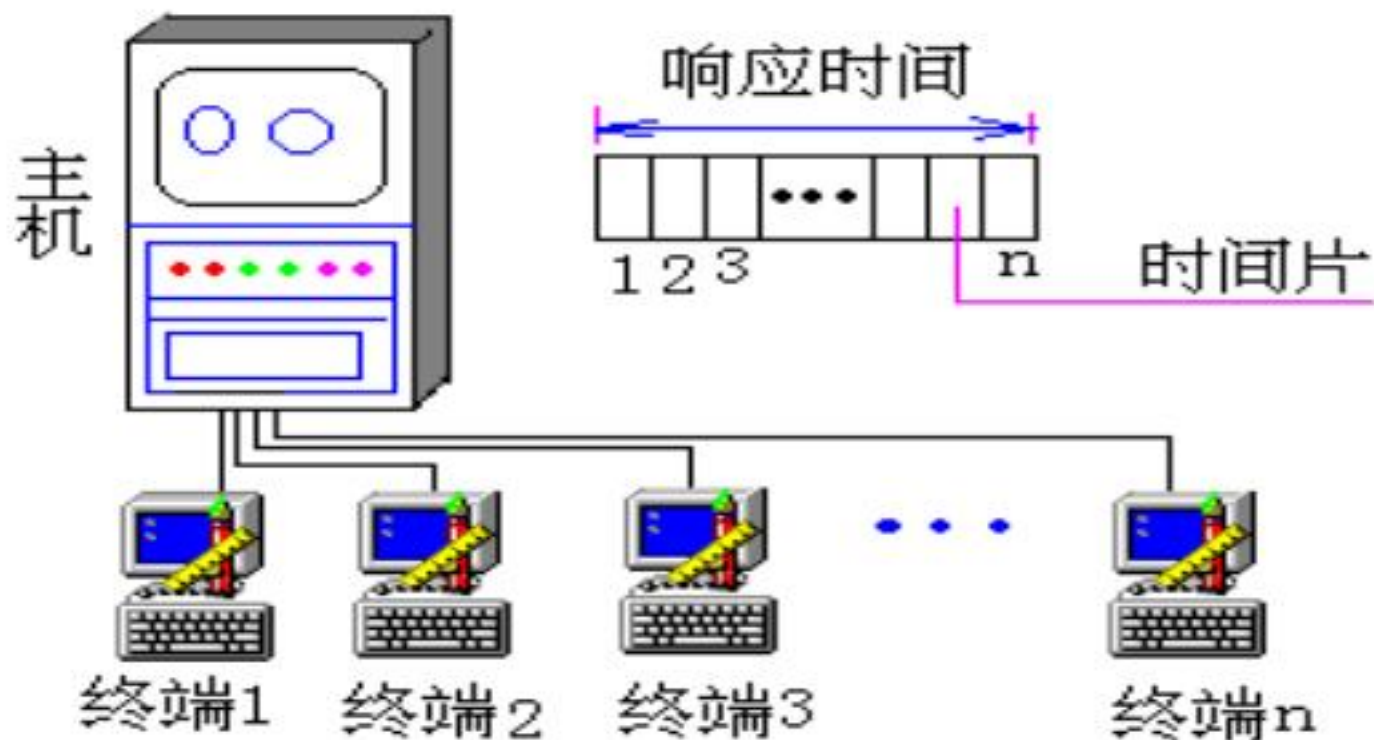
【分时】：两个或两个以上的事件按时间划分轮流的使用计算机系统的某一资源。

【分时系统】：多个用户分时（分时的时间单位称为时间片，通常为几十毫秒）的使用同一计算机的系统称为为分时系统。主要采用时钟和中断技术。

1) 分时系统实现方法：

- 系统配置多路卡及时接受各终端输入命令和数据。
- 及时处理，不允许某作业长期占用处理机，将响应周期分为很短的时间片，在该周期内每个作业轮流执行一次(占一个时间片)。每个作业可以及时接收用户的命令和数据，并能及时得到处理。

分时系统的结构示意图



1.2 OS的发展史—分时

2) 分时系统的特点:

- 同时性或多路性: 多用户同时操作、使用计算机
- 独占性: 各终端用户感觉自己独占了计算机;
- 及时性: 用户的请求能在较短时间内相应;
- 交互性: 用户能计算机进行人——机对话。

3) 调进和调出 (Roll-in / Roll-out)

调进/调出是实现分时系统的一种主要方式, 包括:

- 主存—后援存储器方式: 简单分时系统所采用, 特点内存只存放一个当前执行作业, 其他外存上。
- 多流调入调出方式: 基于多道程序技术, 特点主存中同时存在一个当前执行作业和多个后备作业。

1.2 OS的发展史—分时

4) 分时系统的响应时间

响应时间为用户发出一条指令到系统处理完这条指令并做出回答所需要的时间。是衡量分时系统性能好坏的一条重要标志，但具体的响应时间与系统的用户个数及时间片大小有关。响应时间 T 与时间片 q 和用户个数 n 之间的关系为 $T=nq$ 。

【注意】：分时系统在轮转处理多个终端用户的请求时，在调入/调出上增加了许多时间消耗，其中主要是主存与后援存储器之间的信息交换时间。因此为改善的响应时间需要一方面加快主存与后援存储器之间的对换速度，另一方面减少对换的信息量。

现在的许多操作系统都具有分时处理的功能，在分时系统的基础上，操作系统的发展开始分化，如实时系统、通用系统、个人系统等。

1.2 OS的发展史—实时

- 5、实时操作系统

实时操作系统主要用于过程控制、事务处理等有实时要求的领域，其主要特征是实时性（在规定的时间内完成处理）和可靠性。

1) 实时系统的分类：

- 实时控制：要求与被控制的变化速度相比，其反应速度足够快；工作安全可；需要人工干预时，操作简便。如生产过程控制，宇航自动控制等。
- 实时信息处理系统：要求计算机能够在容许的延迟时间内，相应外部的事件请求，完成对该事件的处理，并控制所有的实时设备和实时任务协调运行。如飞机订票系统，期货、股票交易系统。

1.2 OS的发展史—实时

2) 实时系统的特征:

实时系统具有专用性、种类多、用途各异等特点，其一个基本特征是**事件驱动**，即接收到某些外部信息后，由系统选择某一程序去执行，完成相应的实时任务。

- **实时时钟管理**。向系统提供日期、时刻以及定时任务（每间隔一固定时间循环执行）和延迟任务（推迟一段时间后执行）进行控制的信号。
- **中断管理**。实时系统必须有效和高速的完成各种中断处理。
- **系统容错管理**。高可靠性一直是实时系统的设计目标。

1.2 OS的发展史—实时

- **多重任务管理**。实时系统的物理过程并发性的特点，因此实时系统应具备多重任务处理的能力，且强调任务调度与转换的高效性。
- **系统生成能力**。实时系统一般大多是专用性的系统，其种类与规模相差较大，因此，要求实时操作系统应具有较强的生成能力，以便根据需要裁剪或补贴系统功能，使之与应用系统相适应。

批处理系统、分时系统、实时系统是现代操作系统的三种基本类型。

目前的操作系统，通常具有分时、实时和批处理功能，又称作通用操作系统。可适用于计算、事务处理等多种领域，能运行在多种硬件平台上，如 UNIX系统、Windows NT 等。——通用化、小型化

1.2 OS的发展史—多种其他操作系统

- 网络操作系统
- 分布式操作系统
- 多处理机操作系统
- PC操作系统
- 。 。 。 。 。 。 。 。

1.3 OS的功能和主要特征

- 1、操作系统的功能

操作系统的主要任务是对系统中的软件、硬件实施有效的管理，以提高系统资源（主要包括处理机、主存储器和外存储器、文件系统）的利用率。主要功能如下：

- 1) 处理机管理：
- 2) 存储管理
- 3) 设备管理
- 4) 信息管理
- 5) 用户接口

1.3 OS的功能和主要特征

- 2、现代操作系统的主要特征

设置操作系统的目的就是提高计算机系统的效率，增强系统的处理能力，充分发挥系统的利用率，方便用户使用。因此现代操作系统普遍采用以多道程序设计为基础的并行操作技术。主要特征如下：

- 1) 并发性(concurrency)
- 2) 共享性(sharing)
- 3) 虚拟性(virtual)
- 4) 异步性(asynchronism)

1.3 OS的功能和主要特征

并发(concurrency)：多个事件在同一时间段内发生。
并行(parallel)：多个事件在同一时刻发生。

1) 并发性(concurrency)

操作系统是一个并发系统，各进程间的并发，系统与应用间的并发。操作系统要完成这些并发过程的管理。在多道程序处理时，宏观上并发，微观上交替执行（在单处理器情况下）。

— 程序的静态实体是可执行文件，而动态实体是进程（或称作任务），并发指的是进程。

1.3 OS的功能和主要特征

2) 共享性(sharing)

多个进程共享有限的计算机系统资源。操作系统要对系统资源进行合理分配和使用。资源在一个时间段内交替被多个进程所用。

- 互斥共享方式（如音频设备），资源分配后到释放前，不能被其他进程所用。
- 同时访问方式，（如可重入代码，磁盘文件）。
- 资源分配难以达到最优化

1.3 OS的功能和主要特征

3) 虚拟性(virtual)

一个物理实体映射为若干个对应的逻辑实体（分时或分空间）。虚拟是操作系统管理系统资源的重要手段，可提高资源利用率。

- CPU——每个用户（进程）的“虚处理机”。
- 存储器——每个进程都占有的地址空间（指令+数据+堆栈）。
- 显示设备——多窗口或虚拟终端

如虚拟光驱，虚拟机（VMWare， Virtual PC等）

1.3 OS的功能和主要特征

4) 异步性(asynchronism)

异步性也称不确定性，指进程的执行顺序和执行时间及执行结果的不确定性：

- 程序执行结果不确定，不可再现。相同输入与环境下多次运行结果不同。
- 多道程序设计环境下，程序按异步方式运行。多个进程并发执行，“时走时停”，不可预知每个进程的运行推进快慢，引发执行顺序与时间的不确定。

1.4 操作系统的结构

现代操作系统从内部结构分，通常包含两部分：一是内核；二是核外部分，其通常是一些实用程序。

- 1、操作系统的内核

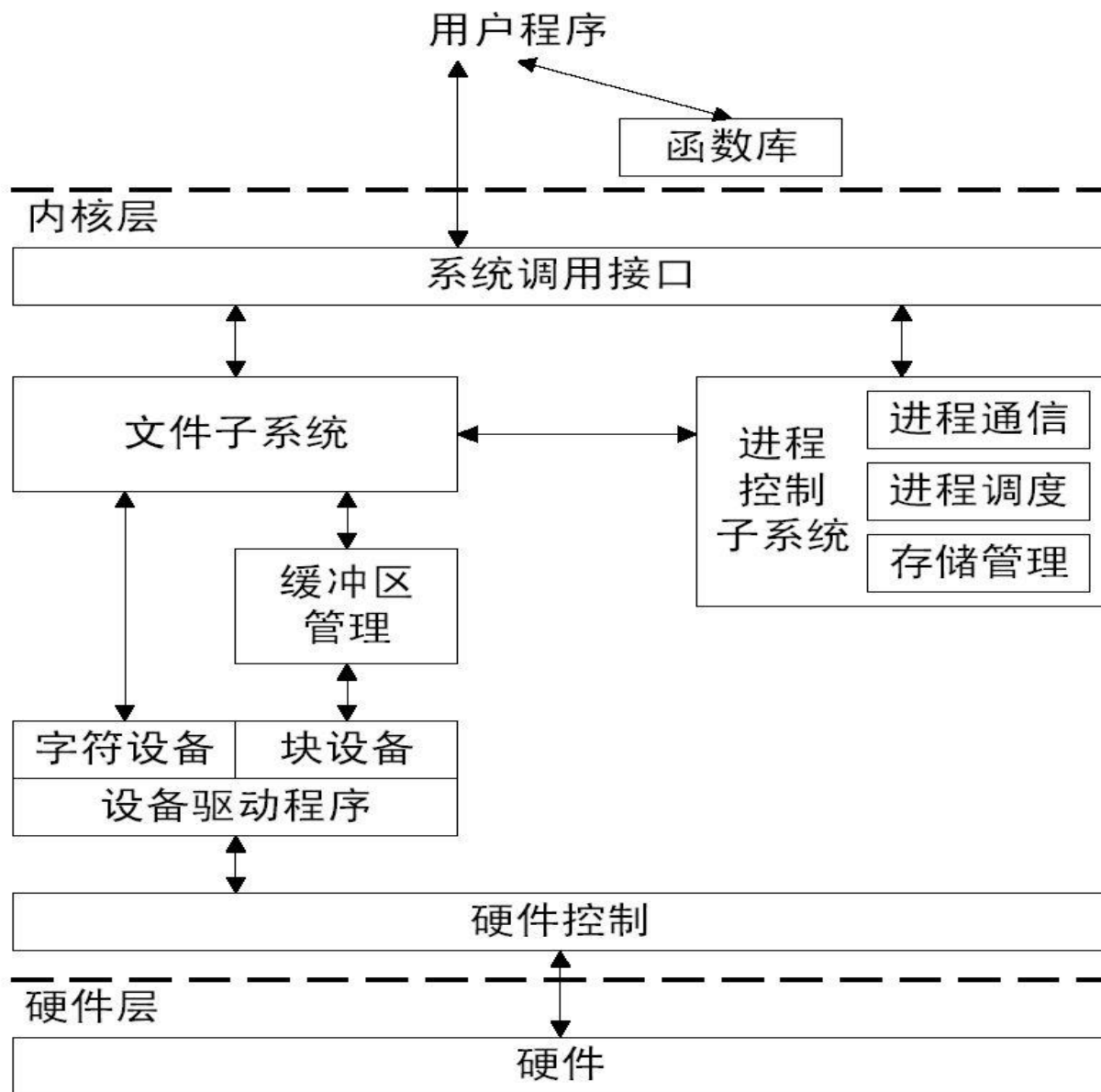
- 1) 内核在操作系统中的地位：

OS内核是对硬件的首次扩充，是实现操作系统各项功能的基础。(UNIX为例)。

传统操作系统内核包括以下功能模块：

- 进程、线程及其管理
- 存储管理。
- I/O管理
- 文件系统。

UNIX的内核结构



1.3 OS的功能和主要特征

2) 强内核与微内核:

操作系统有两种内核组织形式:

- **强内核(Monolithic Kernel)** : 通过陷入内核实现系统调用, 即在内核完成所需要的服务后将结果返回给用户程序, 如UNIX采用的这种结构。
- **微内核(Micro Kernel)** : 微内核的基本思想是良好的结构化、模块化、最小的公共服务。

微内核只提供最基本、最必要的服务: 供进程间通讯、有些存储管理、有限的低级进程管理和调度、低级I/O四种服务。

微内核的特点

【优点】：

- 良好的**扩充性**：只需添加支持新功能的服务进程即可。
- **可靠性**好：调用关系明确，执行转移不易混乱
- 便于**网络服务**，实现分布式处理：以同样的调用形式，在下层可通过核心中的网络传送到远方服务器上(远地过程调用 RPC, Remote Procedure Call)。

【缺点】：

消息传递比直接调用**效率**要低一些（但可以通过提高硬件性能来补偿）

1.4 操作系统的结构

- 2、操作系统的结构设计

【OS结构设计模式】：是将操作系统所提供的特性、服务以及系统所执行的任务统一成一体的一個概括性的框架。

随着操作系统的发展，功能越强，OS自身代码量越大。操作系统采用良好的结构：有利于保证正确性以及自身修改和扩充。

- 1) 整体或模块结构
- 2) 分层结构或虚拟机
- 3) 客户/服务器模型或微内核结构
- 4) 面向对象技术

操作系统的设计原则

- 可维护性：容易修改与否称为可维护性：
 - 改错性维护：改正已发现的错误；
 - 适应性维护：修改软件，使之适应新的运行环境（软、硬件环境）；如：操作系统的移植。
 - 完善性维护：增加新功能；
- 可靠性：可靠性包括两方面：
 - 正确性：正确实现所要求的功能和性能；
 - 稳健性：对意外（故障和误操作）作出适当的处理；
- 可理解性：易于理解，以方便测试、维护和交流；
- 性能：有效地使用系统资源；尽可能快地响应用户请求；

1.4 操作系统的结构

1) 整体或模块结构：

整个系统按功能进行设计和模块划分。系统是一个单一的、庞大的软件系统。这种结构思想来源于服务功能观点，而不是资源管理的观点。

【模块结构的特点】：

模块由众多服务过程（模块接口）组成，可以随意调用其他模块中的服务过程。

- 优点：具有一定灵活性，在运行中的高效率
- 缺点：功能划分和模块接口难保正确和合理；模块之间的依赖关系（功能调用关系）复杂（调用深度和方向），降低了模块之间的相对独立性，不利于修改。

1.4 操作系统的结构

2) 分层结构或虚拟机:

从资源管理观点出发, 划分层次。在某一层次上代码只能调用低层次上的代码, 使模块间的调用变为有序性。系统每加一层, 就构成一个比原来功能更强的虚拟机。有利于系统的维护性和可靠性。

【优点】:

- 功能明确, 调用关系清晰 (高层对低层单向依赖), 有利于保证设计和实现的正确性。
- 低层和高层可分别实现 (便于扩充); 高层错误不会影响到低层; 避免递归调用。

【缺点】: 降低了运行效率

分层原则

- 被调用功能在低层：如文件系统管理—>设备管理—>设备驱动程序
- 活跃功能在低层：提高运行效率
- 资源管理的公用模块放在最低层：如缓冲区队列、堆栈操作
- 存储器管理放在次低层：便于利用虚拟存储功能
- 最低层的硬件抽象层：与机器特点紧密相关的软件放在最低层。
- 资源分配策略放在最外层，便于修改或适应不同环境

各系统对具体划分多少层次有不同的看法。

1.4 操作系统的结构

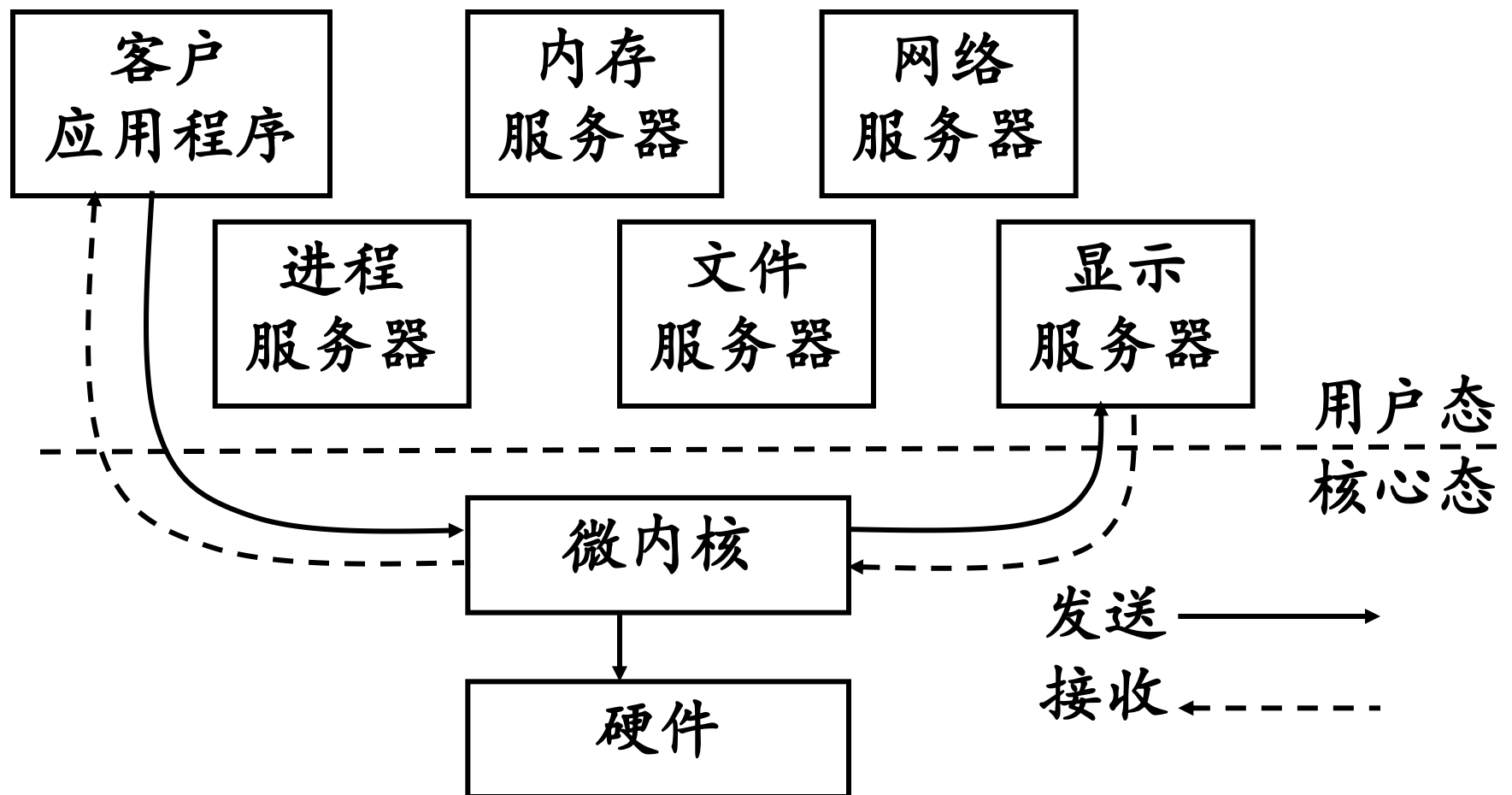
3) 客户/服务器模型或微内核结构：

把操作系统分成若干分别完成一组特定功能的**服务进程（服务器）**，等待客户提出请求；而系统内核**只实现操作系统的基本功能**(如：虚拟存储、消息传递)。

服务器运行于用户态，并循环检测是否有客户请求服务。**客户**可以是一个**应用程序**或者是**另一操作系统成分**。

用户——内核——服务器通过消息联系，如图所示。

客户/服务器模式下的操作系统模型



1.4 操作系统的结构

【客户/服务器模型优点】：

- 简化了基本操作服务，缩小了内核。提供了多种API且易于维护。
- 提高了可靠性。服务程序在用户态，且独立运行，不能直接访问硬件。
- 适合分布式计算环境。采用客户/服务器为基础，且使用消息传递进行通。

1.4 操作系统的结构

4) 面向对象技术:

对象是指具有相同属性、服从相同规则的一类事物的抽象, 其中的具体事物称为对象的实例; 将数据结构和定义在其上的一组操作封装起来表示某个对象。这样数据结构中的数据对外都是隐蔽的, 必须通过封装其内的操作(函数)对它们访问。

【优点】:

- 可修改性和可扩充性
- 继承性
- 正确性和可靠性

数据结构
函数1
函数2
函数3