计算机操作系统

第一章 操作系统引论

1.1 操作系统的目标和作用

1.1.1 操作系统的目标

目前存在着多种类型的OS，不同类型的OS，其目标各有所侧重。通常在计算机硬件上配置的OS，其目标有以下几点：囍

- 方便性 ---方便用户
- 有效性 ---系统管理效率
- 扩展性 ---体系结构：软硬件结构发展
- 开放性 ---体系结构：软硬件结构兼容性

权衡

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来使用计算机系统。

- OS是一个系统软件，因而这种接口是软件接口。

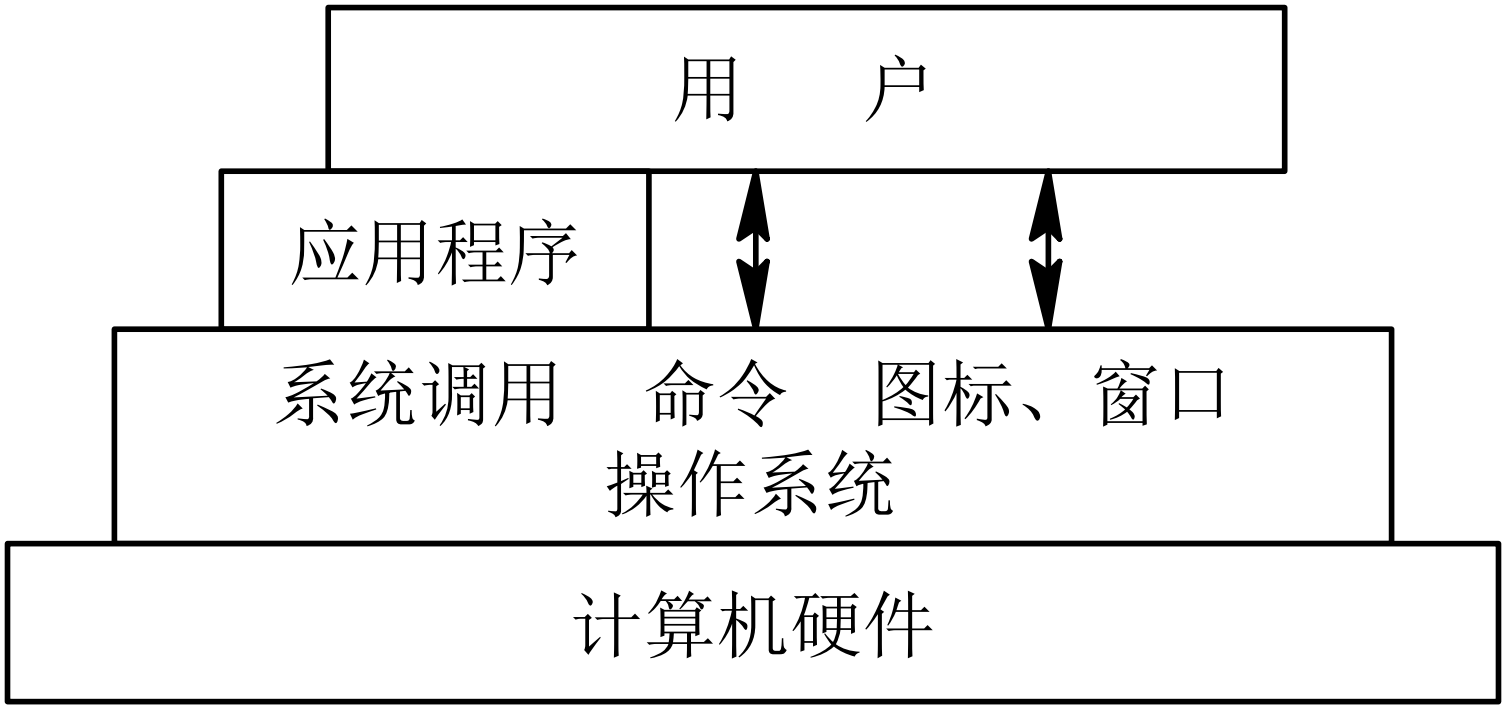
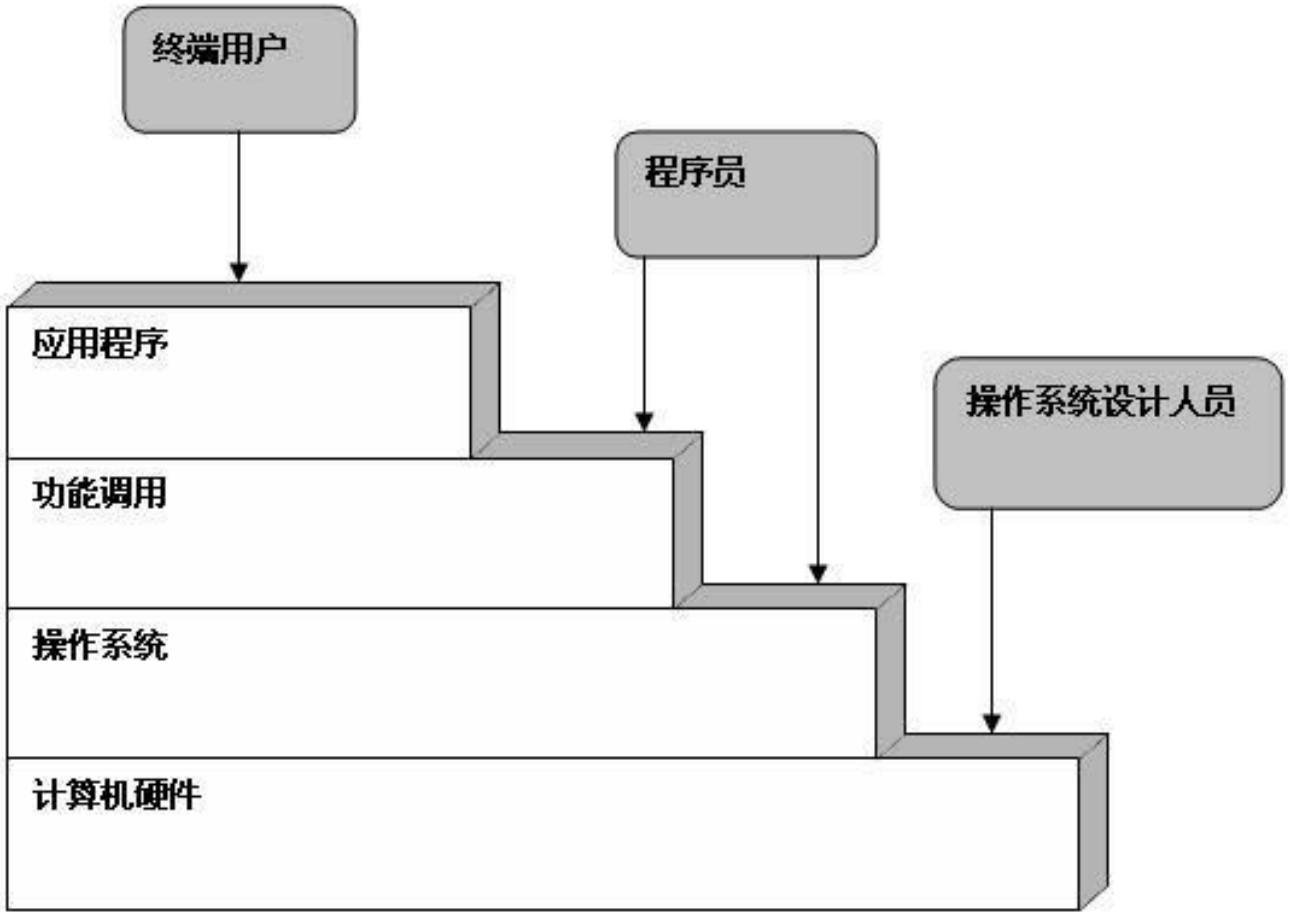


图 1-1 OS作为接口的示意图



计算机系统组成部分的层次图

人机接口接口的不同使用方式：

(1) 命令方式。这是指由OS提供了一组联机命令(语言)，用户可通过键盘输入有关命令，来直接操纵计算机系统。囍

(2) 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，来操纵计算机。囍

(3) 图形、窗口方式。用户通过屏幕上的窗口和图标来操纵计算机系统和运行自己的程序。

OS作用 2. OS作为计算机系统资源的管理者囍

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)。

OS的主要功能是针对这四类资源进行有效的管理，即：

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。

3. OS用作扩充机器㠯

以OS软件扩充硬件性能，得到功能更强大的系统。

通常把覆盖了软件的机器称为扩充机器或虚机器。

1.1.3 推动操作系统发展的主要动力

1. 不断提高计算机资源利用率

2. 方便用户甛

3. 器件的不断更新换代

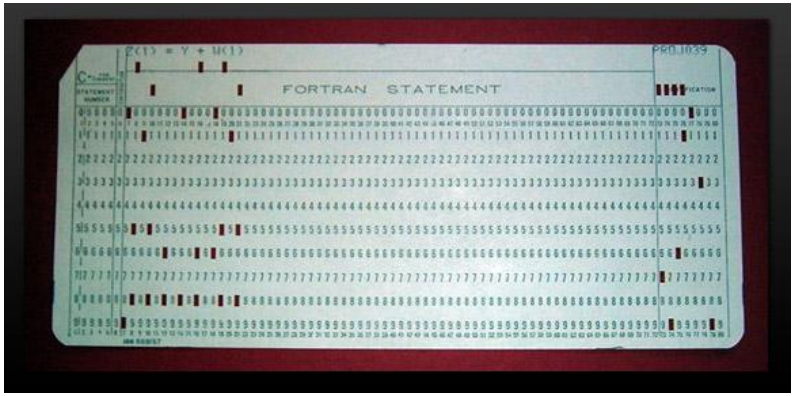
4. 计算机体系结构的不断发展

OS发展历史背景

- First generation 1945 - 1955
  - vacuum tubes电子管
- Second generation 1955 - 1965
  - transistors, batch systems 晶体管时代/批处理
- Third generation 1965 – 1980
  - ICs and multiprogramming 集成电路/多道程序设计
- Fourth generation 1980 – present
  - personal computers 个人计算机时代

1.2 操作系统的发展过程

1.2.1 无操作系统的计算机系统

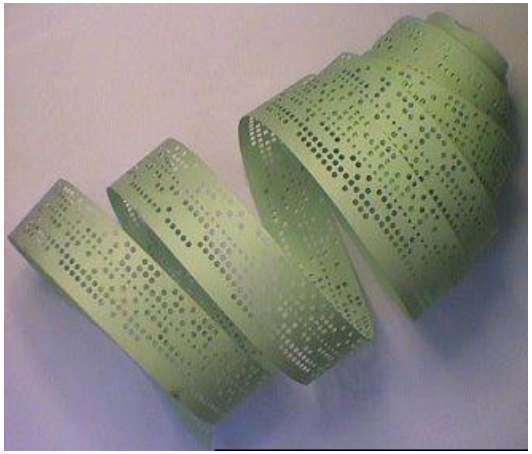


1. 人工操作方式

- 1945年到50年代中期，第一代计算机，尚未出现OS。
- 使用 穿孔纸带(或穿孔卡片) 。
- 人工操作方式有以下两方面的缺点：

(1) 用户独占全机。

(2) CPU等待人工操作。



2. 脱机输入/输出(Off-Line I/O)方式菖

这种脱机I/O方式的主要优点如下：菖

((1))减少了CPU的空闲时间。

(2) 提高I/O速度。

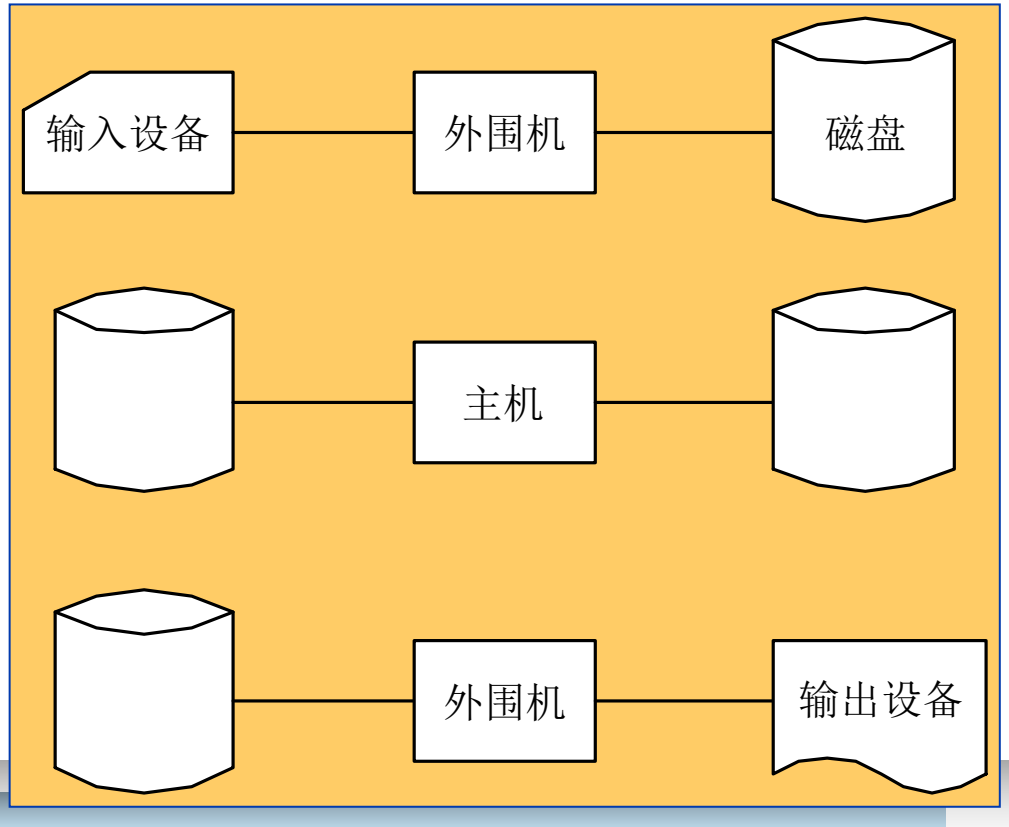
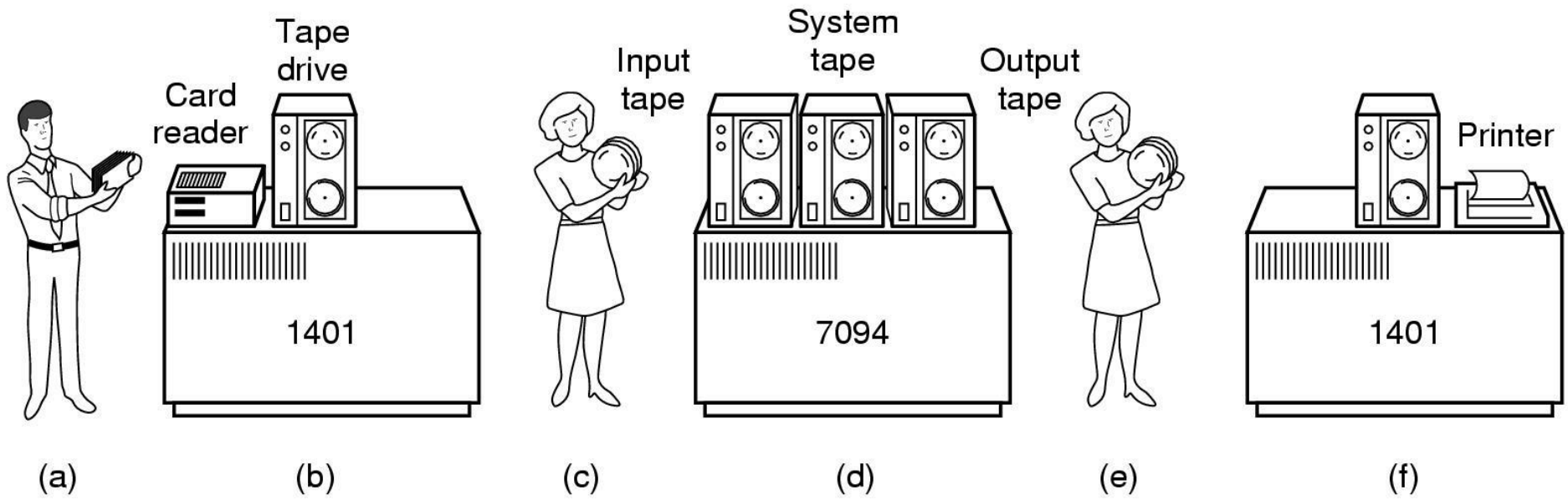


图 1-2 脱机I/O示意图

早期批处理系统的例子



Early batch system

1. bring cards to 1401
2. read cards to tape
3. put tape on 7094 which does computing
4. put tape on 1401 which prints output

1.2.2 单道批处理系统

1. 单道批处理系统(Simple Batch Processing System)的处理过程

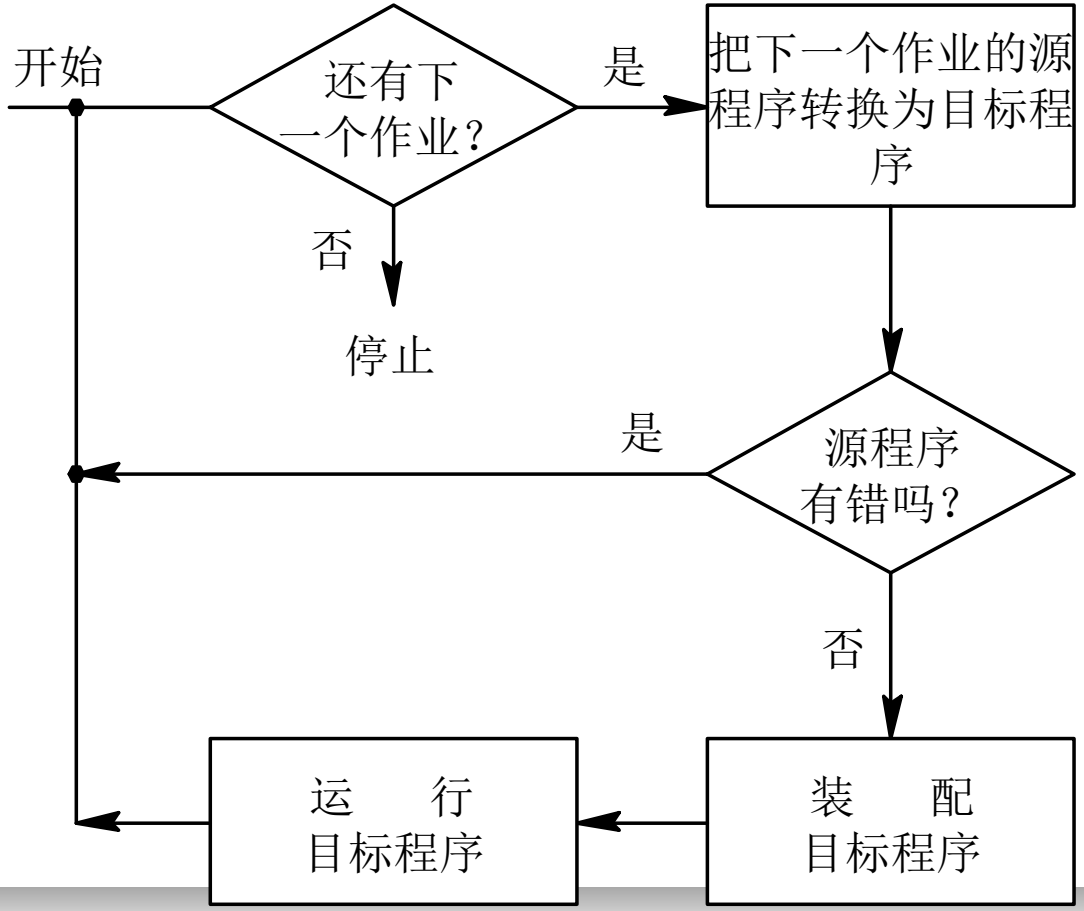


图 1-3 单道批处理系统的处理流程

2. 单道批处理系统的特征囍

•是最早出现的一种OS，只能算作是OS的前身。

•最大进步在于替代了人工操作。

• 该系统的主要特征如下：囍

(1) 自动性。

(2) 顺序性。 囍

(3) 单道性。

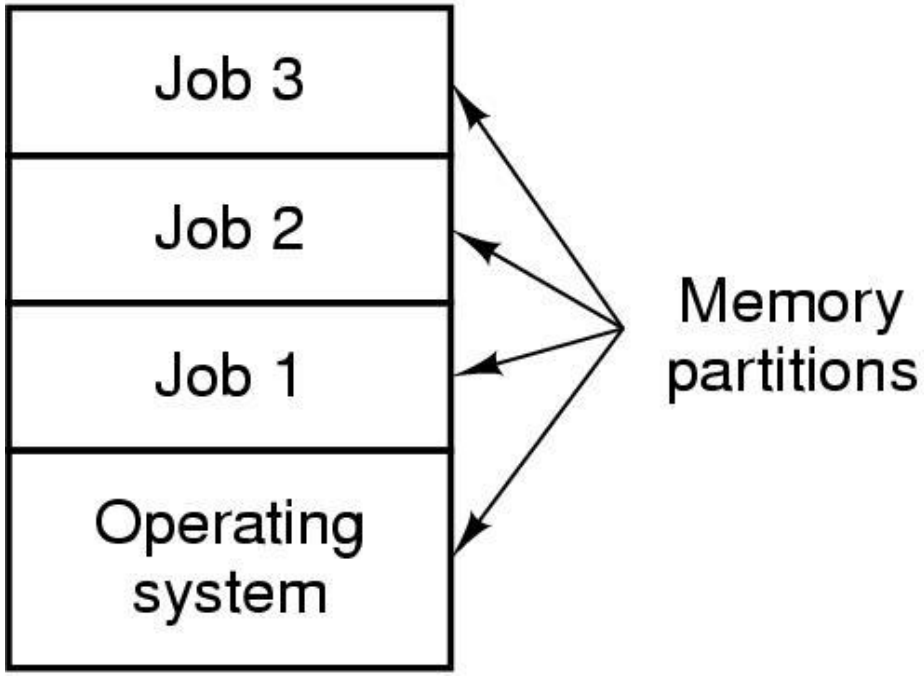
1.2.3 多道批处理系统

1. 多道程序设计的基本概念

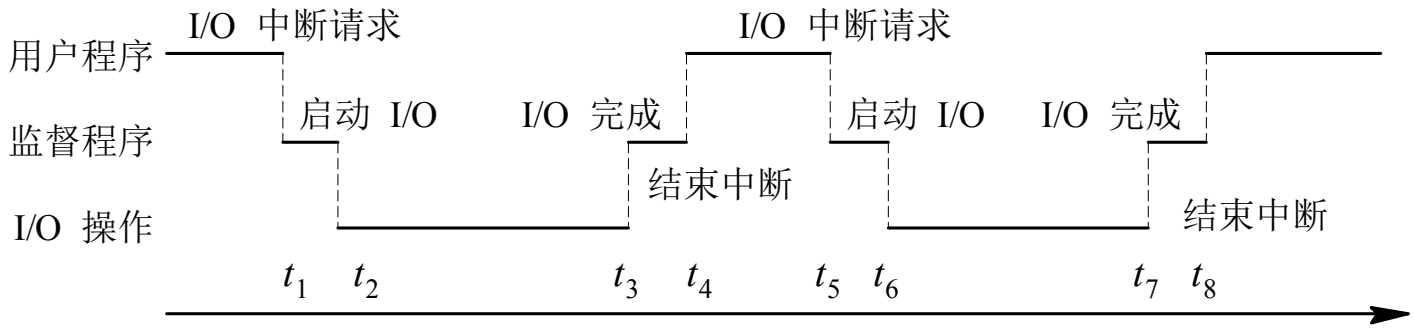
为提高资源的利用率和系统吞吐量，在60年代中期引入了多道程序设计技术，形成了多道批处理系统。

(Multiprogrammed Batch Processing System)。

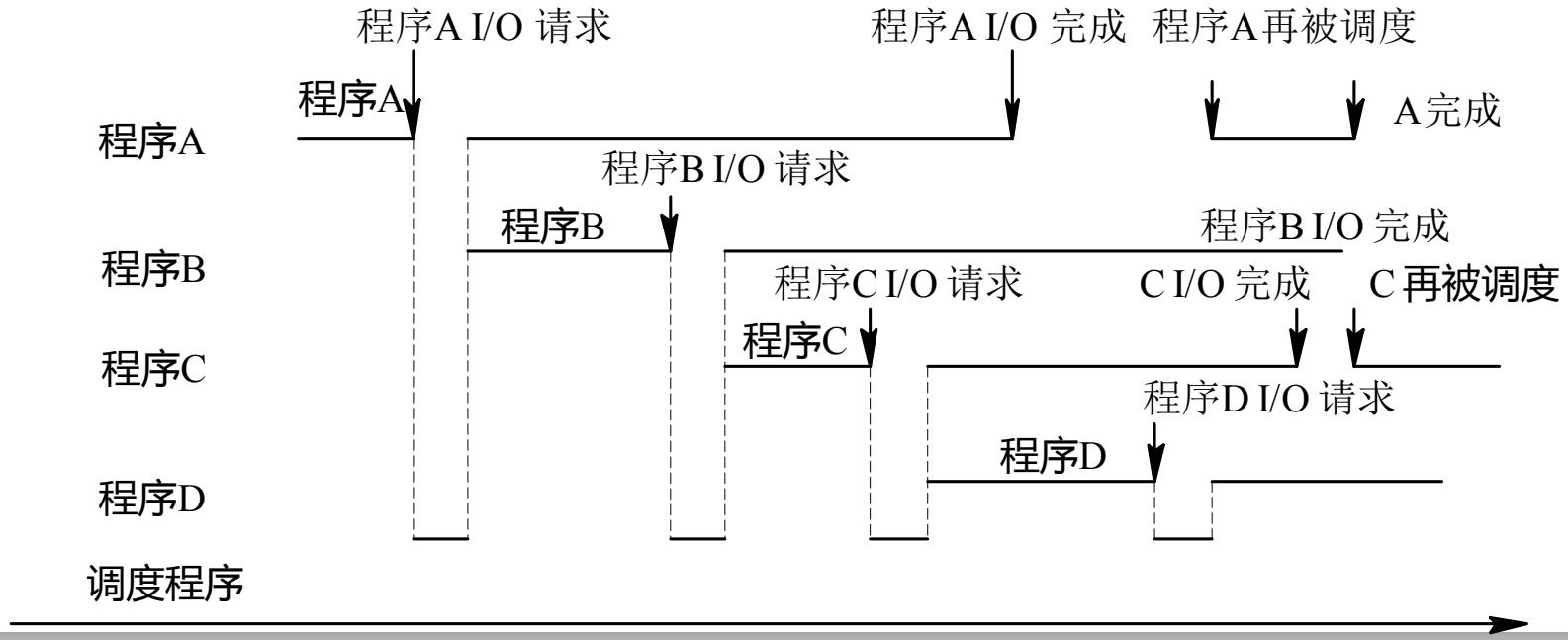
- 后备队列
- 作业调度程序
- 作业调度 算法



- Multiprogramming system 示意图
  - three jobs in memory



(*a*) 单道程序运行情况



(*b*) 四道程序运行情况

图 1-4 单道和多道程序运行情况

（1）提高CPU的利用率。

图 1-4(*a*)示出了单道程序的运行情况，从图可以看出：在t_2~t_3、 t_6~t_7时间间隔内CPU空闲。

在引入多道程序设计技术后， 程序交替运行， 保持了CPU处于忙碌状态。

(2) 可提高内存和I/O设备利用率。

在内存中装入多道程序， 并允许它们并发执行，则大大提高内存和I/O设备的利用率。囍

(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，从而降低作业加工所需的费用。

2. 多道批处理系统的特征

((1))多道性。

(2) 无序性。

(3) 调度性。

3. 多道批处理系统的优缺点

((1))资源利用率高。

(2) 系统吞吐量大。

(3) 平均周转时间长。

(4) 无交互能力。

4. 多道批处理系统需要解决的问题

((1))处理机管理问题。

(2) 内存管理问题。

(3) I/O设备管理问题。

(4) 文件管理问题。

(5) 作业管理问题。

1.2 操作系统的发展过程

1.2.4 分时系统

1. 分时系统(Time-Sharing System)的产生

多道程序系统、批处理系统提供了提高资源（CPU,内存，设备）利用率和系统吞吐量的环境，但是没有为用户提供**与计算机直接交互**的能力。

推动分时系统形成和发展的主要动力，是用户的需求。用户的需求具体表现在以下几个方面：

(1) 人—机交互。

(2) 共享主机。

(3) 便于用户上机。

1.2 操作系统的发展过程

2. 分时系统实现中的关键问题

(1) 及时接收。

(2) 及时处理。

分时系统是多道程序设计的自然延伸，虽然CPU还是通过在多个作业之间的切换来执行多个作业，但是由于切换的速度非常快，用户可以在每个程序运行期间与之进行交互。

- **时间片**

1.2 操作系统的发展过程

3. 分时系统的特征

((1))多路性。

(2) 独立性。

(3) 及时性。

(4) 交互性。

第一个分时操作系统CTSS

- 分时系统的思想——1959年在MIT提出
- 每个用户有一个联机终端
- 调试程序的用户常常只发出简短的命令
  很少有长的费时命令
- 计算机能够为许多用户提供交互式、快速服务
  同时在CPU空闲时还能在后台运行大作业
- 第一个分时系统（CTSS）由 MIT的Fernando Corbato 等1961年在一改装的IBM 7090/94机上开发成功（有32个交互式用户）
- 第一个有虚拟存储器(virtual memory)和页面调度(paging)的机器

1.2 操作系统的发展过程

1.2.5 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

((1))实时控制。

(2) 实时信息处理。

- 当对处理器的操作或数据传输有着严格地时间要求时——需要使用实时系统。
- 常用来控制特定设备，或者特定数据传输控制。

1.2 操作系统的发展过程

2. 实时任务

1) 按任务执行时是否呈现周期性来划分菖

((1))周期性实时任务。

(2) 非周期性实时任务。

截止时间(Deadline)。分为：

① 开始截止时间——任务在某时间以前必须开始执行；

② 完成截止时间——任务在某时间以前必须完成。

1.2 操作系统的发展过程

2) 根据对截止时间的要求来划分囍

(1) 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。

(2) 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。

3. 实时系统与分时系统特征的比较

((1))多路性。

(2) 独立性。

(3) 及时性。

(4) 交互性。

(5) 可靠性。

1.3 操作系统的基本特性

1. 并发特征（Concurrence）
   - 并发与并行
2. 共享特征（Sharing）
   - 互斥共享
   - 同时访问
3. 虚拟特征（Virtual）
4. 异步性（Asynchronism）

1.3.1 并发(Concurrence)

并行性和并发性：

- **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。
  - **宏观与微观**
- 为使多个程序能并发执行，必须分别为每个程序建立进程。
  - 线程 – 进一步提高并发程度

1.3.2 共享(Sharing)囍

在操作系统环境下，所谓共享是指系统中的资源可供内存中多个并发执行的进程(线程)共同使用。

•共享因资源属性不同而采用不同方式:

1. 互斥共享方式

-- 资源分配后到释放前，不能被其他进程所用。

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。

2. 同时访问方式

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。

囂

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。

- 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；
- 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(Virtual)囍

• 所谓“虚拟”，是指**通过某种技术把一个物理实体变为若干个逻辑上的对应物。**

在OS中利用了多种虚拟技术，分别用来实现虚拟处理机、虚拟内存、 虚拟外部设备和虚拟信道等。囍

1.3.4 异步性(Asynchronism)

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。 进程之间由于竞争或合作，**以不可预知的速度向前推进。(不确定性)**

1.4 操作系统的主要功能

1.4.1 处理机管理功能

1. 进程控制菖

进程控制的主要功能是为作业**创建**进程、**撤消**已结束的进程，以及控制进程在运行过程中的**状态转换**。

在现代OS中，进程控制还应具有为一个进程创建若干个线程的功能和撤消(终止)已完成任务的线程的功能。

2. 进程同步

•进程同步的主要任务是为多个进程(含线程)的运行进行协调。

•两种协调方式：

① **进程互斥方式**， 这是指诸进程(线程)在对临界资源进行访问时， 应采用互斥方式；

② **进程同步方式**，指在相互合作去完成共同任务的诸进程(线程)间，由同步机构对它们的执行次序加以协调。

为了实现进程同步，系统中必须设置进程**同步机制**。

例如：**锁、信号量**

3. 进程通信囍

常见于多个合作进程(线程)相互交换信息。

例如，有三个相互合作的进程， 它们是输入进程、计算进程和打印进程。囍

当相互合作的进程(线程)处于同一计算机系统时，通常在它们之前是采用**直接通信**方式，即由源进程利用发送命令直接将消息(message)挂到目标进程的消息队列上，以后由目标进程利用接收命令从其消息队列中取出消息。

4. 调度囍

作业调度:从后备队列中按照一定的**算法**，选择出若干个作业，为它们分配其必需的资源(首先是分配内存)。

- 当代操作系统基本不需要作业调度。

进程调度:是从进程的就绪队列中选出一新进程，把处理机分配给它，并为它设置运行现场， 使进程投入执行。

•在多线程OS中，通常是把线程作为独立运行和分配处理机的基本单位.

1.4.2 存储器管理功能

1. 内存分配

静态分配方式：每个作业的内存空间是在作业装入时确定的；在作业装入后的整个运行期间，不允许该作业再申请新的内存空间，也不允许作业在内存中“移动”；

动态分配方式：每个作业所要求的基本内存空间，也是在装入时确定的，但允许作业在运行过程中，继续申请新的附加内存空间，以适应程序和数据的动态增涨，也允许作业在内存中“移动”。

内存分配的机制中应具有的结构和功能：

① 内存分配数据结构；如段表、页表

② 内存分配功能；分配策略、算法

③ 内存回收功能。囍

2. 内存保护囍

任务:是确保每道用户程序都只在自己的内存空间内运行，彼此互不干扰。

为了确保每道程序都只在自己的内存区中运行，必须设置内存保护机制。

一种比较简单的内存保护机制，是设置两个**界限寄存器**，分别用于存放正在执行程序的上界和下界。**越界检查。**

3. 地址映射菖

将地址空间中的**逻辑地址**转换为内存空间中与之对应的**物理地址**。

逻辑地址空间 → 物理地址空间：

源程序（编译）

→目标程序（链接）

→可装入程序【逻辑地址】

→装入到内存【物理地址】

4. 内存扩充囍

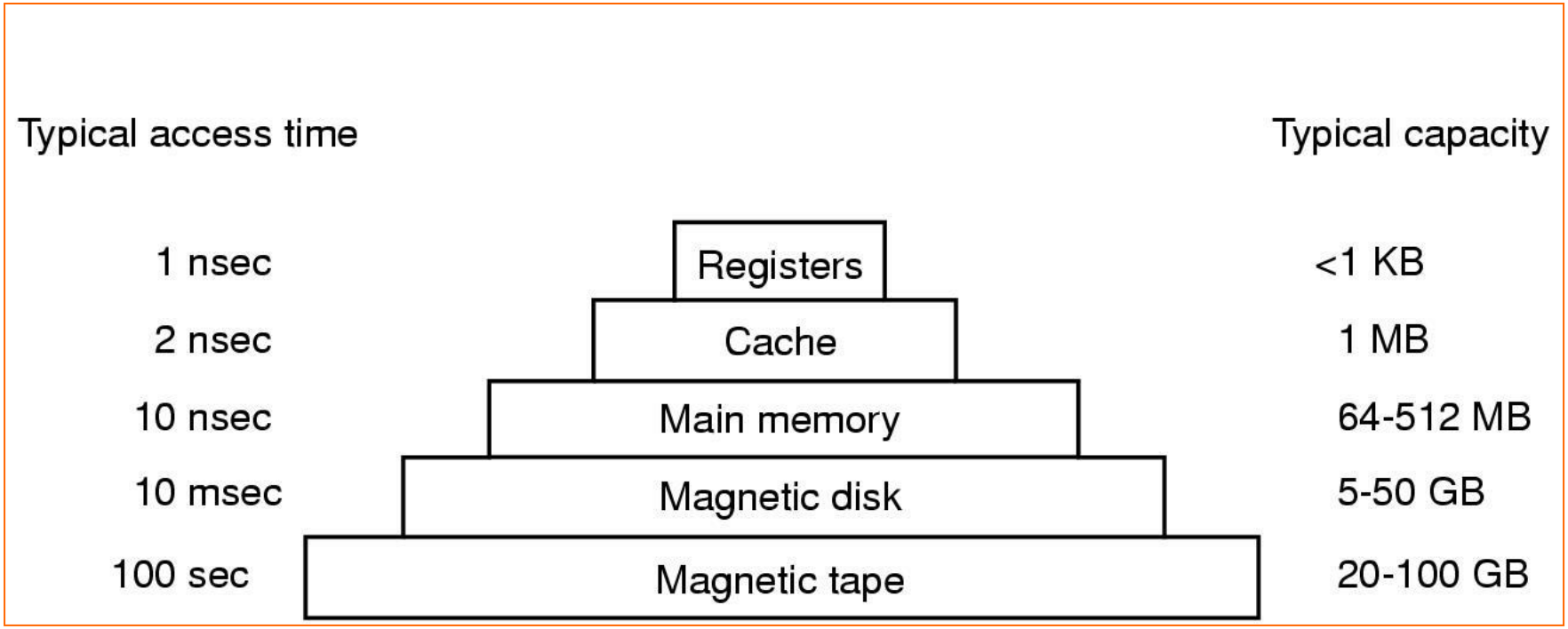
虚拟存储技术，从逻辑上去扩充内存容量。

为了能在逻辑上扩充内存，系统必须具有内存扩充机制，用于实现下述各功能：囍

(1) 请求调入功能。

(2) 置换功能。

- 操作系统课程主要介绍主存储器的管理，但整个存储体系还有更多的层次



存储体系举例

1.4.3 设备管理功能

主要任务: 完成用户进程提出的I/O请求； 为用户进程分配其所需的I/O设备；提高CPU和I/O设备的利用率；提高I/O速度；方便用户使用I/O设备。

设备管理应具有缓冲管理、设备分配和设备处理，以及虚拟设备等功能。囂

1. 缓冲管理囍

CPU运行的高速性和I/O低速性间的矛盾.

现代计算机系统中， 都毫无例外地在内存中设置了缓冲区，而且还可通过增加缓冲区容量的方法，来改善系统的性能。囍

最常见的缓冲区机制有单缓冲机制、能实现双向同时传送数据的双缓冲机制，以及能供多个设备同时使用的公用缓冲池机制。

2. 设备分配

基本任务-- 根据用户进程的I/O请求、系统的现有资源情况以及按照某种设备**分配策略**，为之分配其所需的设备。如果在I/O设备和CPU之间，还存在着设备控制器和I/O通道时，还须为分配出去的设备分配相应的控制器和通道。

3. 设备处理【设备驱动程序】

-- 实现CPU和设备控制器之间的通信.

处理过程：

1. 检查I/O请求的合法性，
2. 了解设备状态是否空闲，
3. 了解有关的传递参数及设置设备的工作方式。
4. 向设备控制器发出I/O命令，
5. 启动I/O设备去完成指定的I/O操作。

设备驱动程序还应能及时响应由控制器发来的中断请求

对于设置了通道的计算机系统， 应能根据用户的I/O请求，自动地构成通道程序。

1.4.4 文件管理功能

1. 文件存储空间的管理囍

- 记录文件存储空间的使用情况的数据结构，
- 对存储空间进行分配和回收的功能。
- 为了提高存储空间的利用率，对存储空间的分配，通常是采用离散分配方式，以减少外存零头，并以盘块为基本分配单位。盘块的大小通常为512 B~8 KB。

2. 目录管理

目录项。目录项包括文件名、文件属性、文件在磁盘上的物理位置等。

由若干个目录项又可构成一个**目录文件**。

目录管理的主要任务， 是为每个文件建立其目录项，并对众多的目录项加以有效的组织，以实现方便的按名存取。 其次，目录管理还应能实现文件共享， 提供目录查询手段 。

3. 文件的读/写管理和保护囍

(1) 文件的读/写管理。从外存中读取数据；或将数据写入外存。

检索文件目录，从中获得文件在外存中的位置。

利用文件读(写)指针，对文件进行读(写)。

修改读(写)指针，为下一次读(写)做好准备。

(2) 文件保护。① 防止未经核准的用户存取文件；

② 防止冒名顶替存取文件；

③ 防止以不正确的方式使用文件。

1.4.5 用户接口

1. 命令接口

(1) 联机用户接口。 （键盘命令及命令解释程序）

(2) 脱机用户接口 (批处理作业用户)。

2. 程序接口㠯

•用户程序取得操作系统服务的惟一途径。

•由一组系统调用组成。

每一个系统调用都是一个能完成特定功能的子程序，每当应用程序要求OS提供某种服务(功能)时，便调用具有相应功能的系统调用。

3. 图形接口

方便性：用户已完全不必像使用命令接口那样去记住命令名及格式，从而把用户从繁琐且单调的操作中解脱出来。

- GUI 系统的复杂性
- 资源利用率的权衡

1.5 操作系统的结构设计

- 操作系统作为一个大型的系统软件，是采用软件工程学的方法开发的。
- 设计原则：保证系统的可靠性，可维护性。

1.5.1 软件工程的基本概念

1. 软件:包括指令，程序，文档
2. 软件工程:囍运用系统的、规范的和可定量的方法，来开发、运行和维护软件。囍

1.5.2 传统的操作系统结构

回顾OS的开发过程，先后引入了分解、模块化、 抽象和隐蔽等方法。 OS结构可分成**无结构（单体结构），模块化结构，层次结构，以及微内核结构**。

- OS结构的更新换代：把第一代至第三代的OS结构，称为传统的OS结构，而把微内核的OS结构称为现代OS结构。

1. 无结构或简单结构操作系统菖

•操作系统设计者注重功能的实现和高的效率上，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，各过程之间可以相互调用，在操作系统内部不存在清晰的结构，因此，这种OS是**无结构**的，也有人把它称为整体系统结构。

- 该阶段OS设计的技巧，只注重编制紧凑的程序，以便于有效地利用内存；
- 对GOTO语句的使用不加限制，导致操作系统庞大而杂乱，缺乏清晰的结构。
- 后果：使所编制出的程序错误很多，调试困难；也使程序难以阅读和理解，增加了维护人员的负担。

2. 模块化OS结构囍

1) 模块化结构囍

- 模块化程序设计技术：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**。
- 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。

•分解– 自顶向下逐步细分

•模块化- 功能与模块对应，并定义模块之间的接口

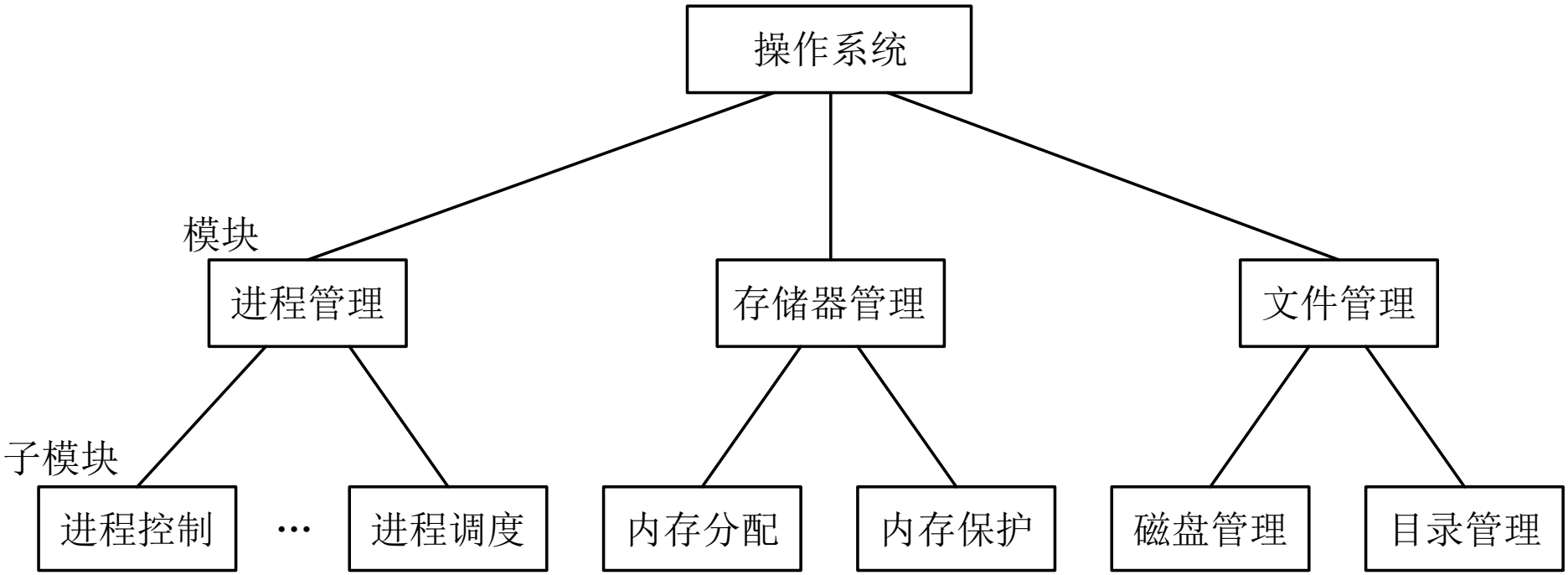


图 1-5 模块化操作系统结构

2) 模块化OS的优缺点囍

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并不精确或错误，很难保证模块完全正确， 导致模块装配成OS时发生困难；

2. 功能模块的划分受系统资源制约关系的影响，例如未能将共享资源和独占资源加以区别； 使模块间存在着复杂的依赖关系，使OS结构变得不清晰。

3. 分层式OS结构㠯

1) 有序分层的基本概念㠯

使我们的每一步设计都是建立在可靠的基础上。

简化了调试和系统验证过程。

2) 层次的设置㠯

(1) 程序嵌套。 考虑在实现OS 的每个功能时所形成的程序嵌套(功能逻辑关系)。

(2) 运行频率。在分层结构中，各层次软件的运行速度是不同的，因为A_1层软件能直接在物理机器上运行， 故它有最高的运行速度。随着层次的增高，其相应软件的运行速度就随之下降，因而A_n层软件的运行速度最低。 为了提高OS的运行效率，应该将那些经常活跃的模块放在最接近硬件的A_1层，如时钟管理、进程调度，通常都放在A_1层。

(3) 公用模块。应把供多种资源管程程序调用的公用模块，设置在最低层，不然，会使比它低的层次模块由于无法调用它而须另外配置相应功能的模块。例如，用于对信号量进行操作的原语Signal和Wait。囍

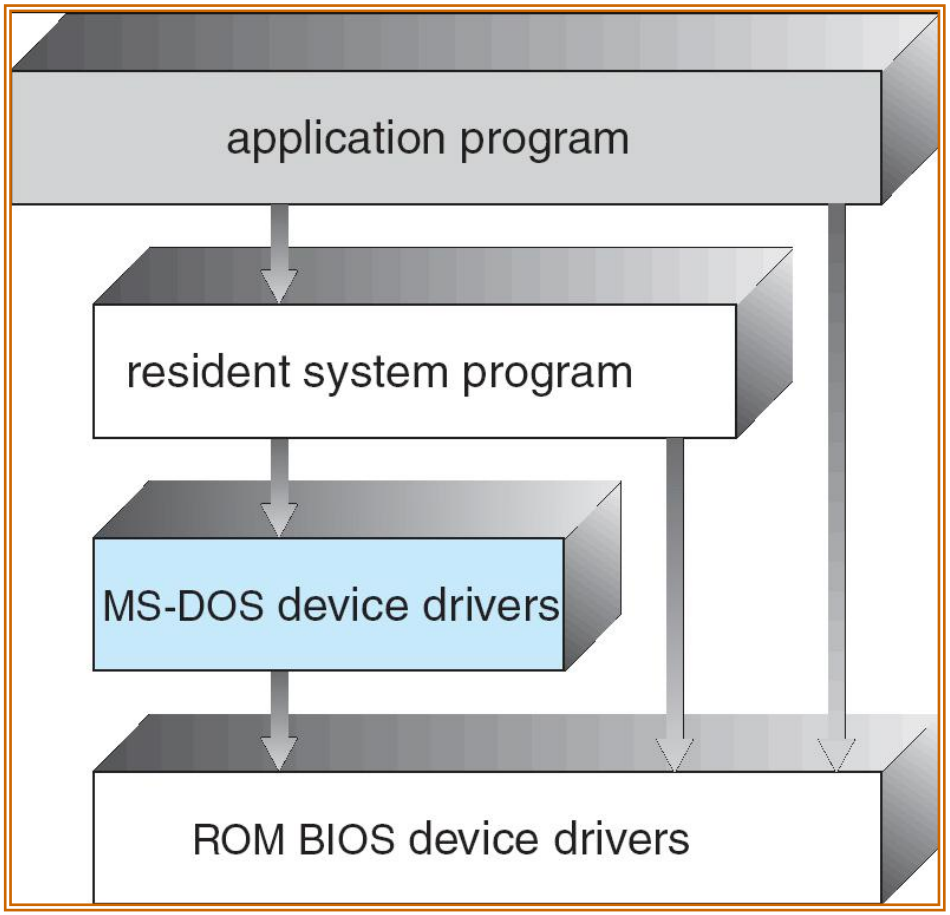
(4) 用户接口。为方便用户(程序)，OS向用户提供了“用户与OS的接口”，如命令接口、程序接口以及图形用户接口。这些接口应设置在OS的最高层，直接提供给用户使用。

OS结构举例 【Silberschatz 教材 p57.】

1. Simple Structure

- MS-DOS
  - – written to provide the most functionality in the least space
  - Not divided into modules
  - Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated

MS-DOS Structure

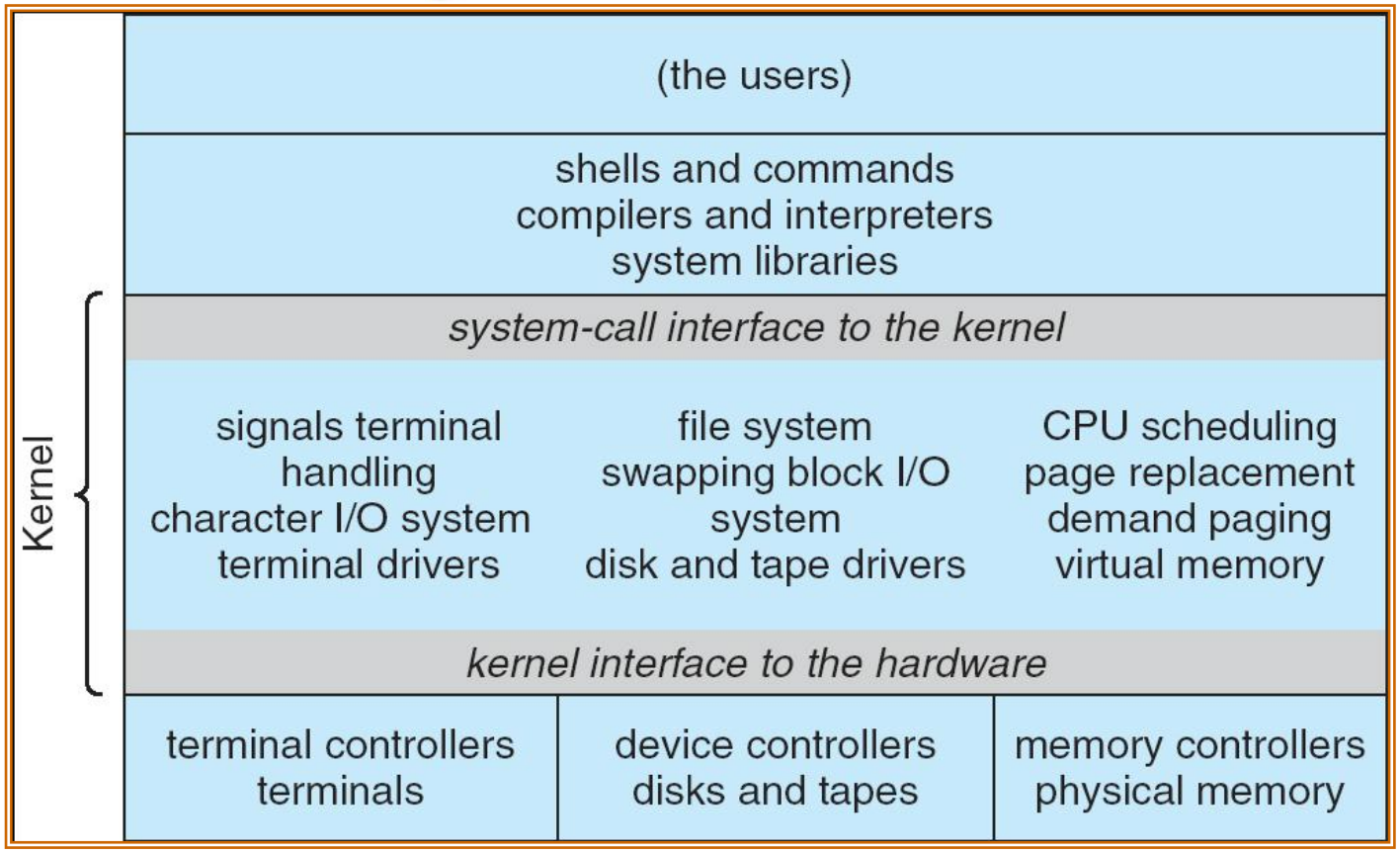


- **MS-DOS的结构没有细致的划分模块**

UNIX

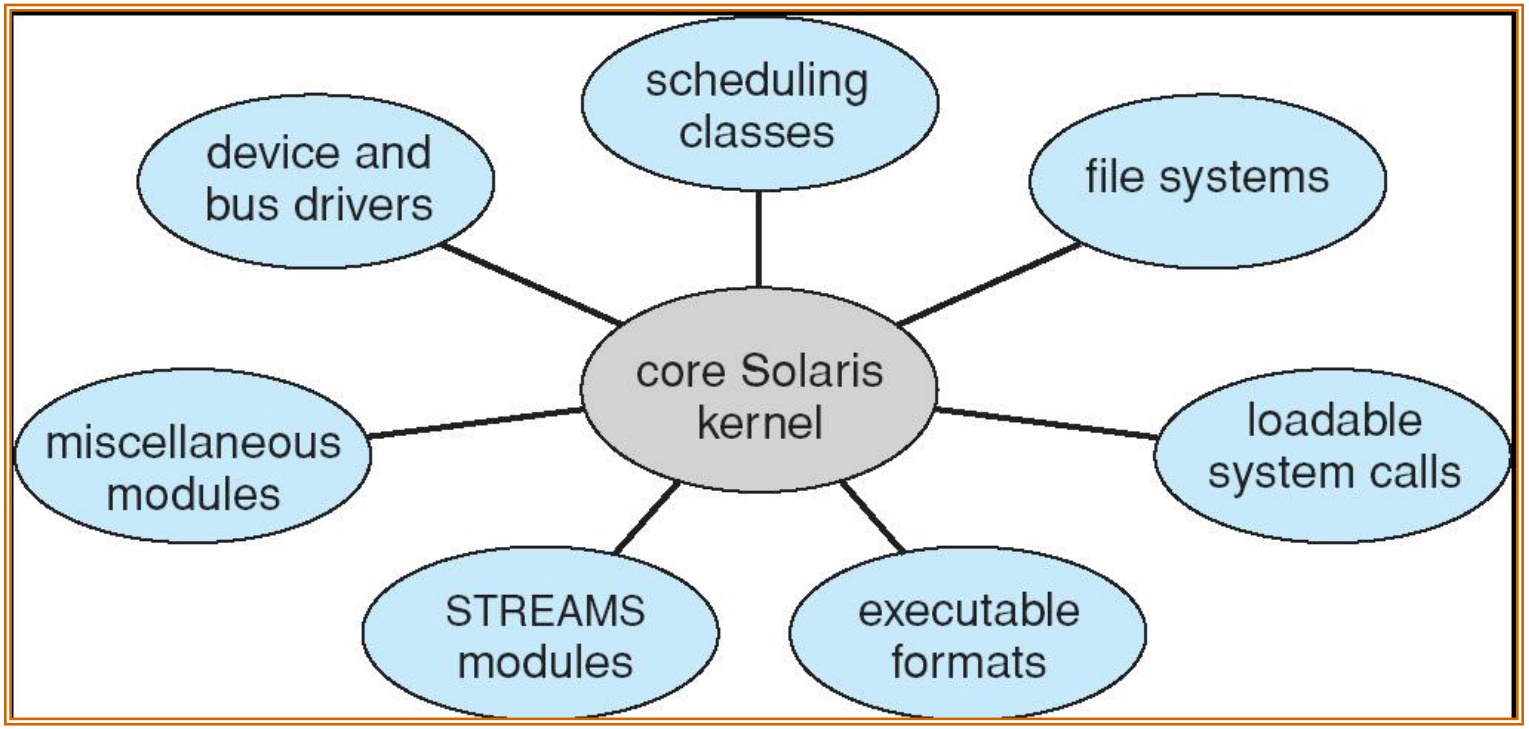
- UNIX – limited by hardware functionality, the original UNIX operating system had limited structuring. 早期UNIX结构受限于硬件
- The UNIX OS consists of two separable parts：
  - Systems programs
  - The kernel
    - Consists of everything below the system-call interface and above the physical hardware
    - Provides the file system, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level

UNIX System Structure



- 【内核：包括在系统调用接口和和下层的硬件接口之间的所有部分。使得系统难以扩展，因为改动会影响到很多其他部分。】

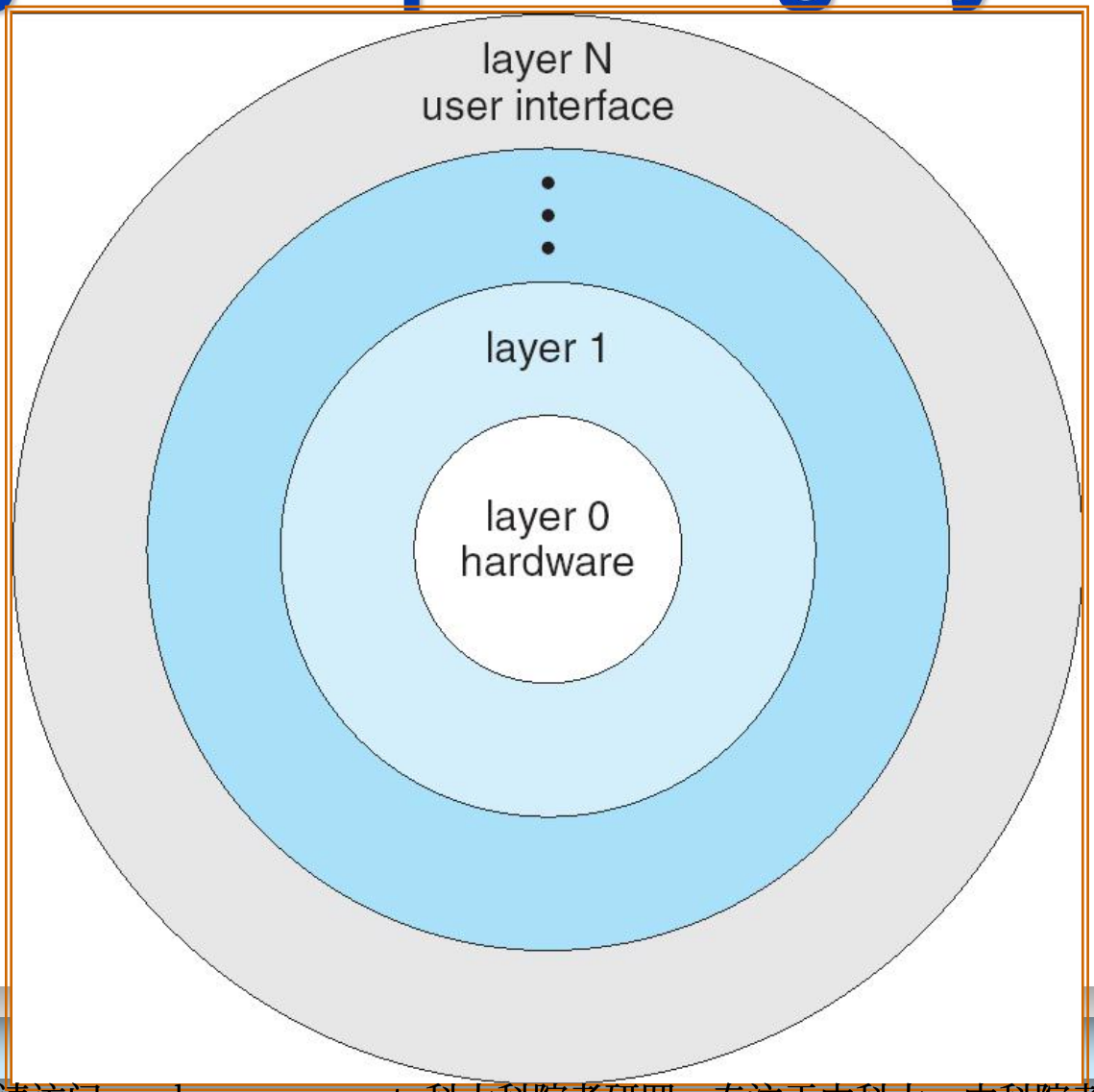
Solaris Modular Approach



Modules

- Most modern operating systems implement kernel modules
  - Uses object-oriented approach
  - Each core component is separate
  - Each talks to the others over known interfaces
  - Each is loadable as needed within the kernel
- Overall, similar to layers but with more flexible

2. Layered Operating System



Layered Approach

- The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers

分层式结构设计的**基本原则**是：每一层都仅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容易

1.5.3 微内核OS结构

1. 客户/服务器模式(Client-Server Model)

1) 基本概念

为了提高OS的灵活性和可扩充性而将OS划分为两部分:

1. 用于提供各种服务的一组服务(进程)，如用于提供进程管理的进程服务器、提供存储器管理的存储器服务器提供文件管理的文件服务器等，所有这些服务器(进程)都运行在**用户态**。
2. 内核，用来处理客户和服务器之间的通信， 即由内核来接收客户的请求，再将该请求送至相应的服务器；同时它也接收服务器的应答，并将此应答回送给请求客户。

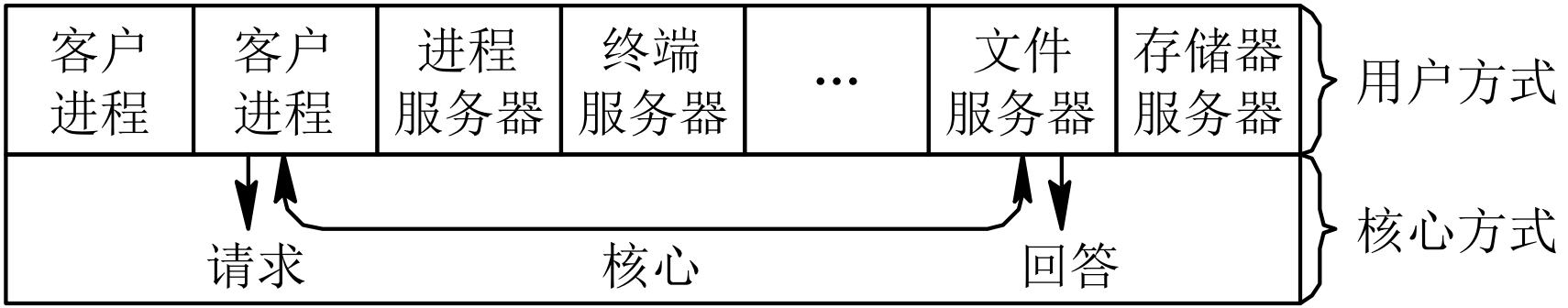


图 1-6 单机环境下的客户/服务器模式

2) 客户/服务器模式的优点囍

((1))提高了系统的灵活性和可扩充性。

(2) 提高了OS的可靠性。

(3) 可运行于分布式系统中。

➢对操作系统结构设计的意义在于，
可以把所有非基本部分从内核中移走。

2. 面向对象的程序设计技术(Object-Orientated Programming)

1) 面向对象技术的基本概念

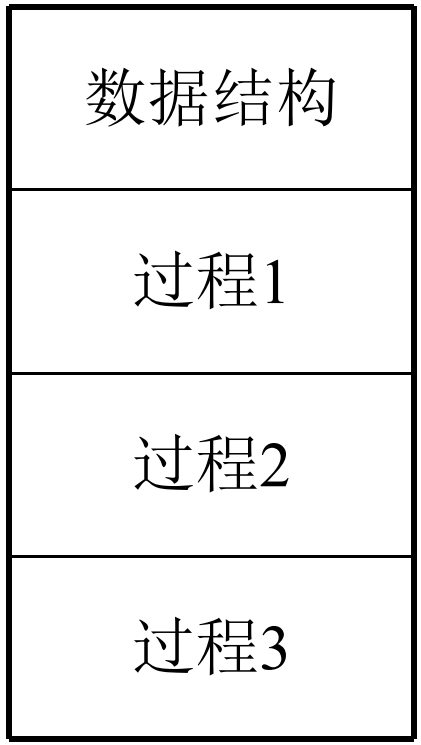


图 1-7 一个对象的示意图

2) 面向对象技术的优点菖

(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。菖

(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。菖

(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。

3. 微内核技术㬎

1) 微内核技术的引入

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。

2) 微内核的基本功能囍

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、存储器管理、进程间通信、 低级I/O功能。囍

(1) 进程管理。

(2) 存储器管理。

(3) 进程通信管理。

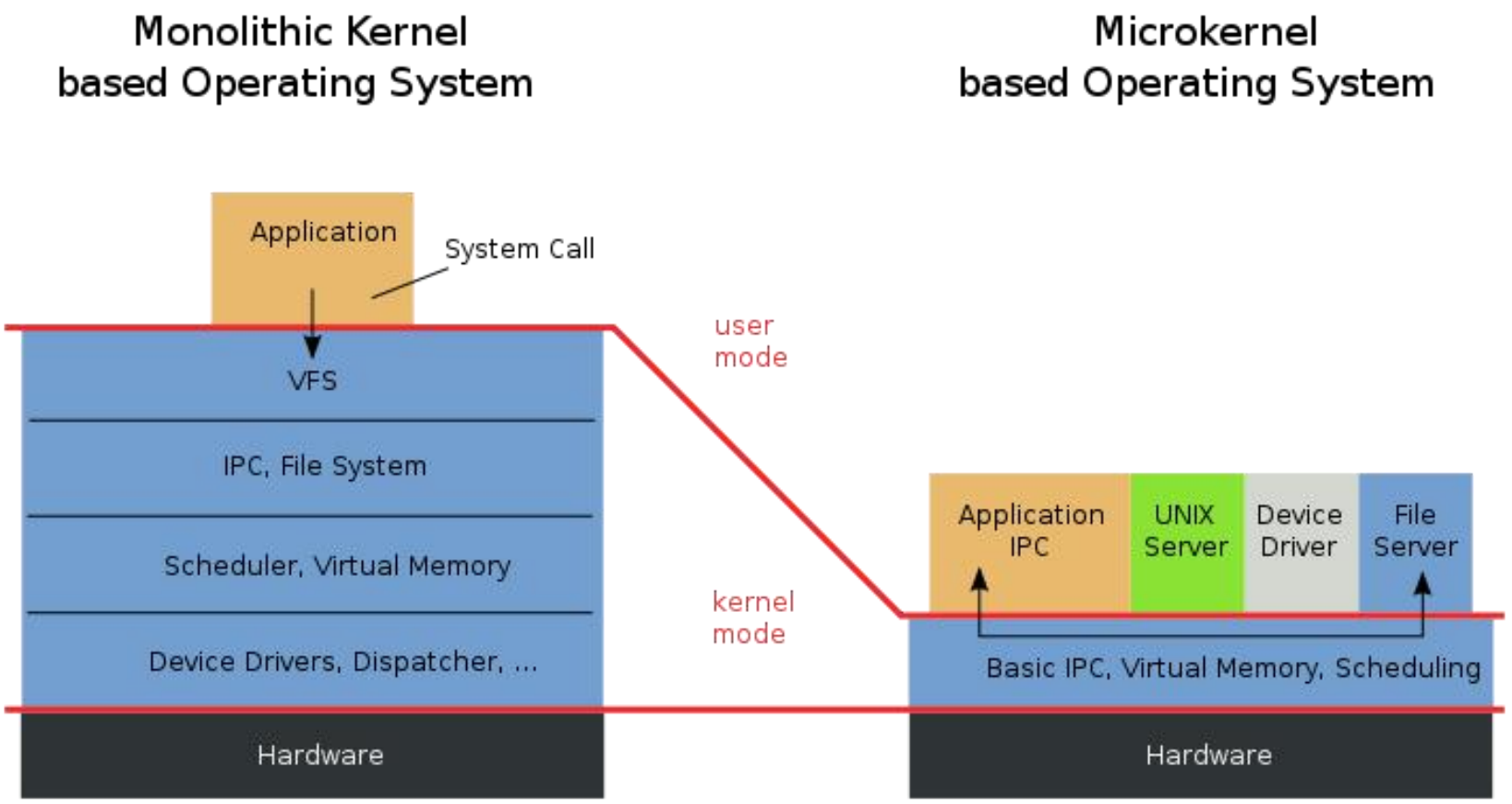
(4) I/O设备管理。

➢哪些服务应该保留在内核中，哪些应该在用户空间实现。没有定论。

- 微内核技术的优点：
  - 系统可扩充性- 新增服务会加到用户空间，内核不需要修改或改动极小。
  - 移植性- 小内核便于移植到不同的硬件平台。
  - 安全性和可靠性- 大多数服务是用户进程而不是系统进程。即使某个服务失败，不会影响操作系统其他部分。

OS Structure

- 整体结构和微内核结构的对比



Microkernel System Structure

- Moves as much from the kernel into "*user*" space
- Communication takes place between user modules using message passing
- Benefits:
  - Easier to extend a microkernel
  - Easier to port the operating system to new architectures
  - More reliable (less code is running in kernel mode)
  - More secure
- Detriments:
  - Performance overhead of user space to kernel space communication

【补充】研究操作系统的几种观点

- 资源管理的观点
- 进程的观点
- 虚机器观点
- 服务提供者观点

（1）资源管理的观点

操作系统- 作为资源管理者。

目的

- 实现资源共享
- 提高资源利用率

硬件资源：

CPU，内存，外部设备(I/O设备，外存，时钟，网络接口等)

软件资源：

硬盘上的文件，信息

两种方式实现资源复用（共享）：时间 及 空间

管理资源涉及的工作

- 记录资源使用状况

  如 哪些资源空闲，好坏与否，被谁使用，使用多长时间等
- 申请资源
- 分配资源

  静态分配策略

  （在程序运行前分配，但效率不高）

  动态分配策略

  （在程序运行过程中何时用资源，何时分配。其缺点是会出现死锁）
- 回收资源

（2）进程的观点

从操作系统运行的角度动态地观察操作系统

从这个观点来看：

操作系统是由一些可同时独立运行的进程（运行的程序）和一个对这些进程进行协调的核心组成

（3）虚机器观点

从操作系统内部结构来看：

- 把操作系统分成若干层
- 每一层完成其特定功能，从而构成一个虚机器，并对上一层提供支持
- 通过逐层功能扩充，最终完成整个操作系统虚机器
- 而操作系统虚机器向用户提供各种功能，完成用户请求

（4） 服务提供者的观点

从用户角度来看：

操作系统为用户提供一组功能强大的、方便易用的命令或系统调用

操作系统作为 标准服务提供者

- 提供每个用户需要的标准工具

操作系统的定义：

•OS是叠加在硬件上的第一层软件，是其他软件和硬件之间的接口。
•OS有效地管理计算机的软硬件资源，合理地组织计算机的工作流程，方便用户使用系统。

其它定义性描述：

Operating System Definition

- OS is a resource allocator 资源分配者
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program 控制程序
  - Controls execution of programs to prevent errors and improper use of the computer

第一章作业：

1. 课后题：【p25： 1，3，6，9，16】
2. 操作系统设计采用微内核技术有哪些优点？
3. 简述研究操作系统有哪几种主要观点？

End of lecture 1