



第2章 操作系统结构

Operation-System Structures

■ 本章目标 CHAPTER OBJECTIVES

- 介绍操作系统提供的服务

Identify services provided by an operating system.

- 讨论组织操作系统的不同方法

To discuss the various ways of structuring an operating system

- 说明如何使用系统调用来提供操作系统服务

- Illustrate how system calls are used to provide operating system services.





2.1 操作系统服务

Operating-System Services

- 操作系统提供执行程序的环境，它向程序及其用户提供一组服务。

An operating system provides an environment for the execution of programs. It provides certain services to programs and to the users of those programs.



操作系统提供的服务

operating system provides certain services

- **用户接口**：包括命令行接口、批处理接口及图形用户接口

User interface: include command-line interface, batch interface and graphical user interface

- **程序执行**：将程序装入内存并运行程序

Program execution: the system must be able to load a program into memory and to run it.

- **I/O操作**：操作系统必须提供完成I/O操作的手段

I/O operations: the operating system must provide some means to perform I/O.



操作系统提供的服务2

operating system provides certain services

- **文件系统操纵**：让程序能够读、写、创建和删除文件

File-system manipulation: program capability to read, write, create, and delete files.

- **通信**：进程间交换信息。

Communications: exchange of information between processes

- **错误检测**：通过检测错误确保正确运算

Error detection: ensure correct computing by detecting errors





操作系统提供的服务3

operating system provides certain services

- **资源分配**: 把资源分配给同时运行的作业
Resource allocation: allocating resources to multiple jobs running at the same time.
- **统计**: 记录用户使用资源的类型及数量
Accounting: keep track of which users use how much and what kinds of computer resources.
- **保护和安全**: 确保所有对资源的访问是受控的且系统不受外界侵犯
Protection and security: ensuring that all access to system resources is controlled .
Security of the system from outsiders is also important



2.2 操作系统的用户接口

User Operating-System Interface

- （交互式）用户与操作系统的接口有两种：

There are two fundamental approaches for users interface with operation system

- 命令行接口 command-line interface
- 图形用户接口 graphical user interface





操作系统的接口

- 操作系统向用户提供了各种使用其服务功能的手段，即提供了操作系统接口。
- 操作系统向用户提供的接口：
 - 命令接口
 - 图形接口
 - 程序接口





2.2.1 命令解释程序

Command Interpreter

- 命令解释程序的主要功能是获取并执行用户指定的命令

The main function of the command interpreter is to get and execute the next user-specified command





命令接口

- 用命令接口进行作业控制的主要方式：
 - **脱机控制方式**：用户将对作业的控制要求以作业控制说明书的方式提交给系统，由系统按照作业说明书的规定控制作业的执行。
 - **联机控制方式**：指用户利用系统提供的一组键盘命令或其他操作命令和系统会话，交互式地控制程序的执行。





联机命令接口

- 联机命令接口提供一组键盘命令供用户请求计算机系统服务。
- 键盘命令可分为：
 - **内部命令**：命令功能简单、程序短小、使用频繁。它们在系统初始启动时被引导至内存并常驻内存。
 - **外部命令**：命令功能较复杂、程序较长、独立作为一个文件驻留在磁盘上，当需要它们时，再从磁盘上调入内存运行。





脱机命令接口

- 脱机命令接口由一组作业控制命令组成。
- 脱机用户不能直接干预作业的运行，他们应事先用作业控制命令写一份作业操作说明书，连同作业一起提交给系统。
- 当系统调度到该作业时，由系统中的命令解释程序对作业说明书上的命令或作业控制语句逐条解释执行。





2.2.2 图形用户接口

Graphical User Interface

- GUI提供基于鼠标的窗口和菜单系统作为接口

A GUI allows provides a mouse-based window-and-menu system as an interface





图形用户接口

- 图形用户接口是通过对出现在屏幕上的对象直接进行操作，以控制和操纵程序的运行。
- 图形用户接口减少或免除了用户的记忆工作量，其操作方式从原来的记忆并键入改为选择并点取，极大地方便了用户。
- 目前图形用户接口是最为常见的人机接口形式，可以认为图形接口是命令接口的图形化。





2.3 系统调用 System Calls

- 系统调用在运行程序和操作系统之间提供接口

System calls provide the interface between a running program and the operating system.

- 运行程序和操作系统之间的参数传递有3种常用方式:

Three general methods are used to pass parameters between a running program and the operating system.





系统调用2

System Calls

- 寄存器中的参数传递

Pass parameters in *registers*.

- 参数存在内存的一张表中，表地址作为寄存器的参数传递

Store the parameters in a table in memory, and the table address is passed as a parameter in a register.

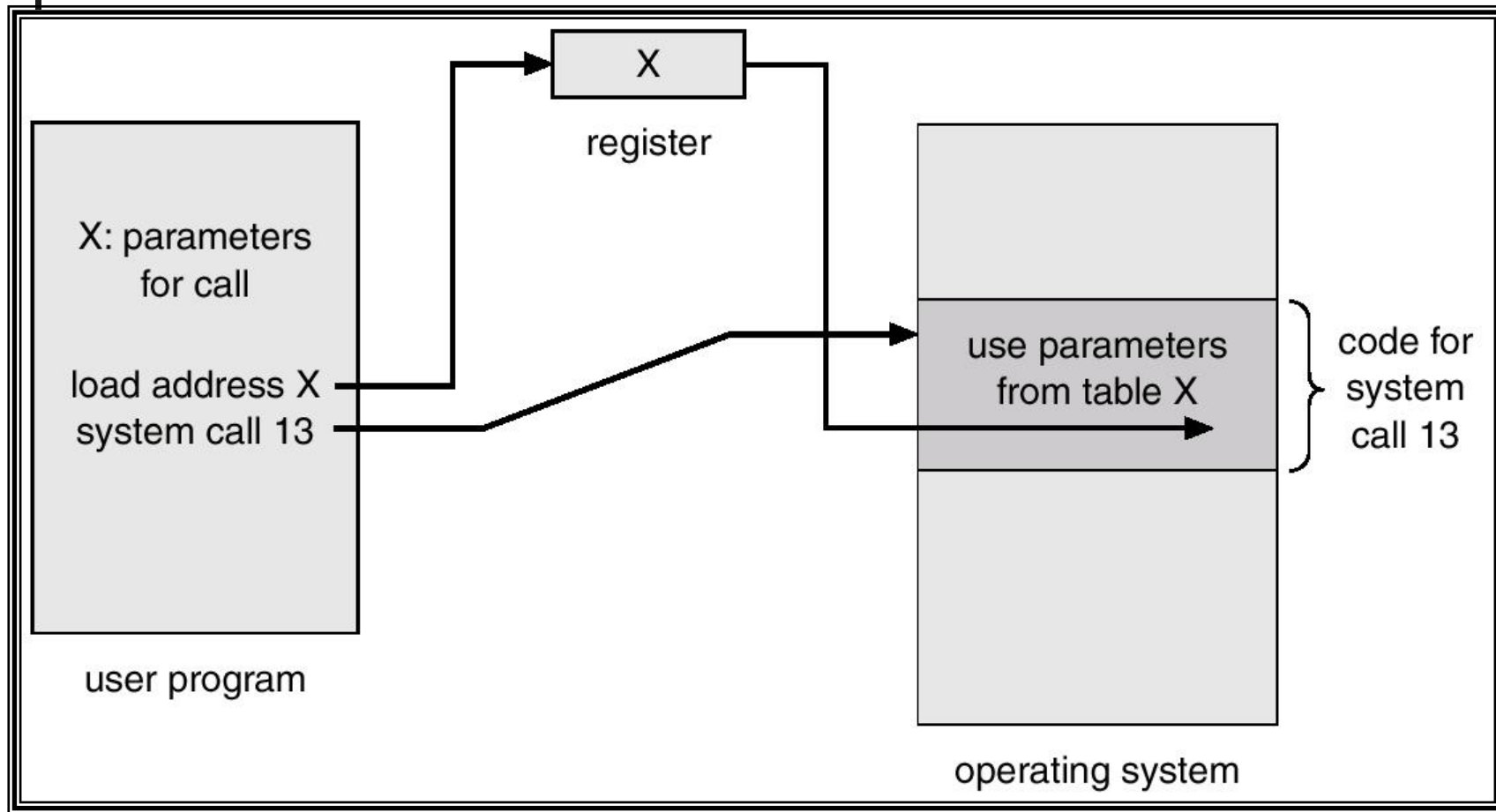
- 程序把参数压入栈，由操作系统弹出

Push (store) the parameters onto the stack by the program, and pop off the stack by operating system.



表格方式的参数传递

Passing of Parameters As A Table





程序接口

- **程序接口**由一组系统调用命令组成。有时也称为系统调用接口。
- 用户通过在程序中使用这些系统调用命令来请求操作系统提供的服务。





系统调用

- **系统调用**：由若干条指令构成的过程，用以实现特定的操作系统服务功能。
- 系统调用命令有时也称为广义指令。它是由操作系统提供的一个或多个子程序模块实现的。





系统调用处理程序的执行过程

- 为执行系统调用命令作准备。主要工作是保留现场，并把系统调用命令的参数放入指定的存储单元。
- 执行系统调用。根据系统调用命令的编号找到相应子程序的入口地址，然后转去执行。
- 系统调用命令执行完后的处理。主要工作是恢复现场，并把系统调用的返回参数送入指定存储单元。





系统调用与过程调用的区别

- **运行状态不同**：系统调用在核心态下运行，子程序在用户态下运行。
- **进入方式不同**：系统调用通过中断机构进入以实现运行状态的改变，子程序直接调用不涉及运行状态改变。



- 系统调用可以大致分为**5**类：

System calls can be grouped roughly into five major categories:

- 进程控制 **Process control**
- 文件管理 **File management**
- 设备管理 **Device management**
- 信息维护 **Information maintenance**
- 通信 **Communications**



■ 进程控制 Process control

- 如创建进程、终止进程、等待事件、唤醒事件
分配和释放内存

Creat process, terminate process, wait event,
signal event, allocate and free memory

■ 文件管理 File management

- 如创建文件、删除文件、打开、关闭、读、写
create file, delete file, open, close, read,
write



- 设备管理 Device management
 - 如请求设备、释放设备、读、写
request device, release device, read, write
- 信息维护 Information maintenance
 - 如读取时间或日期、设置时间或日期
Get time or date, set time or date
- 通信 Communications
 - 如创建、删除通信连接，发送、接收消息
Create, delete communication connect, send, receiver messages





2.4 系统服务 System Services

- 系统服务提供程序开发和执行的便利环境

System services provide a convenient environment for program development and execution



- 系统服务可划分为：

System Services can be divided into:

- 文件操纵 **File manipulation**
- 状态信息 **Status information**
- 文件修改 **File modification**
- 程序语言支持 **Programming language support**
- 程序调入和执行 **Program loading and execution**
- 通信 **Communications**





系统服务3 System Services

- 多数用户所看到的操作系统是由应用和系统程序所定义的，而不是系统调用所定义。

The view of operating system seen by most users is defined by the application and system programs, rather than by the actual system calls.





2.8 操作系统结构

Operating System structure

- 本节介绍操作系统的常用模块如何连接起来以组成内核。

In this section, we discuss how these components of operating systems are interconnected and melded into a kernel.





操作系统的内核结构

- 操作系统是一个大型系统软件，其内核结构主要有三种：
 - 模块结构
 - 层次结构
 - 微内核结构





2.8.1 简单结构 Simple Structure

- 许多商业操作系统没有明确定义的结构。通常这些操作系统最初很小、简单且功能有限，但后来渐渐超出了其原来范围。

Many commercial systems do not have well-defined structure. Frequently such operating systems started as small, simple, and limited systems and then grew beyond their original scope.





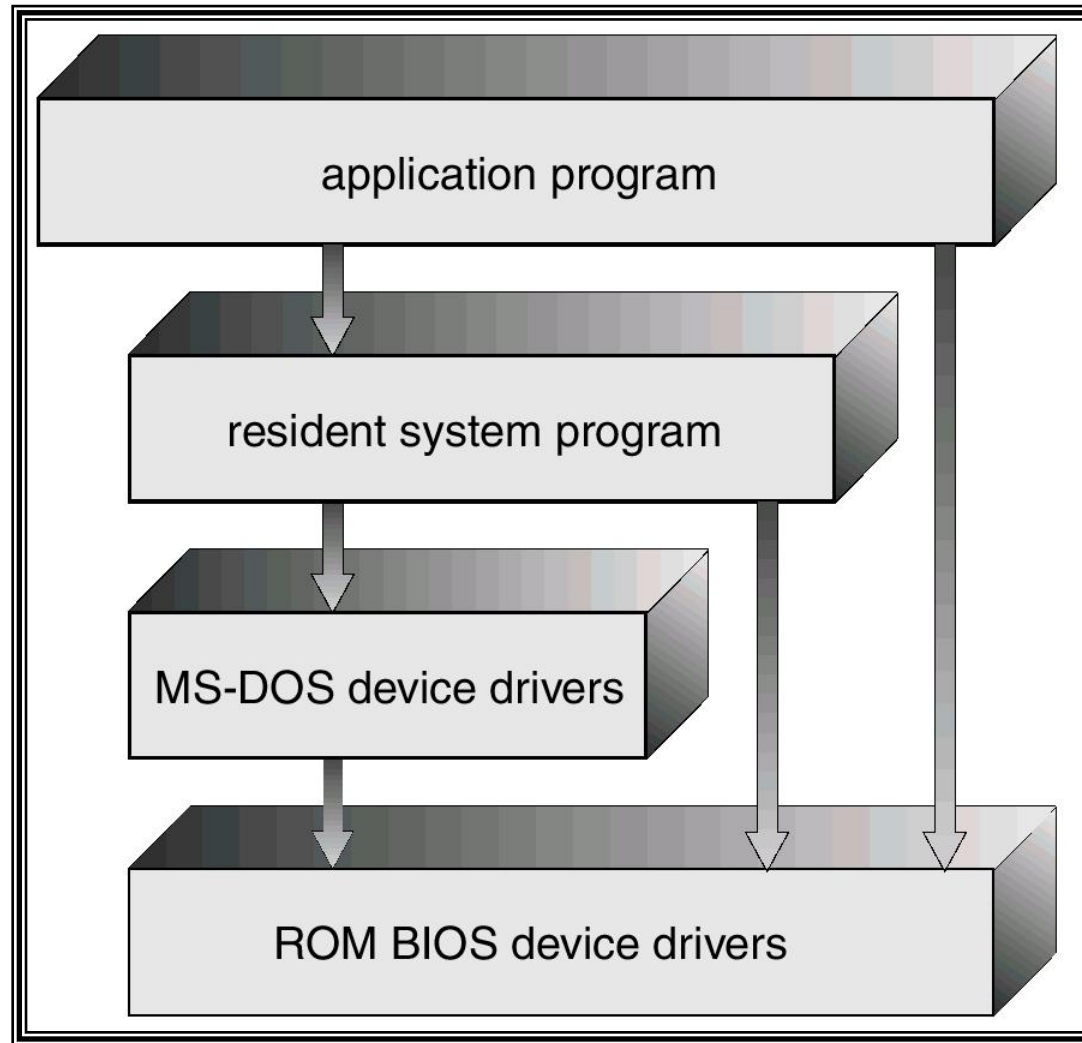
MS-DOS

- MS-DOS-以最小的空间提供最多的功能
MS-DOS – written to provide the most functionality in the least space
 - 不划分模块
not divided into modules
 - 其接口和功能层次没有划分清楚
its interfaces and levels of functionality are not well separated



MS-DOS层次结构

MS-DOS Layer Structure



- 早期UNIX：受硬件功能限制。它由2个独立部分组成。

original UNIX: limited by hardware functionality. It consists of two separable parts

- 系统程序 Systems programs
- 内核 The kernel





UNIX系统结构2

UNIX System Structure

- 内核包括了在物理硬件之上，系统调用之下的一切

The kernel consists of everything below the system-call interface and above the physical hardware

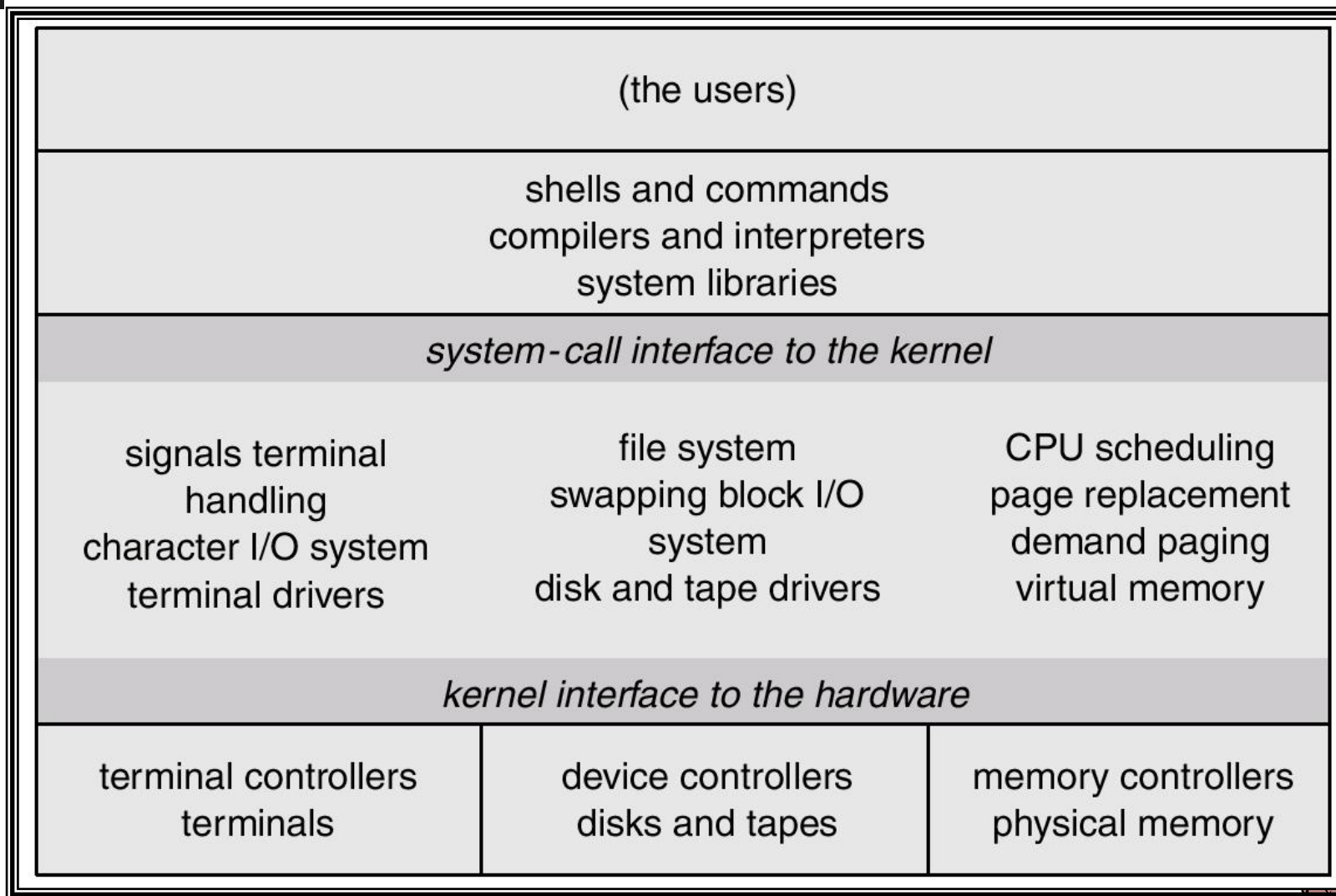
- 内核通过系统调用提供文件系统、CPU调度、存储管理和其他操作系统功能。

The kernel provides the file system, CPU scheduling, memory management, and other operating-system functions.





UNIX系统结构图





2.8.2 分层方法

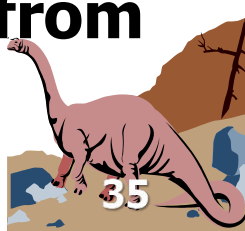
Layered Approach

- 操作系统划分为若干层（级），底层（**0**层）为硬件；最高层（**N**层）为用户接口。

The operating system is broken up into a number of layers (levels). The bottom layer (layer 0) is the hardware; the highest (layer N) is the user interface.

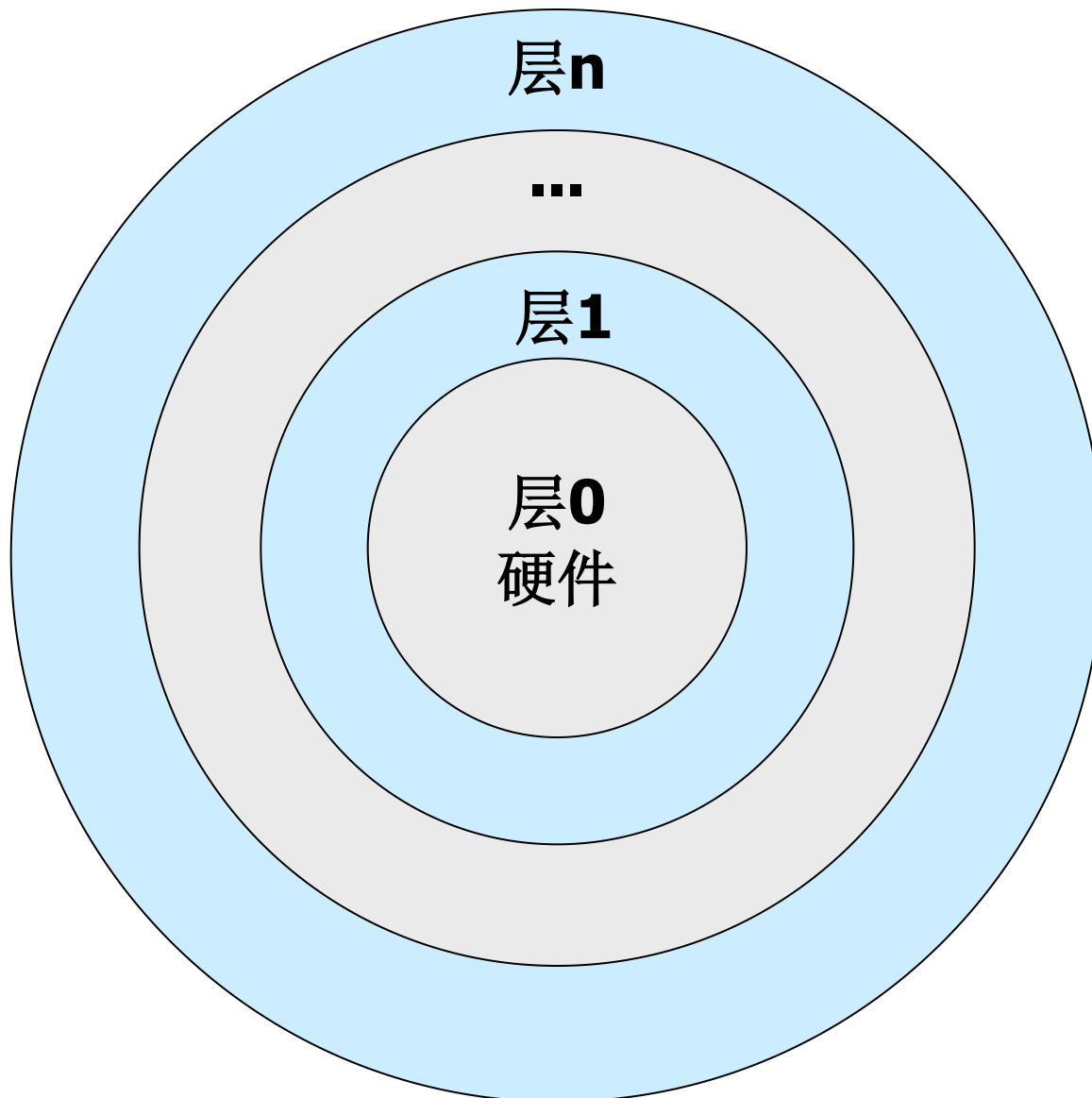
- 每层都利用较低层的功能和服务，为较高层隐藏一定的数据结构、操作和硬件的存在。

layers are selected so that each uses functions (operations) and services of only lower-level layers . each layer hides the existence of certain data structures, operations, and hardware from higher-level layers.



一种分层操作系统

A layered operating system





层次结构

- 层次结构是将操作系统内核按照一定的规则划分为一系列相互依赖的层次，每个层次也可以分解为一系列更小的模块，
- 每个模块完成一个特定的功能，只能与相邻层次发生直接联系，所有这些层次的集合就实现了整个系统。





层次结构特点

- 层次结构是一种特殊的模块结构。
- 其特点为：
 - 给模块赋予了层次顺序，使调用关系变得有序；
 - 在上下两层不变的基础上可以换掉某层，便于移植和扩充。
 - 但以牺牲一定的灵活性为代价。





2.8.3 微内核 Microkernel

- 将所有非基本部分从内核中移走，并将它们实现为系统程序或用户程序，这样得到更小的内核。

Removing all nonessential components from the kernel and implementing them as system and user-level programs, The result is a smaller kernel.

- 微内核通常包括最小的进程和内存管理以及通信功能。

Typically, however, microkernels provide minimal process and memory management, in addition to a communication facility.

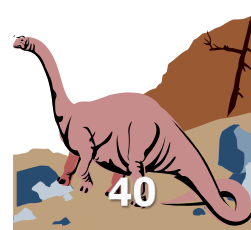




微内核2 Microkernel

- 微内核的主要功能是使客户程序和运行在用户空间的各种服务之间进行通信

The main function of the microkernel is to provide a communication facility between the client program and the various services that are also running in user space.





微内核3 Microkernel

- 优点 Benefits:

- 容易扩充操作系统

- ease of extending the operating system

- 容易从一种硬件平台移植到另一种硬件平台

- easier to port from one hardware design to another

- 更安全可靠 more security and reliability

- 不足：因系统功能总开销增加导致系统性能下降

Performance decreases due to increased system function overhead.





微内核结构

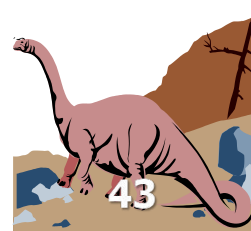
- **微内核结构**将操作系统中的内存管理、设备管理、文件管理等高级服务功能尽可能从内核分离出来，变成几个独立的非内核模块
- 而在内核只保留少数最基本的功能，如调度、进程间通信、地址空间支持等，使内核变得简洁可靠。





微内核结构特点

- 降低了开发难度，具有较好的扩展性及移植性，特别适合大规模开放式的分布系统。
- 但效率较低。





2.8.4 模块 Modules

- 也许最新的操作系统设计方法是用面向对象编程技术来生成模块化的内核。

Perhaps the best current methodology for operating-system design involves using object-oriented programming techniques to create a modular kernel.

- 内核有一组核心部件，以及在启动或运行时对附加服务的动态链接。

The kernel has a set of core components and dynamically links in additional services either during boot time or during run time.



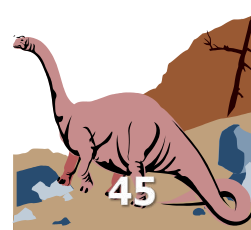


模块2 Modules

- 例如Solaris组织成7个可加载的内核模块及一个核心内核所构成：

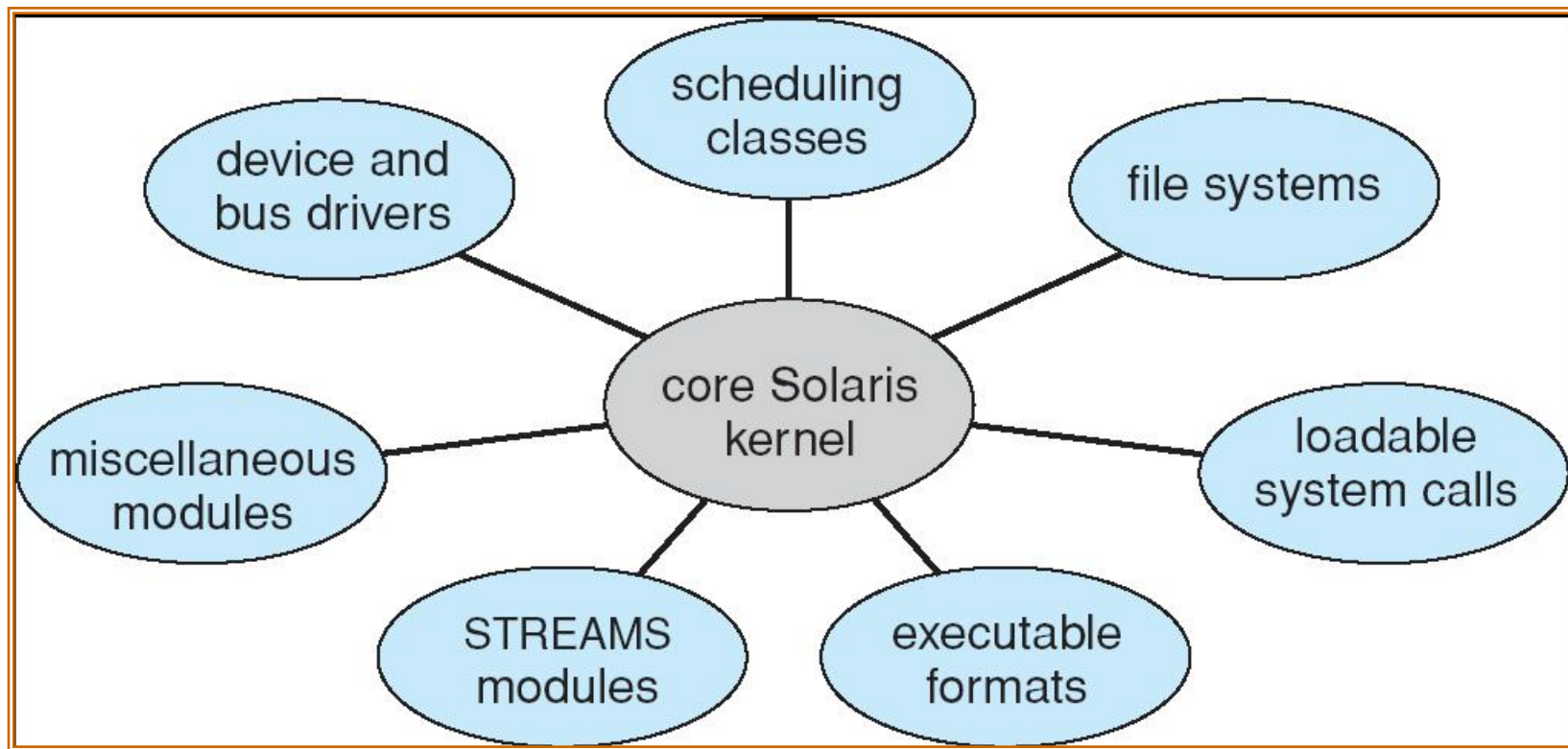
For example ,the Solaris operating system structure is organized around a core kernel with seven types of loadable kernel modules:

- 调度类 Scheduling classes
- 文件系统 File systems
- 可加载的系统调用 Loadable system calls
- 可执行格式 Executable formats
- STREAMS模块 STREAMS modules
- 杂项模块 Miscellaneous modules
- 设备和总线驱动 Device and bus drivers



可加载的Solaris模块

Solaris loadable modules





模块结构

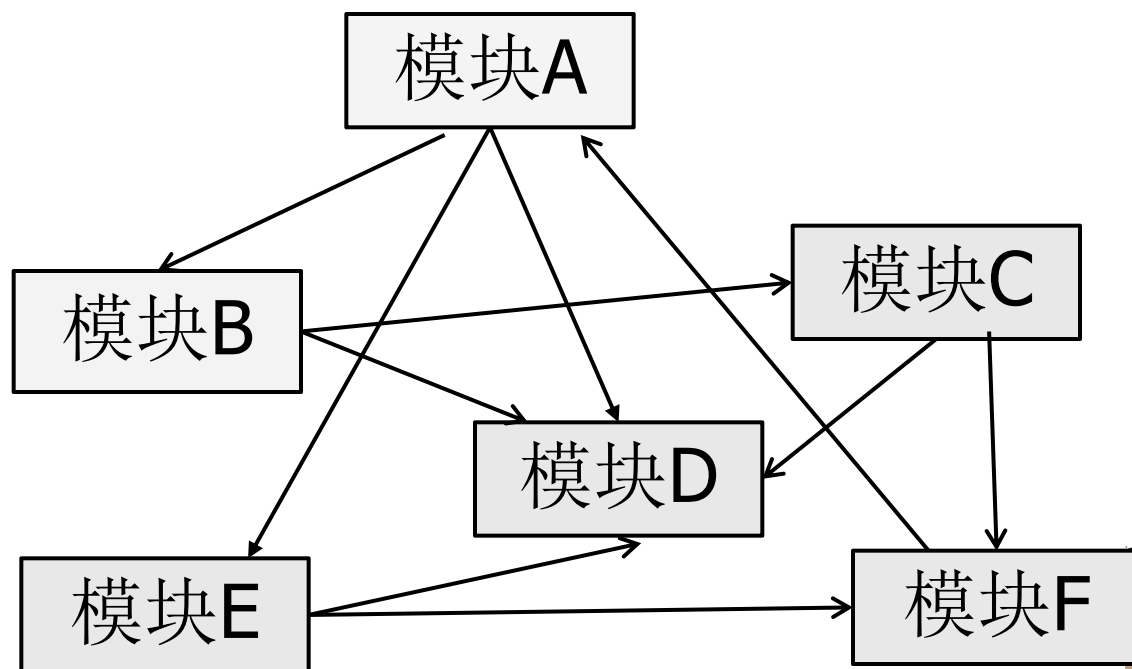
- **模块结构**将操作系统内核按照功能划分为一个个独立的模块，模块之间相对独立，只能通过事先规定好的接口方式来调用。
- 每个模块实现一个完整独立的功能，所有模块之间相互调用，共同构成一个完整的系统内核。





模块结构特点

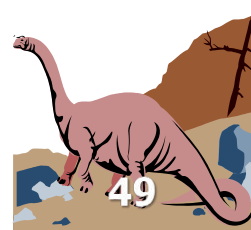
- 效率高；
- 但全局函数使用多造成访问控制困难；
- 结构不清晰，可理解性、可维护性及可移植性差。





宏内核与微内核

- 操作系统内核按照其运行的情况可分为：
 - **宏内核**：也称单内核，在运行过程中，它是一个独立的进程。模块结构、层次结构的系统内核基本都是宏内核。**Linux**系统属于宏内核类型。
 - **微内核**：微内核中，大部分内核模块都作为独立的进程，它们之间通过消息通信使模块之间互相提供服务。微内核本身类似一个消息管理器，通过合理组织内核模块来保证只调入最需要的模块运行。**Windows NT/2000**系统属于微内核类型。





练习

- 2.1 What is the purpose of system calls?





选择题1

- 操作系统提供给编程人员的接口是_____。
 - A. 系统调用
 - B. 子程序
 - C. 库函数
 - D. 高级语言





考研题

- 下列选项中，操作系统提供给应用程序的接口是____。10
- | | |
|---------|-------|
| A. 系统调用 | B. 中断 |
| C. 库函数 | D. 原语 |

