高等院校计算机应用系列教材

操作系统原理与实践教程

(第四版)

史苇杭　卫　琳　主　编

清華大學出版社

北　京

内 容 简 介

本书全面讲述计算机操作系统的基本原理和相关技术。全书共分为10章，深入介绍操作系统的发展历程、通用操作系统的启动过程、处理器管理、进程管理、存储器管理、文件管理、设备管理、系统安全、嵌入式操作系统等内容。

本书内容丰富、结构合理、知识体系完备，主要面向计算机及相关专业学生。本书适合作为普通高等院校操作系统原理课程的教材，也可作为各类培训班教材或自学者的参考用书，对操作系统及其上层应用程序的开发人员也具有较好的参考价值。

本书的电子课件和习题答案可以到 http://www.tupwk.com.cn/downpage 网站下载，也可以扫描前言中的二维码获取。

图书在版编目(CIP)数据

操作系统原理与实践教程 / 史苇杭，卫琳主编.—4 版. —北京：清华大学出版社，2022.4
高等院校计算机应用系列教材
ISBN 978-7-302-60340-5

Ⅰ.①操… Ⅱ.①史… ②卫… Ⅲ.①操作系统—高等学校—教材 Ⅳ.①TP316

中国版本图书馆 CIP 数据核字(2022)第 043600 号

责任编辑：胡辰浩
封面设计：高娟妮
版式设计：孔祥峰
责任校对：马遥遥
责任印制：杨 艳

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-83470000 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市科茂嘉荣印务有限公司
经 销：全国新华书店
开 本：185mm×260mm **印 张**：19.75 **字 数**：506 千字
版 次：2006 年 10 月第 1 版 2022 年 4 月第 4 版 **印 次**：2022 年 4 月第 1 次印刷
定 价：79.00 元

产品编号：087722-01

前　言

操作系统是计算机系统中最重要的系统软件，它管理整个计算机系统的软件和硬件资源；是其他软件和程序的运行基础；是沟通用户与计算机硬件的桥梁。操作系统因其在计算机系统中所处的地位决定了其重要性，它是计算机科学与技术专业的一门专业基础课，是计算机相关专业学生的必修课程。操作系统是计算机领域较活跃的学科之一，其发展极为迅速。

操作系统具有如下特点。

(1) 内容庞杂，涉及面广。操作系统是一个庞大的系统软件，它管理系统中所有的软件、硬件资源，控制计算机的工作流程，提供用户与计算机之间的接口。因此，操作系统课程的内容非常庞大且复杂。

(2) 内容抽象。操作系统在计算机系统中处于裸机与应用层之间，对下与硬件接口，对上提供简单便捷的用户界面。但是，关于操作系统的内容，如操作系统的概念、操作系统的功能以及这些功能如何实现等对于用户或学习者而言仍是比较抽象和费解的。

(3) 发展变化快。操作系统是计算机领域较活跃的学科之一，其发展极为迅速，随着计算机的发展而不断更新，是计算机软件中更新较快的软件，因而更加重了学习难度。

正是由于操作系统的上述特点，使得操作系统课程的学习具有相当的难度。为了解决这些问题，提高操作系统课程的教学质量，在广泛汲取国内外优秀教材和研究成果的基础上，笔者编写了本教材。在编写过程中，力求覆盖面广、内容新颖、重点突出。本书共分为 10 章，参考学时为 60～80 学时。各章内容简述如下。

第 1 章介绍操作系统的概念、功能、特征、发展历史和结构，并从操作系统的发展入手分析操作系统的发展方向，引入不同结构的操作系统的性能比较和分析，对流行的操作系统——Windows 系列和 UNIX/Linux 系统进行简要的介绍。

第 2 章介绍操作系统用户的环境、系统的生成与引导，以及操作系统提供的服务和接口。

第 3 章首先介绍进程与线程的基本概念，重点介绍进程的定义、状态等知识，并说明进程控制过程和方法；然后对处理器调度的实现和调度算法进行阐述，介绍在现代网络环境和实时系统中使用的多处理器调度和实时调度的一些基本原理；最后介绍 Linux 的进程管理。

第 4 章首先介绍进程同步和互斥的基本概念；然后阐述如何通过信号量机制和管程来实现进程的同步和互斥；最后介绍进程死锁的基本概念，并阐述如何预防和避免进程死锁的发生，以及死锁检测和接触的方法。

第 5 章首先介绍存储管理的基本概念和常见的存储管理方法，然后分别介绍各种内存管理技术的基本原理和地址映射、共享与保护等内容，最后介绍 Linux 的存储管理。

第 6 章介绍虚拟存储器的引入、概念和特征，对请求分页存储管理、请求分段存储管理的方法以及置换算法等内容进行了阐述，最后介绍抖动的概念和工作集理论。

第 7 章介绍文件系统中有关文件管理的基本概念、文件的逻辑结构和物理结构、文件存储空间的管理、文件的共享和保护、Linux 的文件系统等内容。

第 8 章介绍设备管理的基本概念、I/O 控制方式、中断技术、缓冲技术、设备分配和 I/O 软件管理、磁盘调度和管理、Linux 的设备管理等内容。

第 9 章首先介绍信息系统安全的概念；然后介绍计算机病毒的基本概念、常见的计算机病毒类型，以及如何预防和检测计算机病毒；接着阐述操作系统的安全机制，包括加密机制、认证机制、授权机制和审计机制，以及访问控制机制；最后介绍 Linux 的安全机制。

第 10 章介绍嵌入式系统的概念和硬件体系，说明了嵌入式操作系统的特征及其与个人计算机的区别，最后介绍常用的嵌入式操作系统。

本书可作为普通高校计算机类及其相关专业本科操作系统原理课程的教材，也可作为自学参考书和考研参考书。本书免费提供电子课件和习题答案，需要者请到 http://www.tupwk.com.cn/downpage 网站下载，也可以通过扫描下方的二维码获取。

本书由郑州大学的史苇杭、卫琳编写而成，其中第 1~4、10 章由史苇杭编写，第 5~9 章由卫琳编写。

感谢石磊教授在本书的编写和出版过程中提出的宝贵意见，让我们从中获益匪浅。参加本书编写的人员还有李翠霞、林楠、韩颖、王瑞娟、陈永霞、曹仰杰等，在此一并向他们表示诚挚的感谢。同时，对清华大学出版社有关同志深表谢意，谢谢他们在本书出版过程中付出的辛勤劳动。

本书的编写参阅了多种书籍和资料，主要的参考文献列于书后，在这里对这些文献的编著者表示诚挚的谢意。由于编者水平有限，书中难免有不当之处，恳请读者批评指正。我们的信箱是 992116@qq.com，电话是 010-62796045。

编者

2022 年 1 月

目　录

∞ 第 1 章 ∞

操作系统概论

本章学习目标

- 掌握操作系统的定义及其在计算机系统中的作用
- 掌握操作系统的特征与功能
- 了解操作系统的形成过程及发展趋势
- 掌握批处理系统、分时操作系统和实时操作系统的特点
- 了解操作系统的设计结构
- 了解常见操作系统的特点

本章概述

在现代计算机系统中，一个或多个处理器、主存储器、外存储器、网络接口以及多种不同的输入/输出设备共同协作，完成用户的各项需求。用户需求的响应过程是十分复杂和关键的，对编写和监督管理上述各种部件的程序员能力要求极高。为了将部分关键的操作封装，同时也达到简化程序员工作的目的，计算机体系中出现了操作系统这一软件层次。它能在管理并正确使用上述部件的同时，为程序员提供一个通用的、相对简单的、能够驱动硬件工作的软件接口。

本章首先从操作系统的定义、特征、功能、设计目标、性能指标等方面阐述操作系统的概念；然后从操作系统的发展入手分析操作系统的发展方向，由此引入不同结构的操作系统的性能比较和分析；最后对流行的操作系统——Windows 系列和 Unix/Linux 系统进行简要的介绍。

1.1 操作系统的概念

1.1.1 操作系统的定义

在现代计算机体系结构中，操作系统(Operating System，OS)起着至关重要的作用。操作系统在计算机体系结构中的位置如图 1-1 所示。操作系统位于硬件体系之上，在操作系统之上的则是各种应用程序。其中每个层次都可以再细分为更多的子层，例如，硬件层从底向上可分为物理设备、由各种寄存器和数据通道组成的微体系层以及主要由指令集组成的机器语言层，提供的是基本的计算资源。应用程序层的软件则通常是基于特定操作系统的、满足特定功能的、直接面向用户的软件，这些软件能够根据用户的具体需求申请特定资源，并按照应用程序规定的方法来使用这些资源。操作系统处于这两个层次之间，用来协调与控制应用程序对硬件的使用。

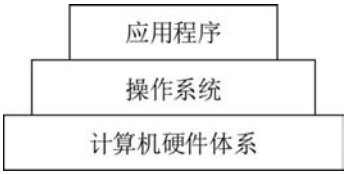

图 1-1　计算机体系结构

在当今社会，几乎人人都与操作系统打过交道，但要精确地给出操作系统的定义却并非易事。由于每个人看待操作系统的角度不同，使用操作系统的目的不同，看到的操作系统也就表现出不同的特征。下面从不同角度来探讨这个问题，并总结操作系统的定义。

1. 资源管理角度

从资源管理角度来看，操作系统可以被视为资源管理与分配器。操作系统是硬件之上的第一层软件，可以与硬件直接交流，对硬件资源具有最直接有效的控制和管理权限。同时，作为应用程序层的基础，操作系统又要为应用程序提供各种使用硬件的方法，即应用程序接口。因此，操作系统层次的软件应该能够直接操控各种计算机资源。

计算机的资源分为硬件资源与软件资源。硬件资源指的是作为计算机运算基础的所有物理设备，以及为方便用户所使用的鼠标、键盘、打印机等各种不同类型的外部设备。这类资源使用特定的电子信号来指挥，由电子工程师设计并提供相应的驱动程序。而在操作系统端则使用这些驱动程序以及特定的指令集来告知硬件如何工作，同时接收硬件发送回来的反馈数据与状态信息。根据硬件资源的功能不同，通常将其分为处理器、存储器、I/O 设备。相应地，操作系统也分别针对不同类型的硬件专门规划了处理器管理模块、存储器管理模块以及 I/O 设备管理模块。计算机的软件资源通常指的是各种程序与数据资源，它们以程序形式或各种不同类型的文件形式存放于外存上，操作系统要将其进行合理化存储，以保证空间利用率和读写效率之间的均衡和有效。

2. 用户角度

从用户角度来看，操作系统是用户与计算机硬件系统之间的接口，该接口在使用便捷性、资源利用率方面表现突出。

操作系统是一般用户可以接触到的最底层的软件，只有通过它所提供的各种接口，才能使用硬件系统。反过来说，操作系统将复杂的底层机器语言和操作屏蔽起来，并将常用操作和指令序列组合后以命令、系统调用函数、图形窗口等方式呈现给用户，帮助用户以更安全、高效、便捷的方式使用系统资源。因此，操作系统被称为一种人机接口。

在大型机和工作站端，这个接口除了能够帮助本机用户更方便地使用资源外，通常还肩负着为该用户与其他联机用户分配资源的重任。而分配资源的最重要原则就是确保 CPU 时间、内存和 I/O 设备得到最充分利用，以达到资源利用率最大化的目的。

3. 机器扩充角度

由计算机所完成的工作，无论繁简，总是可以分解为各个不同硬件的序列性动作，这些动作通过控制器命令完成。控制器命令有不同的种类，可以完成数据读写、磁头臂移动、磁道格式化、状态检测等不同工作。每条控制器命令均需要读写特定位置的数据，并从中分析所要求

动作和被操作数据等信息，然后按照分析结果完成命令动作，最后反馈新的状态信息和返回值到指定位置。显然，要求一般程序员使用控制器命令完成任务是不现实的，程序员需要的是高度抽象的、简单的操作方法。

基于上述原因，一个专门用来隐藏硬件的实际工作细节，并提供了一个可以读写的、简洁的命名文件视图的软件层次被引入计算机体系结构中，这就是操作系统。

综上所述，操作系统是一组管理与控制计算机软硬件资源并对各项任务进行合理化调度，且附加了各种便于用户操作的工具的软件层次。

1.1.2　操作系统的特征

操作系统虽然种类繁多，但其具有一些共同特征，这些特征也是操作系统这一软件层次与应用软件的区别所在。现代操作系统都具有并发性、共享性、虚拟性和异步性，其中并发性是操作系统最重要的特征，其他 3 个特性均基于并发性而存在。

1. 并发性

我们首先要区分两个概念：并发和并行。若在一个时间段内发生了一个以上的事件，则称这几个事件具有并发性；而并行性指的是多个事件在同一时刻发生。

在不同类型的操作系统中，并发性的含义有一定的区别。在单处理机系统中，每个特定时刻只能有一个程序在 CPU 中运行。但一个较长的时间段可以被分为多个小的时间碎片，这些时间碎片可以按照一定的原则发放给多个不同的程序，使得在这个时间段内有多个程序得到一定程度的执行。这些程序具有并发性，但不具有并行性。在多处理机系统中，每个特定时刻有多个 CPU 可以使用，则在这样的时刻，多个可以独立运行的程序就能够保证并行执行。

2. 共享性

操作系统中的共享，指的是多个并发执行的程序按照一定的规则共同使用操作系统所管理的软硬件资源。由于这些资源具有不同的使用要求，因此其共享方式也并非完全相同。操作系统所管理的软硬件资源按照使用方式，可以分为同时访问方式和互斥访问方式。

同时访问方式指的是在一段时间内允许多个程序并发访问。这里的“同时”指的是宏观上的一个“时间段”内的同时访问。从微观上而言，这些程序可能是顺序或者轮替地使用该资源。常见的使用同时访问方式进行共享的资源有磁盘、某些程序的公共缓冲区等。

互斥访问方式指的是在一段时间内只允许一个程序访问资源，而这类资源被称为临界资源。临界资源通常需要较长时间来处理一个不可被中断的任务，如打印机、某些程序的公用数据等。当临界资源空闲时，会对到达的第一个资源请求给出回应，而在处理该请求过程中，若有新的资源请求来到，则不予以理会。这种方法可以保证一个连续任务的无误处理，避免交叉打印或计算错误的发生。

3. 虚拟性

这里所谓的虚拟性并非指虚拟机，而是将计算机体系结构中的各种物理设备映射为多个逻辑设备。这种映射通常是利用时分复用或空分复用的方式实现的，被映射的物理设备有多种，如内存、外部设备、CPU 等。每个不同的设备，由于其工作模式不同，所使用的映射方法也是不同的。映射方法主要有时分复用和空分复用，下面分别对其进行介绍。

(1) 时分复用：使用时分复用方法实现虚拟性的设备主要有处理器和 I/O 设备。

虚拟处理器除了利用时分复用方法外，还利用多道程序设计技术保证多道程序并发执行。在一段时间内，CPU 将处理时间分割为多个时间片，并在不同时间片内完成对多个程序请求的响应，但每个提出请求的用户并不会感觉到有其他人和自己共用 CPU，而是感觉自己独占了资源。在当今硬件能力快速发展的时期，这种方法能够最大限度地发挥联网机器的效用，从而提高 CPU 的处理效率。

对 I/O 设备的虚拟化使用的是 SPOOLing 技术。该技术的核心思想是将物理设备的处理时间分成不同的片段，达到将一个物理设备映射为多个虚拟的逻辑设备的目的。操作系统将这些逻辑设备分别分派给不同用户提出的各项任务，设备将在这些不同的时间片段中为不同用户服务。而在用户任务被响应的过程中，每个用户均认为有多台同类设备同时满足多个用户的数据处理要求。

(2) 空分复用：使用空分复用方法实现虚拟性的设备主要有内存和磁盘。

虚拟存储器技术利用部分外存空间将较小的物理内存“扩充”为较大的虚拟内存。这种方法的核心思想就是仅将程序当前运行所需的数据和代码装入内存，当这个程序的一个相对独立的功能模块运行完成或暂时无法继续进行时，这个模块所对应的数据和代码将被暂存到外存的指定区域，其释放的内存空间将被重新分配给本程序的其他功能模块或其他程序的功能模块使用。使用这种方法可以确保对空间有较大要求的程序也能正常运行于小内存机器上。

磁盘虚拟化则体现为将磁盘分卷(分区)使用，并使用逻辑驱动器分别访问这些卷(区)，用户在查看自己的磁盘空间时将会看到多个不同名称的虚拟磁盘驱动器，并通过这些驱动器访问到不同的卷内的文件信息。

对于时分复用实现的虚拟设备，每台虚拟设备的平均工作速度等于或低于物理设备的 1/N，这里的 N 是该物理设备所对应的虚拟设备的数量。而对于空分复用实现的虚拟设备，每台虚拟设备平均能提供的空间必然也等于或低于其对应物理设备所拥有空间的 1/N。

4. 异步性

异步性指的是操作系统中的各个程序的推进次序无法预知。异步性的产生是由现代操作系统的并发性引起的。在并发执行的多个进程间，即使初始运行条件完全相同，但在每次运行时，各进程何时能够获得所需资源、在什么时刻等待哪些进程释放资源以及当前占有资源的进程何时释放资源等因素都是不确定的，因此用户无法预知各个进程的执行时长和当前进度，即进程是以人们不可预知的速度向前推进的，这被称为进程的异步性。操作系统的异步性是由进程异步导致的。

但由于进程的并发执行是多道程序设计的基本要求，因此操作系统应在不影响进程并发的前提下提供同步机制，使得在初始条件和运行环境相同的基础上多次运行相同进程时能获得相同结果。

异步性是现代操作系统的一个重要特征。

1.1.3 操作系统的功能

现代操作系统的主要任务就是维护一个优良的运行环境，以便多道程序能够有序地、高效地获得执行，而在运行的同时，还要尽可能地提高资源利用率和系统响应速度，并保证用户操

作的方便性。因此，操作系统的基本功能应包括处理器管理、存储器管理、设备管理和文件管理。此外，为了给用户提供一个统一、方便、有效使用系统能力的手段，现代操作系统还需要提供一个友好的人机接口。在互联网不断发展的今天，操作系统通常还具备基本的网络服务功能和信息安全防护等方面的支持。

1. 处理器管理功能

处理器是计算机软硬件体系的心脏，是制约整个计算机体系性能的最重要器件，因此处理器性能是否被充分发挥关系着整个计算机体系的性能。操作系统的主要任务之一就是合理有效地管理处理器，使其在现有环境下尽可能地发挥最大功效，提供更高的处理效率。

处理器的管理功能主要体现在创建、撤销进程，并按照一定的算法为其分配所需资源，同时还要管理和控制各用户的多个进程的协调运行，确保各个进程可以正确地通信。在多道程序的 OS 中，这些管理功能最终通过对进程的控制和管理来实现，而在具有线程机制的 OS 中，这些功能的实现还依赖于对线程的管理和控制。

2. 存储器管理功能

存储器是用来存放程序和数据的容器，是为计算机系统提供运作数据和具体指令序列的器件。操作系统所管理的存储器包括内存、外存等。存储器管理的主要任务就是将各种存储器件统一管理，保证多道程序的良好运行环境，同时还要兼顾内存利用率、逻辑上扩充内存的需求以及用户的感受，提供优良的控制、存取功能，为用户提供操控存储器的手段。

为实现上述要求，存储器管理应具有内存分配、内存回收、内存保护、地址映射和虚拟内存等功能。

(1) 内存分配。内存分配指的是为每道程序分配合适的内存空间，使其能在运行期间将运行所需数据放置在内存指定区域，以保证 CPU 能够顺利地获取指令并存取指定数据。分配内存空间时应尽量提高内存空间的利用率，减少不可用内存空间。此外还应能响应正在运行的程序发出的动态空间申请，以便满足新增指令和数据对新空间的需求。

内存分配分为静态和动态两种方式。静态分配方式指的是程序在装入内存时需要预估所需空间，一旦进入内存开始运行，就不能再申请新的空间，也不能将该程序所占空间“搬运”到其他位置。动态分配方式指的是尽管程序装入内存时申请了一定的空间，但在程序运行期间还可以为运行过程中所需的新的程序和数据再申请额外的空间，以满足程序空间动态增长的需要。

(2) 内存回收。内存回收指的是当程序运行完毕后，将各程序在装入内存时所分配的空间重新置为空闲分区，并交由 OS 统一管理，以备其他程序申请使用。

在内存的分配和回收过程中，为了记录当前内存使用和分配情况，OS 通常还要配置内存分配数据结构，以便为后期分配和回收提供依据。

(3) 内存保护。在多道程序环境中，为了保证每个用户的各个程序独立运行，不会相互影响，需要提供内存保护机制。该机制的主要任务就是确保每道程序都在自己的内存空间运行，决不允许任何程序访问或存取其他程序的非共享程序和数据。

内存保护机制的实现有多种方式，常见的一种处理方法是利用上下界限寄存器。这两个寄存器中存放的数值是当前正在运行程序的内存空间的起始地址和终止地址，每次当 CPU 要求访问某个地址的程序或数据时，OS 会先利用上下界限寄存器与之相比较，若在这两个界限内，

则可以正确访问，否则就拒绝此次的内存访问。这种方式可以确保在进程运行期间不会误访问无权限空间。

(4) 地址映射。在多道程序环境中，每个程序均使用自己的独立空间。这些空间分布于内存的不同位置，但每个程序员在编码时通常并不知道自己的程序进入内存后会被放置在什么位置，因此也不可能在程序中直接使用内存单元地址来操作所需的指令或数据。为了解决这个矛盾，当前的 OS 提供了地址映射机制。该机制的基本思想是将用户与内存分隔，即在程序员编码时，直接以 0 作为程序中出现的其他任何地址的初始位置，该位置被称为逻辑基址。当该程序被编译和连接过后，形成可装入的可执行文件。根据内存当前的使用情况，OS 会在可执行文件真正装入内存时为其分配合适大小的空闲空间，此空间的初始位置称为物理基址。当程序运行，CPU 需要查询某位置的数据或指令时只需给出相对于逻辑基址的偏移量，OS 会根据逻辑空间内容的分布情况自动将该逻辑地址转换为内存中对应的物理地址。地址映射功能需要硬件机构的协助，以保证数据的快速定位与存取。

(5) 虚拟内存。虚拟内存技术在当今的多数操作系统中均有涉及，其指的是利用特殊技术将磁盘的一部分空间实现较快的存取，从逻辑上扩充内存容量，使得用户感觉到内存的容量比物理内存实际所提供的空间要大。这种方式可以提高多道程序速度，提升系统吞吐量，获得更好的系统性能。而为了实现虚拟内存，只需配置简单的内存扩充机制即可。该机制的核心内容是请求调入功能和置换功能。

请求调入功能允许程序仅向内存装入保证启动的必需数据和指令，当程序在运行过程中若需要新的数据和程序时，先中断自身运行，并向 OS 提出调入请求，由其从磁盘将所需数据和指令调入内存，然后继续从被中断的地方执行。

置换功能指的是在 OS 将所需新数据或指令调入内存时，若发现内存空间不足，需要从现处于内存中的数据或程序中选择部分暂时不用的调出到磁盘上，腾出的空间则用来调入当前的急需数据。

3. 设备管理功能

在计算机系统中，外部设备的地位举足轻重，它是用户直接接触的对象，可用来进行人机交互。

设备管理的主要作用是使用统一的方式控制、管理和访问种类繁多的外部设备。设备管理功能主要体现在：接收、分析和处理用户提出的 I/O 请求，为用户分配所需 I/O 设备，同时还要做到尽量提高 CPU 和 I/O 设备利用率、I/O 处理效率，为用户提供操控 I/O 设备的便捷界面和手段。根据设备管理模块的功能要求，可以将其功能分为设备分配、缓冲管理、设备处理、虚拟设备等。

设备分配的主要功能是根据用户的 I/O 请求和系统的设备分配策略，从系统当前空闲资源中选择所需类型设备，并将其使用权限交付给用户。若 I/O 子系统中还包括通道和设备控制器，则设备分配还要负责选择空闲通道和控制器并交付用户使用。

缓冲管理的主要功能是合理组织 I/O 设备与 CPU 间的缓冲区，并提供获得和释放缓冲区的有效手段。在计算机系统中，凡是数据到达速度和离开速度存在差异的地方均可设置缓冲区以缓解速度矛盾，高速 CPU 和低速 I/O 设备之间就符合这个条件，并且由于这二者的利用率关系到整个系统的处理效率和响应速度，因此在 OS 中均为其设置缓冲层次。设置缓冲区的作用不

仅体现在缓解速度矛盾上，还可以显著改善系统性能。OS 常见的缓冲区机制有单缓冲机制、双缓冲机制、缓冲池机制等。

设备处理程序即通常所说的设备驱动程序，是 CPU 和 I/O 设备之间的通信程序。其工作过程为：当设备驱动程序接收到上层软件发送来的 I/O 请求时，要先检查其合法性，然后查看设备是否空闲、设备的工作方式等信息，接着按照要求的参数格式向设备控制器发送具体的 I/O 命令，指挥控制器启动设备按照顺序完成指定动作。为了保证通信，设备驱动程序还应能接收从控制器发来的中断请求，分析该中断请求的类型，然后启动处理该中断类型的相应中断处理程序，由其完成最终的处理过程。若是具有通道的 OS，设备处理程序中还要配备根据用户请求构造通道程序的功能。

虚拟设备功能是通过特殊的虚拟技术实现的，该技术可以将一台物理设备虚拟为多台逻辑设备，每个用户使用一台逻辑设备，即将独占的物理 I/O 设备交由多个用户共享使用。这种方法能够大大提高 I/O 速度，改善设备利用率，对每个用户而言也感觉自身具有一台独享的物理设备，改善了用户请求的响应感受。

4. 文件管理功能

在现代操作系统中，程序运行所需的代码和数据量十分庞大，而内存空间有限且无法长期保存信息，因此这些资源通常以文件形式存储在磁盘、光盘等外部介质上，只有在程序运行需要时才调入内存。为了能保证和核准用户可以正确使用这些资源，所有的 OS 都提供了文件管理机制。其主要功能就是管理外存上的静态文件，提供存取、共享和保护文件的手段，以方便用户使用，同时禁止无权限用户对他人资源的误访问或有权限用户对资源的误操作。文件管理机制还要能有效管理外存空闲区域，根据文件的大小为其分配和回收空闲区。为了满足用户对响应时间的要求，文件管理机制还应实现目录管理，以便快速地定位文件。

文件管理机制能有效保护文件安全，提高资源利用率，为用户提供快速检索和使用文件的手段，是 OS 不可或缺的组成部分。

5. 人机接口

为了更大程度地减少操作人员的次要工作、方便用户使用系统功能，在现代操作系统中无一例外地配置了用户界面，即所谓“用户与操作系统的接口”。该接口分为图形用户接口、命令接口和程序接口三类。

图形用户接口使用文字、图形和图像来形成文件，同时也使用各种图标将操作系统的多种类型的文件直观形象地表示出来。用户使用时只需用鼠标进行单双击操作即可完成全部操作。这种接口使得用户可以简单便捷地完成工作，即使是刚接触计算机的人员也能使用。该接口的实现使得计算机在社会生活的多个领域得到广泛普及，计算机变得非常简单易用，甚至非计算机专业的人员也可以利用计算机的高速处理和运算能力加速本专业的工作流程。因此，在 20 世纪最后的十年间，图形界面已经成为所有主流操作系统的必备模块。

命令接口是 OS 提供给用户的另外一种直接或间接控制自身工作的途径。用户使用命令接口向自身工作发送命令，控制工作运行。常见的命令接口有脱机命令接口和联机命令接口两种。脱机命令接口为批处理用户使用，由一组作业控制语言组成。作业控制语言可以用来定义作业说明书。由于当批处理作业运行时，用户不能直接与自身作业通信，只能由系统完成作业控制

和管理，此时系统的控制和干预方法均来自作业说明书。联机命令接口为联机用户使用，由一组键盘指令和命令解释程序组成。在使用联机命令接口时，需要利用键盘顺序输入多条指令。而命令解释程序每接收到一条指令就对其进行解释并执行。命令接口的好处在于直接驱动和控制相关设备，能得到更高的执行效率。

程序接口是出现于应用程序中的接口，其用来保证用户程序能获得操作系统服务，由一组系统调用组成。这些系统调用均是完成某些具体系统功能的子程序，对用户而言，这些系统调用表现为对应的库函数，当用户需要使用系统功能时只需像使用一般函数一样调用这些库函数即可。

1.1.4 操作系统的设计目标

现代操作系统的设计目标是有效性、方便性、开放性、可扩展性等特性。

有效性指的是 OS 应能有效地提高系统资源利用率和系统吞吐量。方便性指的是配置了 OS 的计算机应该更容易使用。这两个性质是操作系统最重要的设计目标。早期由于硬件的昂贵，设计人员更关注的是有效性，即使得系统中的资源利用率和系统吞吐量尽可能高。但随着硬件价格的不断降低以及计算机在各领域的广泛使用，在当今主流的微型计算机系统中，尤其是个人用户使用的微型计算机中，设计人员更关注的是 OS 的方便性，以便非计算机专业人员也能正确地使用计算机。

开放性指的是 OS 应遵循世界标准规范，如开放系统互连(Open Systems Interconnection，OSI)国际标准。这是因为在当前，由于 Internet 的快速发展，计算机操作系统早已从传统封闭的单机环境变为开放的多机环境。为了使不同厂家生产的计算机和设备能通过网络集成与共享，保证应用程序的可移植性和互操作性，OS 必须提供统一的开放环境，遵循相同或相似的国际标准，这就要求操作系统具有开放性。

可扩展性指的是 OS 应提供良好的系统结构，使新设备、新功能和新模块能方便地加载到当前系统中，同时也要提供修改老模块的可能，这种对系统软硬件组成以及功能的扩充保证称为可扩展性。随着当今新型设备、新界面样式、新功能的不断快速涌现，可扩展性也早已成为操作系统的重要设计目标。

1.1.5 操作系统的性能指标

操作系统性能的优劣显著地影响用户工作的效率和成本，而衡量其性能优劣的指标有系统吞吐量、资源利用率、响应速度等。

系统吞吐量指的是在单位时间内系统所能处理的数据量，该指标可以用来衡量系统的处理效率。

资源利用率指的是各类资源在单位时间内为用户服务的时间比例，可表明系统资源的分配利用是否合理。

响应速度指的是系统从接收到用户请求至完成请求处理、反馈响应信息这一完整过程的速度，其优劣大大影响到用户感受。

当操作系统采用较为合理的工作流程和资源管理方式时，即可改善资源的利用率，加速程序运行，缩短响应时间，增加多道程序度，提高系统吞吐量。

1.2　操作系统的形成和发展

操作系统作为一个系统软件，并非和计算机硬件同时问世，而是在长期的应用过程中逐步设计和改善，其设计不断汲取新的程序设计理念，跟随着用户的需求和硬件的变化不断发展而成。本节主要介绍操作系统的发展历程。

1.2.1　人工操作阶段

在真正意义上的操作系统尚未出现时，第一代计算机就已经面世了。其硬件采用数量庞大的真空管构造，体积巨大，占据了整个房间，而处理速度却不到当今个人计算机的百万分之一。当时并没有真正的程序设计语言，所有的程序设计均使用机器语言完成，需要使用插件板或穿孔卡片来记录程序，每个用户均需要使用人工操作的方式直接使用计算机硬件系统，将穿孔卡片装入专用的输入设备后，再手动启动将其输入计算机，最后再启动计算机处理数据。很明显，在一个用户使用计算机时，其他用户只能等待，且由于真空管寿命有限，经常会出现一个用户工作还没有完成就由于真空管烧毁而作废，这些都大大延误了计算机对用户工作的处理能力。

在这种人工操作方式中，出现了两个严重问题。第一，计算机由一个用户独占，除非该用户工作完成，下一个用户总是需要长时间等待。在计算机问世的最初阶段，甚至需要在墙上挂上计时表，以便预约计算机使用权。可见该方式并不能使程序员从烦琐而机械的重复性工作中解脱出来。第二，CPU 利用率低。在该方式中，安装卡片、启动输入设备进行输入时，CPU 并没有工作，而是等待数据输入，因此单位时间内 CPU 的利用率极低。

由人工操作的过程可以看出，该方法的资源利用率低，出现了严重的人机矛盾。

1.2.2　单道批处理

在 20 世纪 50 年代，功耗低、可靠性高的晶体管问世后，使用晶体管构造的计算机终于可以批量生产并销售。此时的晶体管计算机可以长时间运行，完成一些有用的工作，如科学和工程计算等。同时，设计人员、生产人员、操作人员、编程人员和维护人员的职业分工第一次明确。汇编语言和 FORTRAN 语言的出现和流行，也使很多编程人员开始使用它们来编写自己的工作程序，这些程序包含了一些特殊的程序，专用于完成批量作业的处理。这些程序就是现代操作系统的前身，被称为单道批处理系统。

在单道批处理系统尚未出现的年代，编程人员编好 FORTRAN 语言或汇编程序后，首先将其穿孔为一系列卡片，然后将这些卡片交给专业的操作人员，操作人员将卡片关联到系统上，等待数据处理完成并打印，最后再将打印结果送交编程人员。这个过程需要操作人员在不同的机器之间走动，且在每一个作业的完成过程中，人员走动的同时 CPU 是空闲的，因此造成了大量的时间浪费。为了减少 CPU 时间的浪费，开发了单道批处理系统。其工作过程为：操作人员利用一个特殊的输入设备将编程人员交付的多个作业卡片依序输入磁带上，当磁带满时或工作量累积到一定程度时，操作人员将磁带取下，安装到用于进行计算工作的大型机上。此时批处理系统启动，自动将磁带上的第一个作业读入处理，然后将结果写入输出磁带上。当前任务完

成后，批处理系统将选择下一个作业读入并处理，CPU 就这样顺序工作直到输入磁带上无作业需要处理才停止。CPU 停止后，操作人员将输出磁带装到一台专门用来进行输出打印工作的外围机上，输出外围机将顺序打印输出磁带上各作业的处理结果。这种一次性输入、处理和输出多个作业的方式可以保证输入外围机、输出外围机和 CPU 同时运行，具有顺序执行、自动运行和单道运行的特性，可以尽可能地将 CPU 的等待时间降低，在一定程度上缓解了人机矛盾，提高了 CPU 利用率和系统吞吐量。尽管如此，单道批处理系统的资源利用率仍然不尽如人意，现在已经很少使用。

1.2.3 多道批处理

20 世纪 60 年代，随着电子技术的发展，使用小规模集成电路制造的第三代计算机开始出现。第一台使用小规模集成电路的主流机型是 IBM 360，其与上一代晶体管计算机相比，在空间、能耗、处理速度等方面均有显著提高，很快就获得了成功。在 IBM 360 出现之前，大多数计算机厂商均使用两条不同的生产线专门生产两种类型的计算机，一类专用于科学和工程计算，另一类专用于磁带归档和打印。这种做法使得产品开发和维护的费用极为昂贵。IBM 为改变这种状况，开始在其新研制的 360 机上配置 OS/360 来解决这个问题。配置了 OS/360 的 IBM 360 机既可以用于科学计算，也可以用于商业计算，同时还可以处理磁盘归档和打印等事务。但由于不同用户的需求存在差异，导致 OS/360 需要集成各种不同处理程序，其结果就是其体积极其庞大且复杂度极高。同时大量的编程人员编写的汇编程序也将大量错误引入操作系统，IBM 需要不断地发布新的版本以更正这些错误。

尽管 OS/360 具有庞大的体积和大量的程序错误，但其最初的设计使其天生就可以用来满足大多数用户的要求，尤其是其引入的多道程序设计思想对后续的操作系统设计影响深远。多道程序设计思想被引入操作系统设计过程后，直接产生了如 OS/360 这样的多道批处理系统的出现。

在多道批处理系统中，所有用户的作业需要首先在外存排队等待装入，此时形成的作业队列被称为“后备队列”；接着由作业调度程序按照选定的调度算法从后备队列中选择一个或多个作业装入内存的不同分区，这些分区相互不重叠。同时在作业运行期间，仅能访问分配给自己的内存空间中的程序和数据，每次要求访问指定地址的信息时，操作系统要主动检查其是否超出本作业的内存空间，以防止越界访问和操作。这种方法为 OS 带来了显著的好处，主要表现为以下几点。

(1) 提高 CPU 利用率。在单道批处理系统中，内存中只有一个作业，虽然该作业的执行速度比较好，但在该作业执行 I/O 操作时 CPU 无事可做，只能消极等待 I/O 完成。引入多道程序设计思想后，一个作业需要转向 I/O 执行时，可以将 CPU 交给另一个作业使用，因此只要内存中作业数量达到一定程度，就可以保证 CPU 始终处于忙碌状态，其利用率甚至可以接近 100%。

(2) 提高内存和 I/O 设备利用率。很明显，多个程序共处内存可以尽量减少内存的空间浪费，且当有大作业到达时，只需配合一定的内存分配策略，也可以通过适当降低多道程序度的方式腾出足够空间交给大作业使用。对 I/O 设备而言，单道批处理系统一个时刻最多有一个设备被使用，但在多道批处理系统中，可能会有多个 I/O 设备同时被多个不同程序使用，则在单位时间内，每个设备的利用率都得到了提升。

(3) 增加系统吞吐量。在 CPU 和 I/O 设备都不断忙碌的情况下，单位时间内系统所处理的数据量必然会大幅增加，即系统吞吐量增加。

此外，作为批处理系统，多道批处理系统也需要累积一定数量的作业后一次性提交给系统处理。只是此时每个作业的处理效率比较高，作业读取间隔比较短，若仍由操作人员搬运磁带来进行作业输入显然会为处理速度带来不利影响，且第三代计算机本身也可以处理磁带归档和打印工作，因此在多道批处理系统中还引入了 SPOOLing(Simultaneous Peripheral Operating On-Line，假脱机)技术。该技术使用两个特殊的程序模拟单道批处理系统中的脱机外围机，专门用来实现外围机所做的磁带归档和打印工作，但由于其本质上是联机程序，因此该技术也被称为假脱机技术。

虽然多道批处理系统具有资源利用率高、系统吞吐量大等优点，但批处理系统本身所具有的平均周转时间(从作业进入系统到其完成任务退出系统所经历的平均时间)长、无交互能力等缺点，使得用户任务响应时间大大延长(与单道批处理相比)，且修改和调试程序均需要等待本作业所在的这一批作业处理完成才能进行，很不方便。

1.2.4　分时操作系统

分时操作系统(Time Sharing System)也属于第三代计算机时期出现的操作系统。该系统的出现起因于多数程序员对于多道批处理系统需要与他人争抢机时的痛苦感受，即每个编程人员将作业提交后需要等待很长的时间才能得到运行结果，且程序中很小的错误都会导致编译失败，编程人员简单修改后还需要重新预约机时、等待结果。因此大部分的编程人员还在怀念第一代计算机的使用方法，即一个用户可以独占一台机器一段比较长的时间，以便集中完成自己的工作。此外，编程人员也希望能随时与机器交互，以便对程序做出及时调整。

分时操作系统仍然基于多道程序设计思想，但其性能表现与多道批处理系统截然不同。其可供多个用户同时使用，能提高计算机资源利用率，是多用户共享计算机的最好办法，尤其是在多用户查询上表现最为突出。

实现分时操作系统需要及时接收和及时处理用户命令。其中，及时处理是人机交互能够实现的重要保障。为了能使用户输入命令后在指定期限内及时控制自己作业的运作，且由于在磁盘上的文件不能运行，无法保障随时交互，因此必须将每个用户的当前作业装入内存。另外，在实现分时操作系统时，还要注意每个用户不能长期独占 CPU 至自身运行完成。因此，分时操作系统将 CPU 处理时间进行分割，形成一个个特定大小的时间片段，这些片段被称为时间片。分时操作系统的每个用户在获取 CPU 控制权后，最长只能执行一个时间片长度，然后就必须让出 CPU，交给其他等待用户使用。

分时操作系统具有多路性、及时性、独立性和交互性特征。

(1) 多路性。多路性指的是多个用户终端同时连接在一台主机之上，系统按照时间片轮转的方式将 CPU 轮流交予各个用户进程使用。这种分时共享的方式，使得在一个时间段内，每个用户都感觉自己的作业被响应了。但从 CPU 本身而言，每个时刻还是只有一个用户程序使用。分时操作系统的多路性提高了资源利用率，降低了硬件成本，进一步挖掘了计算机系统的能力，使计算机在更广泛的领域得到更好的应用。

(2) 及时性。及时性指的是用户请求的响应时间要短。由于分时操作系统的设计初衷之一就是保证快速的人机交互，使用户可以及时控制作业运作，并能及时修改和调试程序，因此等

待运行结果的时间就要尽可能的短，此时的时间长短以用户感受为准。

(3) 独立性。独立性指的是每个用户均感觉自己独占了一台终端，且不知道其他用户的存在，即每个用户均独立运行自己的程序，相互之间不干扰。

(4) 交互性。交互性指的是用户使用终端与系统进行对话，以便向系统请求各种不同的系统服务，这些服务可以帮助用户完成不同类型的需求。

第一个真正的分时操作系统 CTSS 由麻省理工学院(MIT)开发完成，但直到第三代计算机广泛采用了硬件保护后，分时操作系统才真正流行开来。继 CTSS 之后，MIT 又联合贝尔实验室、通用电气公司共同开发了 MULTICS，用于支持数百名分时用户。尽管 MULTICS 在商业上并未取得辉煌成就，但其开发过程中涌现的优秀概念一直影响着现代操作系统的发展。最典型的案例是曾参与开发 MULTICS 的一位开发人员 Ken Thompson 在一台无人使用的 PDP-7 小型计算机上开发了一个简化的单用户版 MULTICS。该系统就是后来乃至现在流行于世的 UNIX 操作系统的前身。UNIX 系统以其稳定、开源、可裁剪等特性著称，在服务器操作系统领域的地位至今无人可比。1994 年，以芬兰学生 Linus Torvalds 为首的一个兴趣小组通过互联网组织在一起，共同开发了一个类似于 UNIX 的小型免费操作系统，该系统被命名为 Linux。Linux 系统以短小精干、功能强大、代码开源、永久免费等特征闻名于世，在商业应用和个人桌面领域都有亮眼表现。本章后面设有专门章节介绍该系统。

1.2.5 个人计算机操作系统

20 世纪 80 年代初，大规模集成电路出现后，计算机也随之出现了第四代——个人计算机。个人计算机又称微型计算机，指的是以极低廉的价格购买的计算机却能够做到原来由大型机才能完成的工作。

微型计算机上配置的操作系统被称为微机操作系统。最早的微型计算机操作系统 CP/M(Control Program for Microcomputer)诞生于 1974 年，由英特尔公司的顾问 Gary Kildall 等编写，是为英特尔新推出的第一代通用 CPU——8080 配置的专用操作系统。在随后的几年中，由 Kildall 组建的 Digital Research 公司重写了 CP/M，使其可以在 8080、Zilog Z80 以及使用其他 CPU 芯片的多种微型计算机上运行。由于具有较强的可适应性、可移植性以及简单易学等优点，CP/M 就此控制了 8 位微型计算机世界达 5 年之久。之后，随着用户需求的不断变化，微型计算机操作系统也产生了不同的变种，获得了长足的进展。根据其运行方式，可以将整个发展过程中出现的微型计算机操作系统分为以下几类。

1. 单用户单任务操作系统

单用户单任务操作系统，顾名思义，指的是在一段时间内只能有一位用户使用微型计算机，且这个用户一次只能运行一个用户程序。这是最早且最简单的操作系统，主要配置在早期的 8 位、16 位微型计算机上。

单用户单任务操作系统的典型代表除了上述的 CP/M 外，还有国人熟悉的 MS-DOS。MS-DOS 系统最早运行于 IBM PC 上，是由 Bill Gates 开发的 BASIC 解释器加上 Seattle Computer Products 公司开发的 DOS 操作系统合并、修改后形成的。其随着 IBM PC 捆绑销售，在商业上取得了巨大成功。1983 年，拥有 Inter 80286 CPU 的 IBM PC/AT 推出，升级后的 MS-DOS 2.0 版本具有许多来源于 UNIX 的先进功能。例如，由于对树状目录以及磁盘设备的支持，该系统很

快将 CP/M 彻底击溃。在后来的 80386、80486 中，MS-DOS 不断升级，加入新的先进功能，受到大多数用户的欢迎，成为事实上的 16 位单用户单任务操作系统标准和典型代表。

2. 单用户多任务操作系统

单用户多任务系统仍然要求一段时间内由一个用户独享计算机，但这个用户可以合理利用这段机时，一次运行多个任务，使它们并发执行。这种做法通过提高多道程序度来增加系统吞吐量和资源利用率，缩短用户任务的响应时间，从而大幅提高系统性能。该类型操作系统的典型代表是 Windows 系列。

在 Windows 操作系统出现之前，苹果公司推出的个人计算机 Macintosh(简称 Mac)的操作系统——mac OS 已经取得了巨大的成功。其第一个在商业 OS 版本中采用了 GUI(Graphical User Interface，图形用户界面)，即使用窗口、图标、菜单、鼠标等进行用户任务的操作控制，适合于那些没有计算机知识的非专业用户使用。正是 Mac 的这一特性，使得微软公司的决策层意识到 GUI 在计算机推广过程中的重要性，因此在构建 MS-DOS 的后继产品时，微软公司将一个图形环境软件层次附加在 MS-DOS 上，形成了早期的 Windows，该系列操作系统的核心还是 MS-DOS，图形界面只是附加在其上的一个 Shell 层。这种运作方式持续了 10 年，直到 1995 年微软发布一个独立的 Windows 版本——Windows 95 后才得到改观。Windows 95 仅使用 MS-DOS 引导系统和兼容老的 16 位程序，同时还添加了对 Internet 的支持。此后的改进版 Windows 98 将微软的 Internet 浏览器 IE 集成到系统中，方便了用户上网检索，同时还集成了多媒体功能的支持。在此基础上，微软不断推出基于 32 位、64 位的新版操作系统，它们通常具有家用和商业工作站两种不同版本。目前最新的 Windows 系列产品是 Windows 11。与其前辈相比，该系统具有更个性化的界面，系统运行效率更快，对多媒体体验给予了更多关注，能够为用户带来更为舒适的感受。

3. 多用户多任务操作系统

多用户多任务操作系统允许多个用户同时登录主机，但每个用户要使用自己的终端来共享系统资源；在每个终端上，各个用户可以同时运行多个并发任务，以进一步提高系统吞吐量和资源利用率。除微型计算机外，在大、中、小型计算机上通常都要配置多用户多任务操作系统，以保证多个用户的联机共享需求。甚至在微型计算机上，也有很多用户选择使用此类系统，作为个人用户而言，选择最多的还是 UNIX 及其各种变种。

UNIX 在工作站和其他类似网络服务器这样的高端计算机上影响最为广泛，功能也十分强大。此外，在采用了 RISC 芯片的计算机上 UNIX 的应用也十分普遍。关于 UNIX 操作系统及其最为流行变种 Linux 将在第 1.4.2 节进行介绍。

1.2.6　实时操作系统

在操作系统的发展历程中，出现了多种多样的实时系统，在工业应用领域，实时操作系统更为常见。这类系统的特征是将时间作为关键参数，即系统应能及时获取用户请求，并在指定时间内开始或完成规定任务，同时还要保证所有任务协调一致地工作。实时操作系统通常被用于进行实时控制或实时信息处理。

在现代化工业现场，许多生产过程的控制均需要依赖计算机对数据的快速组织处理能力，由硬件采样装置采集上来的样本值被及时(或即时)传送到控制主机中，由主机根据事先制定的

控制规则自动指定相关硬件执行预定操作，以完成产品的加工。根据不同产品的生产过程和品质要求、形状等特性，被采集的数据可以是温度、压力、姿态、状态值等，唯一的共同点就是这些数据都用来保证产品的质量。此外，由于计算机处理能力的不断提升，实时控制过程也不断缩短，使得产量不断提高。正是由于实时系统的准确、及时、高效等特性，使其当前在工业、军事等领域得到长足进展，推动了科技的进步。常见的实时控制系统有数控系统、工控系统、制导系统以及多种智能家电。

除了实现控制，实时系统也用来进行实时信息处理，此类系统通常所管理的数据均放在服务器上，由多个终端通过网络关联，服务器接收各终端发来的远程服务请求，分析请求后再对信息进行检索和处理，再将处理结果返回给提出申请的用户。为保证响应效率，这个过程中的每一步都要尽可能地快速、准确。常见的实时信息系统有票务系统、检索系统等。

无论是哪种实时系统，其所处理的任务都具有一定的紧迫性，且与某些外部设备相关，甚至若有一个步骤没有达到时间要求，后续所有工作都会出错。因此在设计实时系统时，需要综合考虑不同类别实时任务的特性。按照不同的划分方法，实时任务有不同的分类。

(1) 按周期性对实时任务进行分类。按周期性分类，可以将实时任务分为周期性实时任务和非周期性实时任务。周期性实时任务通常由外部设备按照设备周期定期发送信号来启动，主机收到信号后为该设备进行某些特定工作的执行。非周期性实时任务则不期待周期性信号的出现，而是根据任务对截止时间的要求以及当前系统中资源使用情况选择合适的时机启动，该时机必须能保证任务满足截止时间要求。

(2) 按对截止时间要求的严格程度对实时任务进行分类。实时系统中的截止时间分为开始截止时间和完成截止时间两种。前者要求实时任务在指定时刻(或之前)必须开始运作，后者要求实时任务在指定时刻(或之前)必须结束。按照实时任务对开始截止时间或完成截止时间要求的严格程度，可以将实时任务划分为硬实时任务和软实时任务两种。硬实时任务指的是该任务必须在指定截止时间(或之前)开始或完成。通常直接控制设备姿态和动作的实时任务需要严格遵守时间要求，否则可能导致后续步骤无法执行，因此此类任务通常为硬实时。软实时任务则要求不是这么严格，偶尔可以超出时限，对系统正确性和安全性不会有太大的影响。常见的软实时任务如检索信息等，通常是以人类的耐心为标准，虽然有时间要求，但并非十分严格，因此可以偶尔超出。

1.2.7 网络操作系统

计算机发展到如今，系统在进行本地计算时，经常需要通过网络存取位于异地计算机上的数据，因此现在的操作系统都要求支持单个计算机通过局域网和广域网互联的功能。随着网络的不断普及，逐渐形成了一些新的操作系统类型，其中最重要的是网络操作系统和分布式操作系统。

网络操作系统(Network Operating System，NOS)是网络用户和网络之间的接口，除实现通用操作系统功能外，还需要管理网络中的共享资源，实现用户之间的通信，同时还要向用户提供多种系统服务。因此，网络操作系统应具有数据通信、资源共享、网络管理等基本功能。

网络操作系统的数据通信功能表现在几个方面。第一，要能建立与解除除应用层外的所有各网络层次间为保障数据通信所设立的各种连接，例如物理层的物理连接、数据链路层的相邻

节点间的无差错信息传输连接等。第二，要能正确拆分和组装报文。这一功能主要针对长消息操作，即将较长的报文在传输层分割并分别打包为网络层上的分组格式，再依序将多个分组发送给对方网络层，在接收方计算机上的对应网络操作系统要能根据特殊序号将所有分组组合为与初始信息相同的正确报文。第三，要能实现传输控制。传输报文时，网络操作系统要能根据发出报文的用户信息为报文制作报头，用以记录与发送者和接收者相关的信息，以保证传输控制的正确进行。第四，要能做到流量控制。由于在分组交换网中采用“存储转发”形式进行信息传输，为保证不丢失数据，应确保路由线路上的缓冲区能及时容纳新信息，其实现方式就是控制进入路由线路的信息流量。第五，要提供差错的检测与纠正功能。为尽量降低数据错误，在真正使用数据前应进行检测，若发现错误也要能加以纠正。

网络操作系统资源共享的范畴通常包括对硬盘空间的共享和网络打印功能的共享等。对硬盘空间，既可以使用虚拟软盘方式将服务器硬盘分区后分配给各用户，也可以使用文件服务方式要求系统将服务器的文件系统中特定目录或文件的存取权限交付给用户。值得一提的是，无论使用哪种方式，用户实际所利用的空间均是位于服务器的硬盘之中。另一种常见的共享资源是网络打印机。网络操作系统使用 SPOOLing 技术将网络中的打印机共享给工作站或其他所有的计算机，每个工作站或服务器发送给打印机的任务将在打印机所在地进行排队等待，直到全部打印工作完成。网络资源的共享可以尽量提高资源利用率，降低成本。

网络操作系统中实现网络管理的软件负责监视网络及其组件的运行情况。其向不同设备不定时发送特定信息，并接收和分析响应信息，以确定该设备甚至整个网络是否正常运作。

网络操作系统经过多年发展，其基本设计目前已经成熟。

1.2.8　分布式操作系统

与计算机网络操作系统一样，分布式操作系统也是随着互联网的发展而出现的一种新型系统软件。如 1.2.7 节所述，网络操作系统是在单机上部署的，其运行性能取决于本地计算机。当这样的一台主机希望访问位于网络上的另一台计算机的资源时，必须指明要求访问的是哪个站点上的哪一台计算机，为了保证访问指令被正确解析，发出的请求还需要以目标机的指令、数据格式来实现资源共享。可以看出，这种方法在实现多主机协作时很烦琐且容易出现错误。

为正确完成多主机协作，分布式操作系统的理念逐渐流行起来。分布式操作系统是一个逻辑上的全局操作系统，由若干独立的计算机构成，这些不同的处理器通过互联网构成一个统一的系统，在该系统中使用了分布式计算结构，将原来单机系统 CPU 的任务分散给不同的处理器，这些处理器可以用来实现不同的功能，它们协调工作，共享系统软硬件资源，共同控制分布式程序(进程)的运行。分布式操作系统能够给用户提供统一的界面、标准的接口，以便利用系统的各种资源实现所需要的全部操作。值得注意的是，组成分布式系统的各个主机体系(包括处理器、存储器、外部设备)既可以独立处理本地任务，又可以合作完成大型工作。

1. 分布式操作系统和网络操作系统的区别

分布式操作系统和网络操作系统虽然都以计算机网络作为物理基础，也使用了分散的多个处理器，但其在系统透明性和系统耦合度等方面存在本质区别。

在系统透明性方面，分布式操作系统中不同主机的用户可以看到统一的界面，每个用户均认为整个系统运行于本地计算机，当用户要求使用某资源时，只需使用本地机的操作指令声明即可，分布式操作系统将自动解析用户申请，选择相应位置的主机运行指定操作，并利用相应位置的指定类型资源。但对用户来说，这些资源所在位置和操作执行的位置均无须了解，系统对用户而言是透明的。在网络操作系统中，用户申请的资源所在位置需要显示给出，用户需要对整个网络布局有一定的认识，系统对用户而言是不透明的。

在系统耦合度方面，在部署了分布式操作系统的计算机体系中，计算机网络和各独立主机是其物理基础，主机之间的通信仍要经过通信链路进行信息交换。其除了具有一般计算机网络的模块性、并行性、通信性等特征外，还有进一步的发展。例如，在常规网络中，并行性仅意味着各主机独立，但在分布式操作系统中，并行性意味着在完成某些大型工作时，多个主机需要合作，这是因为分布式系统不再是一个物理上的松散耦合系统，而是逻辑上紧密耦合的系统。在网络操作系统中，各主机均有自己的一套处理机制，仅仅依靠通信链路上的消息传递为用户提供一些服务，逻辑上并未构成一个整体，因此网络操作系统是逻辑上松散耦合的系统。

2. 分布式操作系统的特性

分布式操作系统是全新的操作系统设计思想，具有多机合作和健壮性等特性。

多机合作指的是多个处理器共同协作，由操作系统自动对任务进行分配和协调。分布式操作系统具有多机合作功能，使其具有响应时间短、系统吞吐量高、可靠性高等优点。

分布式操作系统具有良好的健壮性，当系统中的一个甚至多个处理器出现故障时，剩余的处理器仍能自动重新构成一个新的基于较少处理器的操作系统，该系统仍能继续工作，甚至在配置了冗余处理器的分布式系统中，重构的新系统可以继续失效部分的全部工作。一旦故障排除，系统将自动恢复为原来的状态。这目前仍然是一个热门的研究领域，也是未来操作系统的发展方向之一。

1.2.9 操作系统的进一步发展

纵观操作系统的发展轨迹，可以看到操作系统除了在向分布式发展外，还有两个比较热门的领域——集群式系统和嵌入式系统。

1. 集群式系统

在操作系统的热门研究领域，集群式系统是一个尚未明确定义的概念，但其基本思想可以表述为由两个或多个独立的系统单元组成的大型系统，这里的单元可以是单处理器系统，也可以是多处理器系统。多处理器系统，指的是由通过通信信道紧密通信的多个处理器组成的系统，其分为对称多处理器系统(系统中每个处理器均安装有一个操作系统的副本，它们相互通信，共享总线、时钟等系统资源，在系统中的地位相同)和非对称多处理器系统(系统中每个处理器的地位不同，有一个主处理器和多个协处理器，主处理器主控全局，协处理器向主处理器要任务或执行预先设定的任务)。

由于集群式系统允许使用多个相对独立的多处理器系统作为自身的构成单元，且组成集群系统的多个个人机硬件成本之和比大型机硬件的成本要低廉许多，使得集群式系统可以具有较高的性价比。例如，弗吉尼亚理工大学(Virginia Polytechnic Institute and State University)于 2003

年开发出的 System X 就是一个典型范例，它由 1100 台 Mac G5 汇总而成，每秒钟可以执行 10 万亿次浮点操作，成本却只需将近 500 万美元，比传统大型机便宜许多。

由于集群系统的特殊结构，它还应能提供优秀的容错能力，以便多个单元出现故障时，系统仍能以较好的性能持续运行。同时还要能及时检测到故障的发生，并恢复故障节点的运行。

最流行的集群式系统的实例是 Beowulf 系统，它由一组基本相同的计算机构成，每一个节点上都运行 Linux 系统和一组适用于 Linux 内核的软件包，主要应用于科学计算、大任务量计算等环境。

2. 嵌入式系统

嵌入式系统是随着各种数字化设备的流行而出现的。在实现专有目的的数字化设备的设计过程中，人们希望将计算机的快速处理能力加入进来，因此在数字化设备中提供了类似微型机的硬件构成，并将嵌入式系统植入这些硬件中。作为一种特殊的系统软件，嵌入式系统为其各种用户级嵌入式软件提供支持，同时控制整个系统的各项操作，合理管理和分配系统资源。它除了具有操作系统的基本功能外，还具有实时性、微型化、可裁剪、高可靠性和高可移植性等特点。

(1) 实时性。常见的嵌入式系统均出现在为用户提供快速计算和即时响应的小型计算机中，如掌上计算机、PDA 等。这种交互需求必然要求其中的嵌入式系统具有实时性，以便保证反馈信息的及时和有效。

(2) 微型化。在数字化设备中，由于其体积限制和实时性要求，不可能提供如微型机那样大的内外存空间，因此为了能在有限空间中运行，嵌入式系统必然不能像大型机甚至微型机系统那样庞大。

(3) 可裁剪。大部分的数字化设备用途比较单一，在系统运行过程中其处理的计算也相对简单，且不同用户对设备功能的要求也不同，因此嵌入式系统应该能根据用户需求进行自行定制裁剪，以便使用最小的软件集合实现最符合用户需求的系统。同时，嵌入式系统的可裁剪性也将能满足用户要求的系统控制在最小体积。

(4) 高可靠性。可靠性是所有操作系统的基本要求，但和普通的实时系统一样，嵌入式系统对可靠性的要求比较高。例如，一个用于控制数控机床的嵌入式系统直接控制刀片的姿态和行动轨迹，若有一点失误，将会导致材料的浪费，甚至引发严重的质量问题，因此其可靠性是系统设计和测试过程中一定要重点关注的。

(5) 高可移植性。嵌入式系统通常会在一系列功能相似的嵌入式硬件上运行。为满足不同硬件或不同应用场合的特殊需求，嵌入式系统还应该能在简单修改后就可以在不同的环境中正确有效地运行，即嵌入式系统应具有可移植性，不依赖于特定硬件。

最为常见的嵌入式系统有 TinyOS、μC/OS-III 和 Linux。尤其是在复杂嵌入式应用领域，Linux 系统由于具有开源、外围软件多等特点而表现抢眼。

1.3 操作系统的结构

早期的操作系统仅用来完成很简单的工作，因此其规模很小，只需要一个熟练的程序员花

费一小段时间就可以完成编写。但随着操作系统功能的不断增加，开发一个完整的操作系统对个人而言越来越难。在此背景下，人们逐渐开始使用工程化的方法开发操作系统，并由此产生了“软件工程学”。

使用软件工程思想设计操作系统时，除了要满足第 1.1.4 节所述的操作系统设计目标外，还要根据用户需求分析采用哪种形式的操作系统结构更适用。在操作系统设计方法的发展历程中，出现了多种不同的操作系统结构，其中比较有代表性的有整体结构、分层结构、虚拟机结构、外核结构和客户机/服务器结构。

1.3.1 整体结构

整体结构又称为模块化结构，它采用结构化程序设计技术，将复杂的 OS 需求分解后根据相关性分类，每个类别使用一个或几个模块实现。使用这种结构的 OS 中，各模块分别具有独特的某方面管理功能，如进程管理、存储器管理等，且相互之间使用规定好的接口通信。接着再将各模块进一步划分为子模块，并定义通信接口。子模块还可以进一步细分为更小的模块，这种设计方法被称为模块—接口法。系统中的模块按照其基本调用需求可以分为主程序、服务程序和公用程序三类。主程序用来声明要调用的服务程序，服务程序则用来执行系统调用，而一个服务程序必须使用多个实现具体操作的公用程序的组合才能完成。图 1-2 所示为一个简单的整体结构操作系统的结构模型。

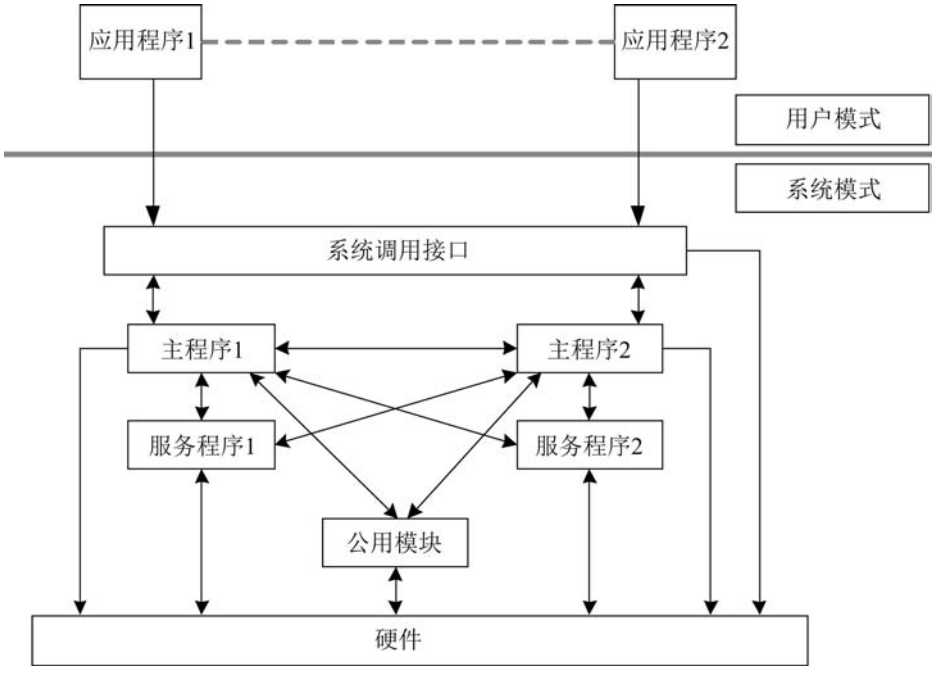

图 1-2　整体结构操作系统示意图

整体结构的操作系统具有突出的优点。使用模块—接口法开发的 OS，各个模块间可以通过预定义的接口相互调用，因此系统中已有的功能可以直接被其他新加入的模块使用。由于各模块是按照功能划分，其内部数据关联性(内聚度)高，对外表现出较强的独立性(耦合度)，可以独立编译形成可装入代码，因此新加入的系统功能可以独立设计、编码和调试后再与原系统连

接形成新的整体。同时，在系统最初设计时，也可以让多个小组分别开发相对独立的各个模块，以缩短系统开发周期。

但使用模块—接口法设计的整体结构操作系统仍然在某些方面具有缺陷。这些缺陷通常起因于设计过程，因此也是整体结构操作系统不易克服的严重缺点。例如，在设计过程中对接口的规定经常在实际应用过程中发生无法满足需求的状况。由于各模块的设计是一个小组同时进行的，因此每个模块的设计决策主要来自本小组内部需求的考虑，但从整个系统考察，很有可能无法找到可靠的决定顺序。因为前驱决策会影响后继决策，这种无序性将会导致设计的基础无法保证可靠，而基于不可靠的设计所开发的 OS 必然不会有很好的可靠性。

1.3.2　分层结构

由于整体结构 OS 具有内聚度低、耦合性高、可靠性差、可扩展性小等缺点，而操作系统的规模则是在不断扩大，整体结构 OS 逐渐无法满足人们的需求，因此设计新的操作系统结构成为必然。在这样的背景下，首先被广泛采纳的设计方法是分层法，形成了分层结构的 OS。这种方法将整体结构 OS 中的各个模块按照决定次序分层，即每层都是在其下层的基础上构建的。这种构建方法的基本思想是在裸机之上，根据模块功能与系统硬件的相关性以及各模块之间的调用关系，构建多个层次的软件，最终形成完备的 OS。分层结构实现过程通常采用自底向上的方法进行，以保证系统模块按照设计过程中的决策次序构建。

使用自底向上的方法构建出的各个层次具有可靠的基础，每一层仅能使用其底层(包括紧邻底层和更低的层次)所提供的功能和服务，且在开发过程中，先开发最底层，调试完成后即形成与上层无关的可靠基础。接着在其上构建第二层软件，由于已知其下层软件是正确的，因此凡是调试和测试过程中出现了问题，可以确知是由第二层软件造成的，因此只需修改第二层软件即可。按照这种方法继续，直到形成完整的 OS 为止。按这种方法构造的操作系统具有层次特性，每个层次由多个模块构成，且只依赖于其紧邻下层。

典型的分层结构 OS 为 MULTICS，其由一系列同心环构成，内层环比外层环具有更高的级别。当外层环要调用内层环的过程时，必须执行一条等价于系统调用的 TRAP 指令。另一个著名的分层结构 OS 实例是 E.W.Dijkstra 和他的学生开发的 THE 系统(1968)，在这个系统中第一次提出了操作系统分层设计的思想，第一次出现了分层结构的 OS。该系统是为荷兰的 Electrologica X8 计算机配置的批处理系统，共分为 6 层，如表 1-1 所示。其中，0 号层提供了基本的多道程序环境，当发生中断或时间片用完时，处理器将在该层进行相应分配，其上层的进程无须知道处理机分配的具体细节；1 号层的各模块通常用来完成内存分配，提供页面置换功能，以保证该层所管理的内存和对换区中的页面能正确命中；2 号层处理进程与操作员之间的通信，以保证操作员的意图可以正确地被系统获知和处理；3 号层管理 I/O 设备和相关的缓冲区，为上层提供统一的使用外部设备的接口，方便用户使用各种不同类型的外设功能，同时保证设备控制对用户透明；4 号层中的各用户程序无须了解 I/O、进程通信、存储器和处理器的工作细节，只需使用特定接口将自己的申请发送给紧邻下层即可；5 号层是操作人员最终所在的层次。

表 1-1　THE 系统结构

层号	功能
5	操作员
4	用户程序
3	I/O 管理
2	操作员与进程通信
1	存储器和磁盘管理
0	处理器分配和多道程序设计

分层结构的操作系统基于正确的层开发方式便于保证系统正确性，同时层次化的设计方式便于进行功能扩充和系统维护。但每个层次都仅依赖其紧邻下层的特性使得各层间都要定义通信机制，这样当一个用户请求需要使用较低层次的功能时，需要多次穿越层边界，这样的做法会增加通信开销，导致系统效率降低。

1.3.3　虚拟机结构

虚拟机结构 OS 最初是为了满足用户对分时系统的需求而出现的。第一个真正意义上的虚拟机结构 OS 是 VM/370，其设计初衷是实现两个功能：提供多道程序环境和实现一个比裸机更方便的可扩展的虚拟计算机。在部署 VM/370 的计算机体系中，裸机和虚拟机被彻底隔离开来。

VM/370 的核心程序为虚拟机监控器(Virtual Machine Monitor)，其运行于裸机之上并提供多道程序功能。该系统向上层提供多个对裸机硬件精确复制的虚拟机，这些复制品均包含核心态、用户态、I/O 处理、中断以及其他真实机器所应该具有的全部功能。正因如此，凡是能在一台物理裸机上运行的操作系统均可以出现在一个特定虚拟机上，分配给各用户的不同虚拟机上可以随用户的个人爱好和操作习惯不同而采用不同的操作系统。因此经常会出现一台裸机上的各个虚拟机中运行类型各不相同的操作系统的现象。在这些虚拟机中，除了用来实现批处理和事务处理的 OS/360 的后续版本外，还有用来保证分时共享的单用户、交互式系统——会话监控系统(Conversational Monitor System，CMS)。

图 1-3 所示为配有 CMS 的 VM/370 的结构。当一个 CMS 程序执行系统调用时，该调用被陷入其虚拟机的操作系统上，而不是 VM/370 上，在用户看来就是直接在自己独享的一台裸机上工作。接着再由 CMS 发出常规的设备 I/O 指令来响应系统调用，这些 I/O 指令被陷入 VM/370 后，由 VM/370 实际执行并完成指令要求。

上述方式将多道程序设计功能与虚拟机的扩展能力彻底隔离，使得每个部分的设计和实现都更加简单和灵活，更容易维护。

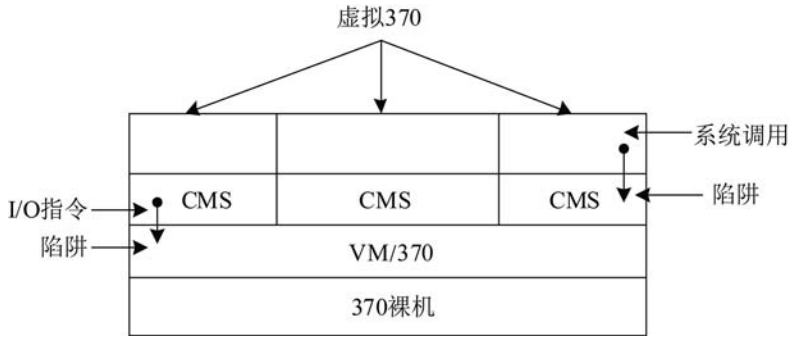

图 1-3　配有 CMS 的 VM/370 的结构

1.3.4　外核结构

在虚拟机结构的操作系统中，由于所有用户使用的虚拟机都是对整个物理裸机的精确复制，因此所有用户见到的可用空间均是从 0 到最大编号。这种使用方法要求虚拟机监控器必须设置和维护一张地址重映射表，以便找到用户的实际文件所在地。

为了使各虚拟机相互独立，麻省理工学院的研究人员以虚拟机结构 OS 为基础，构造了一个外核结构的操作系统。该系统为每个虚拟机分配连续的指定大小空间，作为其自身的独立空间使用。该系统的核心程序称为外核(Exokernel)，其在核心态运行，为虚拟机分配资源，并对虚拟机发送来的资源访问申请进行检查，以保证没有任何虚拟机能访问其他虚拟机(用户)的空间。这种方法可以将每个用户层虚拟机限定在已分配给它的指定资源环境中，保证各虚拟机的独立性。

使用外核机制可以减少映射次数，在虚拟机监控器中无须设置地址重映射表，只需要在外核程序中记录分配给各虚拟机的资源情况即可。此外，该机制还可以将外核内的多道程序设计环境与用户空间的操作系统代码有效隔离，且外核处理虚拟机冲突的开销并不大。

1.3.5　客户机/服务器结构

虚拟机结构和外核结构都试图把传统操作系统的大部分功能转移到 CMS 中，VM/370 做得已经相当不错，但却仍然是一个十分复杂的系统软件。若想保持系统处理效率，就不可能模拟过多的虚拟 370。因此，现代操作系统普遍采用的方法是将操作系统的大部分功能尽量从核心态中移出，只将最基本的操作组成一个很小的微内核，这就形成了所谓客户机/服务器(C/S)结构的操作系统。

客户机/服务器结构的操作系统是随着计算机网络的发展和流行而出现的，在 20 世纪 90 年代已经风靡全球。这种结构的操作系统对支持分布式操作系统具有天然的优势。它将处理本地业务的用户进程放在具有一定独立处理能力的客户机上，形成客户机进程；而把网络之上所有用户同时共享的信息以及完成客户机与服务器通信功能的微内核放在服务器上，形成服务器进程。这样当一个用户提出访问系统资源申请时，客户机进程先发送请求给服务器进程，由后者将请求转发给能完成指定服务的客户机，待其完成要求工作后将结果发送给服务器进程，再由服务器进程将结果回送给发出申请的客户机进程。如图 1-4 所示，一个用户请求访问某文件的数据，该文件由网络文件系统管理，需要使用文件服务进程，因此用户所在的终端上的客户机

进程向核心态的微内核发送一个请求，微内核中的服务器进程接收申请并分析，然后将其传送给其他客户机上的文件服务进程进行数据操作。当数据处理完成后，结果还需要通过微内核传送给发出申请的客户机进程，客户机进程再将最终处理结果反馈给用户。

图 1-4　客户机/服务器结构

客户机/服务器结构的操作系统具有不同于传统集中式 OS 的一系列独特优点，使其在网络时代大为流行。首先，该系统的数据可以进行分布式处理和存储。客户机本身均具有一定的处理能力，部分数据处理和存储工作可由本地客户机完成，减少了服务器的任务量。其次，对于重要数据，可以将其放在受到严密保护的服务器所在的局域网内集中管理，以便保证数据安全。再次，C/S 结构有较好的灵活性和可扩充性，客户机/服务器机类型可选范围很大。最后，易于修改用户程序。对客户机的修改和增删很方便，甚至可以由用户自行进行。但在整个系统中，性能的优劣取决于服务器的处理能力，一旦服务器出现问题，整个系统将不可用，因此需要配置多个服务器，以改善和尽量避免瓶颈的产生。

1.4　常用操作系统简介

1.4.1　Windows 系列操作系统

Windows 系列操作系统是当今应用范围最广的操作系统，尤其是在个人桌面系统领域，至今无比肩而立者。众所周知，Windows 系列操作系统是微软公司的主要产品，目前已经发展到了 Windows 11。

Windows 虽然以图形界面著称于世，但不是最早的图形界面操作系统。1970 年，美国 Xerox 公司成立了一个专门从事 LAN、激光打印机、图形用户接口和面向对象技术研究的机构 PARC，该机构于 1981 年推出世界上第一个具有 GUI 的商用系统：star 8010。但由于各种原因，它并未获得商业上的成功。

而苹果公司的创始人之一 Steve Jobs 在参观 PARC 研究中心后意识到 GUI 的应用前景，开始着手进行基于自己公司微机的 GUI 系统。几年后，第一个成功的商用 GUI 系统 Apple Macintosh 推向市场，获得了广泛好评。但其硬件依赖性使得该系统在逐渐占据了市场主导地位的基于 Intel x86 的微机之上无法运行。微软公司在这里发现了商机，开始开发自己的具有 GUI 的 Windows 系统，并在开发之初就致力于将该系统打造为基于 Intel x86 的微机系统上的标准 GUI 操作系统。

1985 年和 1987 年，微软公司以 DOS 系统为基础，加入一个简单的 GUI 层次，分别形成

了 Windows 1.03 版和 Windows 2.0 版，这两个版本受限于当时的硬件水平和 DOS 操作系统，并未取得较大反响。随后几年中，微软公司对 Windows 的内存管理、图形界面等做了大量重要的改进，在使得图形界面更为美观易用的同时还能支持虚拟内存，并最终于 1990 年推出 Windows 3.0，该系统一经面世就取得了巨大的商业成就，成功奠定了微软公司在操作系统尤其是个人桌面系统市场上的霸主地位。一年后，微软公司又推出了 Windows 3.1 系统，其在 Windows 3.0 基础上加入了可缩放的 TrueType 字体技术、对象链接和嵌入技术、多媒体技术等，使其具有更好的系统性能和可靠性，进一步稳固了市场。但直至 Windows 3.1，Windows 系列产品还是必须运行于 MS-DOS 操作系统之上，只是作为操作系统核心层次上的一个用户界面 shell 存在，整个系统的性能受到 MS-DOS 的严重桎梏。

为了从根本上改进系统体系结构、提高性能，微软公司又于 1995 年推出了全新的 Windows 95 操作系统。作为一个操作系统发展史上的里程碑产品，它可以独立运行，不必依赖于 MS-DOS，同时还提供了更加优良易用的、面向对象的 GUI，从而缩短了用户的学习时间，减轻了用户的学习负担；提供了对 32 位高性能抢占式多任务和多线程的支持，提高了进程并发效率；支持 Internet、高级多媒体服务、即插即用等新功能；在内存管理方面使用 32 位线性空间，提供了更灵活有效的分配和回收机制；为了保证以往的 16 位程序和硬件能继续使用，降低用户更新成本，还提供良好的向下兼容性等。

之后几年推出的 Windows 98 除了对 Windows 95 进行稳定的升级外，还支持多显示器显示、WebTV，同时采用了新的 FAT32 文件系统。Windows 98 还将 IE 集成进了 GUI，使得用户能更方便地浏览信息。2000 年 2 月，Windows 2000 推出，它可以通过互联网进行自动升级。但 Windows 2000 的最大优势在于采用了 NT 技术，该技术是微软公司推出的面向工作站、网络服务器和大型计算机的网络操作系统架构，可以与通信服务紧密集成，提供文件和打印服务，能运行客户机/服务器应用程序，分为工作站和服务器两种版本。

2001 年，Windows XP 发布，它将 NT 架构和从 Windows 9x 中继承来的用户界面融合在一起，进一步改善了 GUI，使得操作系统与一般用户的距离越来越近，同时在文件管理、速度和稳定性等方面比其前辈有了一定的改善。2014 年 4 月 8 日，微软公司宣布 Windows XP 正式退役。该系统作为微软公司最著名的产品，在十几年的时间内一直发挥着无可比拟的作用，其影响甚至在当前仍然存在。

为取代 Windows XP 并继续控制操作系统市场，微软公司于 2009 年 10 月推出了 Windows 7。该系统以 Windows NT 6.1 架构为基础，其设计对用户的更高应用需求更加关注，包括提高移动工作能力以保证对笔记本电脑的良好支持、基于应用服务的设计、对用户个性化的满足、优化视听娱乐感受、操作更便捷等。Windows 7 从公开测试之日起，就得到了广泛的赞誉，其相对于前代产品 Vista 具有更快速的启动、更美观的界面、更强大的多媒体支持、更好的无线连接、更人性化的用户账户控制等特点。与此同时，微软还发布了其服务器版本——Windows Server 2008 R2。

在 21 世纪的最初十年间，微软在操作系统领域的霸主地位受到了来自新老对手的威胁，例如苹果公司的 Mac OS X、各种版本的 UNIX 和 Linux 系统等。为了继续自身的辉煌，微软公司于 2012 年 10 月 26 日正式推出了 Windows 8，并宣传其具有革命性变化。其特点是采用了独特的 Metro 开始界面和触控式交互系统，旨在让用户的日常计算机操作更加简单和快捷，为用户提供高效易用的工作环境。该系统除可用于个人计算机外，还可以用于平板设备，在续航能力、移动速度、内存占用等方面表现良好，且兼容 Windows 7 所支持的软硬件。

2015 年 7 月 29 日，微软公司正式发布 Windows 10 操作系统，该系统可用于包括平板电脑在内的多种硬件平台。Windows 10 向使用 Windows 7、Windows 8.1 以及 Windows Phone 8.1 的用户提供首年免费升级服务。该系统在易用性和安全性方面有了极大的提升，除了针对云服务、智能移动设备、自然人机交互等新技术进行融合外，还对固态硬盘、生物识别、高分辨率屏幕等硬件进行了优化完善与支持。因此，Windows 10 迅速成为世界上使用最广泛的 PC 操作系统，有超过 13 亿设备使用该系统。

2021 年 6 月 24 日，微软公司推出新的 Windows 11 系统，并于当年 10 月 5 日正式发行。Windows 11 同样支持 Windows 10 设备升级。与前代产品相比，Windows 11 采用了全新的引导界面、开始菜单、设置面板、文件资源管理器等，将 Skype 升级为更便于办公协作和好友交互的 Windows Teams，可直接运行安卓应用，并对触摸体验、显示样式、Wi-Fi 支持、语音识别等方面进行了全面提升，更为美观易用。

1.4.2 UNIX 和 Linux 系统

1. UNIX

UNIX 系统可以说是当今除了 Windows 外应用最为广泛的操作系统。它出现于 20 世纪 60 年代晚期，最初用于小型机，后来逐渐在微型机、大型机、中型机和多处理机系统中应用，在商业和研究领域都获得了极大的成功。尽管 Windows NT 架构出现后，UNIX 系统受到了巨大的挑战，但其技术成熟度以及在稳定性、可靠性等方面均遥遥领先于 Windows NT，且由于开源软件社区的大力支持，使其目前仍是唯一被专业人士认可的、能在从巨型机到微型机甚至是嵌入式硬件平台上稳定运行的多用户、多任务操作系统。为了满足网络应用的需要，近年来的 UNIX 系统均配备了对网络提供支持的软件包，使得 UNIX 可以在企业网络中扮演网络 OS 的角色。

1971 年，Ennis Ritchie 和 Ken Thompson 首次使用汇编语言编写出 UNIX 的最初版本。1973 年，Ritchie 使用 C 语言对 UNIX 进行重写，形成了最早的 UNIX 正式版本 V5。随后几年发布的 V6 及其源代码被免费提供给大学和研究机构使用，在鼓励其对 UNIX 加以改进的同时，也使得一代程序人员对这个系统接受并熟悉起来，随着他们毕业并走向工作岗位，20 世纪 70 年代成为 UNIX 的时代。

UNIX 系统不断发展，后来顺序出现了 UNIX V7、UNIX System III、UNIX System V、UNIX SVR 2、UNIX SVR 3、UNIX SVR 4、UNIX SVR 4.2 等版本，而最初进入大学和研究机构的 V6 和 V7 版本则被各单位加以改进，形成了多个不同的 UNIX 版本。其中最著名的就是由加利福尼亚大学伯克利分校开发的 BSD UNIX 系列。

由于 UNIX 具有与生俱来的开放性、先进性和广泛的用户基础，在网络时代到来时，也有很多开发人员希望将其用作网络操作系统。很快，一系列 UNIX 网络应用程序就被装配到系统中来，在 V7 版本中已有一个最简单的网络支持软件。之后在各公司和研究院所、机构发布的 UNIX 版本中不断有新的网络支持程序加入，甚至形成了多个国际标准，其中最著名、最重要的就是 TCP/IP 协议。目前主流 UNIX 厂商都在不断地加强网络方面的研发力度，形成了各种用于企业网络的 UNIX 网络操作系统，如 SCO 公司的 Unixware NOS、Sun 公司的 Solaris NOS 等。

UNIX 的内部机制十分复杂，这里仅对其内核结构进行简单探讨。如图 1-5 所示，整个 UNIX 系统可以分为 4 个层次。第一层硬件层位于最底层处，是整个系统的物理基础。第二层内核层紧邻硬件层，用来管理系统的处理器、存储器、设备和文件资源。该层需要出现两个接口，一个是向下与硬件层通信的接口，由一组驱动程序和基本例程组成；另一个是内核与 Shell 的接口，由系统调用和命令解释程序等组成。而在这两个接口之间出现的就是 UNIX 的核心功能，它包括进程控制子系统和文件子系统。第三层是操作系统与用户的接口——Shell，在该层中通常还会出现编译程序。最高层为应用程序层。

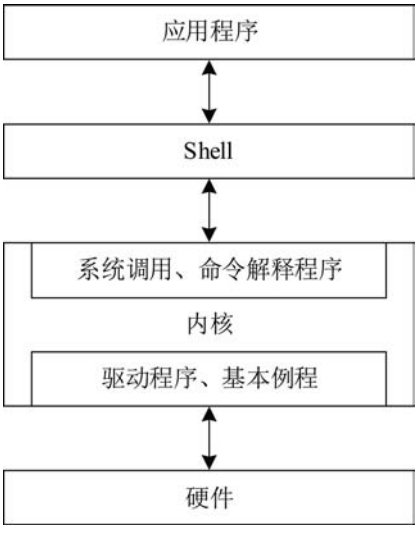

图 1-5　UNIX 内核结构

2. Linux

Linux 起源于 UNIX 社区，由芬兰学生 Linus Torvalds 首次创建。作为一个学生，Linus Torvalds 在学习一个 UNIX 的变种 MINIX 时萌发制作一个自己的、更健壮和有用的操作系统的想法。经过大量的学习和一段时间的努力后，Linus Torvalds 终于在 1991 年将自己的第一个作品发布到论坛上，并保证每个想要使用或研究该作品的人都可以免费得到源代码。很快来自全世界的大量顶尖程序人员加入修改和完善该源代码的队列中，最终产生了 Linux 的第一个正式发行版本 V1.0 版。作为第一个真正全开源的 UNIX 版本，Linux 受到全世界的关注和喜爱。在其出现 5 年后的 1996 年，Linux 成为一个重要的操作系统，并在个人桌面、服务器等多个领域表现出色。直到今天，它依然是一个首要的开源类 UNIX 系统。在教育领域，对 Linux 的学习也有助于学生理解内核工作方式。

Linux 作为一个类 UNIX 系统，本质上只是提供了一个操作系统内核，即只实现了操作系统的基本功能，如对内存、设备、文件系统、处理机的管理和分配等重要功能。而其他的扩展功能，尤其是直接面向用户的 Shell 层次的各种功能，则由分布在全球的各类公司、研究机构、大专院校等单位根据自身或客户需要自主开发。由于这些 Shell 工具经常是开源的，用户完全可以根据自身需要在选定的内核上安装最适合自己的外部功能程序，构成一个个性化的完整操作系统。对于非专业且对 Windows 极为失望的用户而言，可以选择那些由软件公司开发和打包的、具有美观桌面和类似于 Windows 的操作方式的发行版本。

1.5 小结

操作系统在计算机科学的发展中起着至关重要的作用，任何一个版本的操作系统都汇集了当时最先进的计算机科学的新成果和新技术。从最早的人工操作系统到如今的嵌入式操作系统、网络操作系统、分布式操作系统的整个发展过程中，操作系统的功能不断扩充，帮助用户更方便快捷地利用计算机性能。

操作系统的发展虽然快速，但其基本功能仍可以被描述为对软硬件资源的控制和管理，以及向用户提供直观、快捷的接口界面。这里的软硬件资源指的是文件、处理器、存储器和外部设备。

文件管理功能的管理对象主要是辅存上的各种程序和数据的集合。这部分管理功能包括如何有效地管理文件，以便存取文件时准确定位文件所在的位置。此外还要向用户提供操作文件的统一接口，使用该操作系统的用户只能利用这种标准接口申请系统为其服务。最后，在多用户共享文件时，除了保证有效共享外还要考虑文件的信息安全问题。

处理器管理在多道程序环境中作用举足轻重，它要控制和管理各进程的生命周期和执行次序，以协调多道程序间的关系，达到充分利用处理机资源的目的。其主要功能包括进程控制、进程同步、进程调度和进程通信等。

存储器管理对象主要指主存，由于任何程序想要运行必须以进程形式先装入内存，因此内存是计算机体系中仅次于 CPU 的重要资源。如何提高有限内存的利用率和存取效率是存储器管理部分要考虑的主要问题。存储器管理的主要功能包括内存分配、内存回收、内存保护、地址映射和虚拟内存等。

设备管理部分涉及类型多样的、复杂的外部设备，这些设备要与主机并行工作，因此其与 CPU 的速度矛盾和进程竞争使用问题最不好解决。设备管理主要包括设备的分配和回收、设备输入输出控制、设备缓冲管理等。另外，为了方便用户，这部分还要向用户提供统一的操作接口。为提高设备利用率，还可以提供虚拟设备功能，使一台物理机器能被多人共享，加快程序的执行过程。

在操作系统的结构划分上，可以将其分为整体结构、分层结构、虚拟机结构、外核结构和客户机/服务器结构。在不同的操作系统应用领域，可以选用不同结构，以提供更为高效易用的操作系统。

1.6 思考练习

1. 试说明什么是操作系统，其具有什么特征？其最基本特征是什么？
2. 设计现代操作系统的主要目标是什么？
3. 操作系统的作用体现在哪些方面？
4. 试说明实时操作系统和分时操作系统在交互性、及时性和可靠性方面的异同。
5. 试比较分布式操作系统和网络操作系统的异同。
6. 什么是操作系统虚拟机结构？它有什么好处？

7. 试说明客户机/服务器结构的操作系统为什么获得广泛应用。
8. 请简要描述处理器管理的主要功能。
9. 请简要描述存储器管理的主要功能。
10. 请简要描述文件管理的主要功能。
11. 请简要描述设备管理的主要功能。