

有心在 志所在
众志成城
大爱皆无疆



坚决打赢疫情防控阻击战！！

疫情面前，让我们一起努力！



第一章 操作系统引论



同学们，开始上课了！



即将开启

《操作系统原理》课程 学习旅程了！

明确：我们要如何开始课程学习

明确：课程的考核及评价标准

明确：教材及参考资料

明确：要遵守的课程规则

明确：咱们的接头方式



明确 1：我们要如何开始课程学习

二、课程性质及目标

《计算机操作系统》课程属于计算机科学与技术、软件工程、网络工程专业的专业教育课程。操作系统课程是深入了解和使用计算机的必备知识，是进行系统软件开发的理论基础，也是一门理论性和实践性并重的核心主干课程。本课程主要探讨操作系统的目标、作用、基本特性，以及操作系统的组成和实现方式。✎

通过本课程的学习，使学生全面地了解和掌握操作系统的目标、作用和模型，从资源管理的角度领会操作系统的功能和实现过程。使学生对计算机系统的软硬件资源的管理具有深入的理解，能够运用操作系统的工作原理，提出实际工程问题的解决方案，解决具体实际问题。✎

节选自 2018 版培养方案的教学大纲



明确 1：我们要如何开始课程学习

三、课程目标对毕业要求的支撑

1. 课程目标

目标 1：通过概念理解原理，通过原理理解计算机管理计算机软硬件资源的方式。

目标 2：掌握操作系统对处理器、内存、设备等资源的管理的具体实现方法，使学生能够综合分析各种资源管理之间的关系，具有求解计算机软硬件系统出现的复杂问题的能力。

目标 3：能够根据所学的操作系统知识，提出实际工程问题的解决方案，解决计算机软硬件系统中出现的复杂工程问题。

毕业要求

毕业要求 1：工程知识

指标点 1.3：具备软件开发等基本知识，并能用于解决软件工程领域开发相应软件所需工程技术问题。

毕业要求 2：问题分析

指标点 2.2：分析软件应用领域的技术问题，能够应用专业知识分析、设计解决方案。

毕业要求 3：设计/开发解决方案

指标点 3.2：能够针对工程中软件需求设计软件总体结构、具有把复杂问题进行功能模块分解能力。



明确 2：课程的考核及评价标准

3、通过多个环节的训练和考核，促进学习目标的达成：↵

（1）平时成绩（包括上课表现、作业、测试等）↵

（2）实验成绩↵

（3）期末考试↵

4、成绩评定↵

平时成绩↵	实验成绩↵	期末考试↵
20%↵	10%↵	70%↵

节选自 2018 版培养方案的教学大纲



明确 3：教材及参考资料

教材

汤小丹、梁红兵、哲凤屏、汤子瀛编著 . 计算机操作系统 . 西安电子科技大学出版社 .

参考书目

- [1] 汤子瀛等 . 计算机操作系统（第二版） . 西安电子科技大学出版社 .2001.
- [2] 刘福岩等 . 计算机操作系统（第一版） . 兵器工业出版社 .2004.
- [3] 张尧学 , 宋虹 , 张高 . 计算机操作系统教程（第四版） . 清华大学出版社 .2013.
- [4] William Stallings 著 . 陈向群 , 陈渝等译 . 操作系统 - 精髓与设计原理 . 电子工业出版社 .2017.
- [5] 周自恒 , 理黎明 , 曾祥江 , 张文旭译 . 30 天自制操作系统 . 人民邮电出版社 .2012.
- [6] 梁红兵 , 汤小丹 . 计算机操作系统（第四版）学习指导与题解 . 西安电子科技大学出版社 .2013.



明确 4：要遵守的课程规则

不许迟到

不许抄袭作业

不许逃课

不许溜号

不许交头接耳



明确 5：咱们的接头方式

最最重要的事情！

我的电话： 13504314198

QQ： 173545532

我们的群号： 852891219



第一章 操作系统引论

1.1 操作系统的目标 and 作用

1.2 操作系统的发展过程

1.3 操作系统的基本特性

1.4 操作系统的主要功能

1.5 操作系统结构设计



第一章 操作系统引论



第一章 操作系统引论



第一章 操作系统引论

AGP 插槽

PCI 插槽

主板

北桥芯片

内存

硬盘

电源

CPU 及风扇

光驱



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：

1. 方便性 { 便于理解计算机
 : 通过接口方便地使用

2. 有效性 : { 提高系统资源利用率
 提高系统的吞吐量

3. 可扩充性 : { OS 应方便修改、增加，以适应计算机硬件、体系结构以
 及应用发展的要求。

性 : { OS 应提供统一的开放环境，以实现应用的可移植性和互操作性（不同的设备能通过网络加以集成，并能正确、有效地协同工作）

4. 开放性 :



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

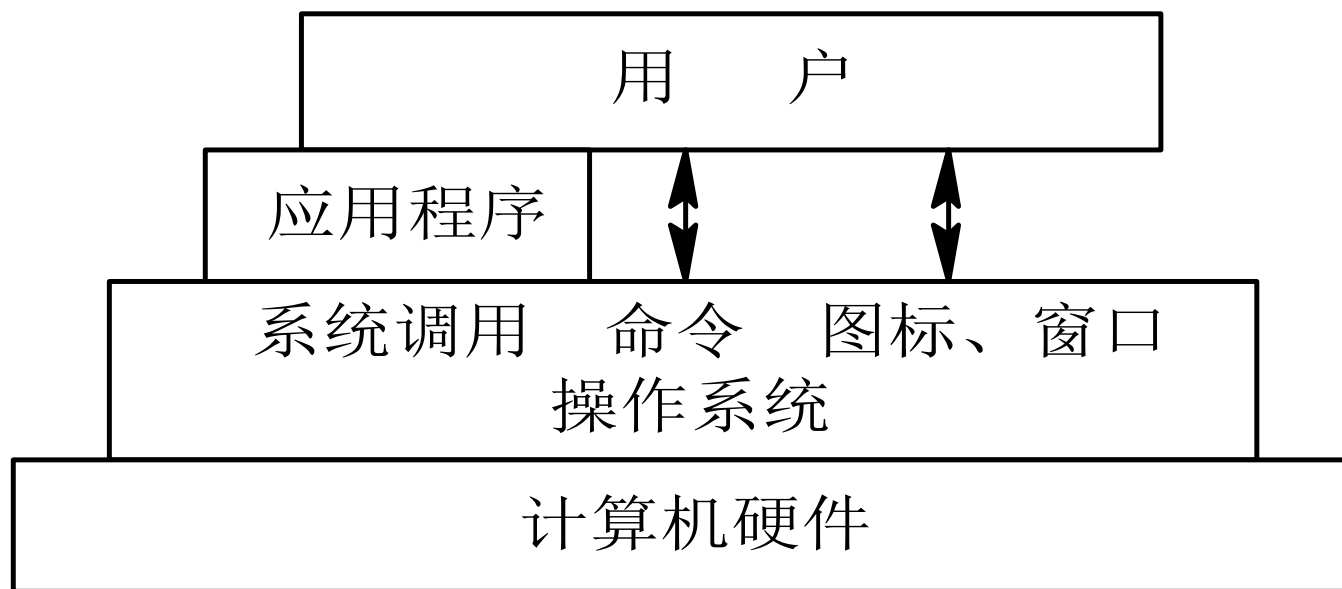


图 1-1 OS 作为接口的示意图



(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。

(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。

(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。



2. OS 作为计算机系统资源的管理者

在一个计算机系统中，硬件和软件资源归纳起来可分为四类：处理器、存储器、I/O 设备以及文件（数据和程序）。

相应地，OS 的主要功能也正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。

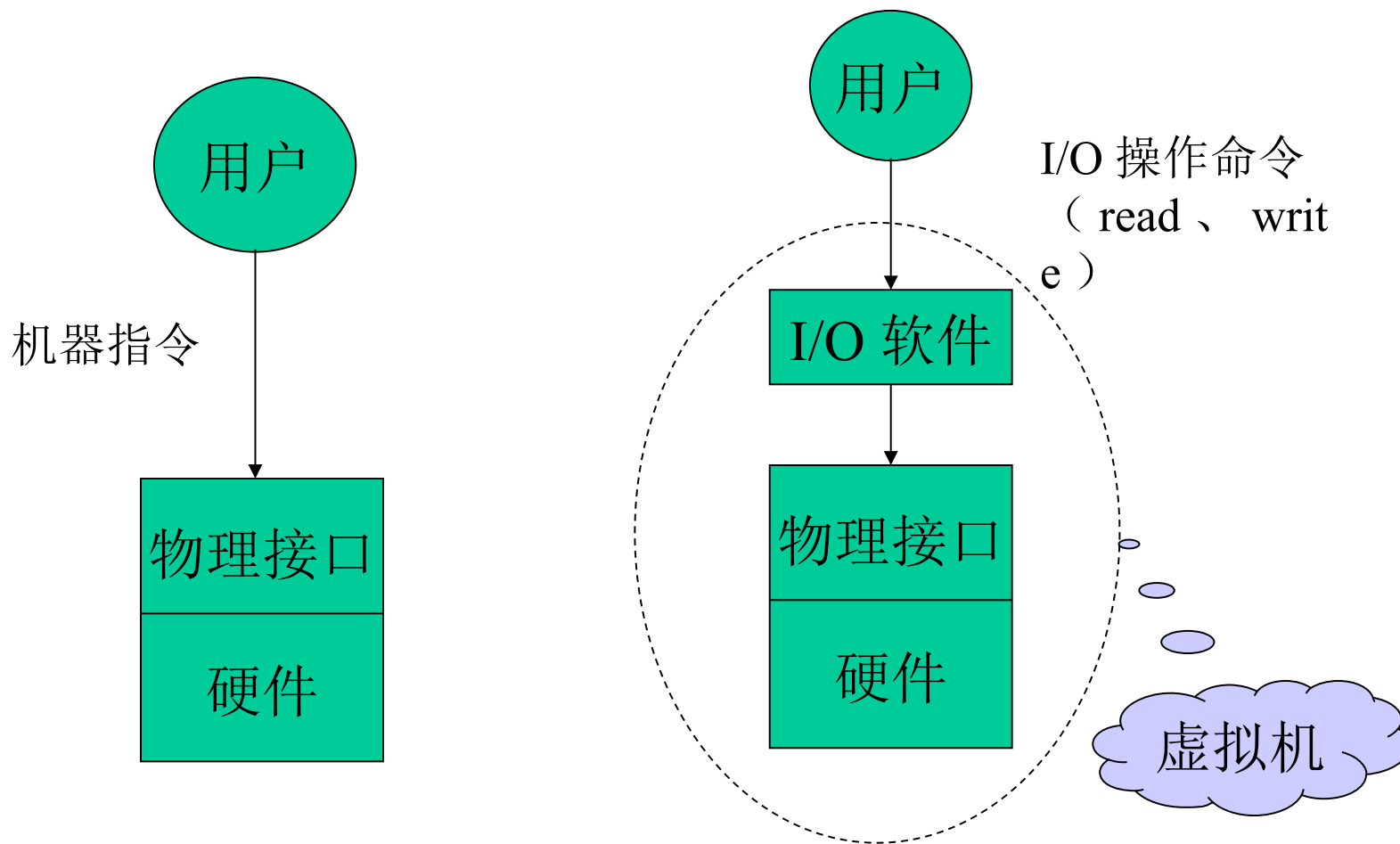
事实上，当今世界上广为流行的一个关于 OS 作用的观点，正是把 OS 作为计算机系统的资源管理者。



3. OS 实现了对计算机资源的抽象

对于一台完全无软件的计算机系统（即裸机），即使其功能再强，也必定是难于使用的。如果我们在裸机上覆盖上一层 I/O 设备管理软件，用户便可利用它所提供的 I/O 命令，来进行数据输入和打印输出。此时用户所看到的机器，将是一台比裸机功能更强、使用更方便的机器。**通常把覆盖了软件的机器称为扩充机器或虚拟机。**

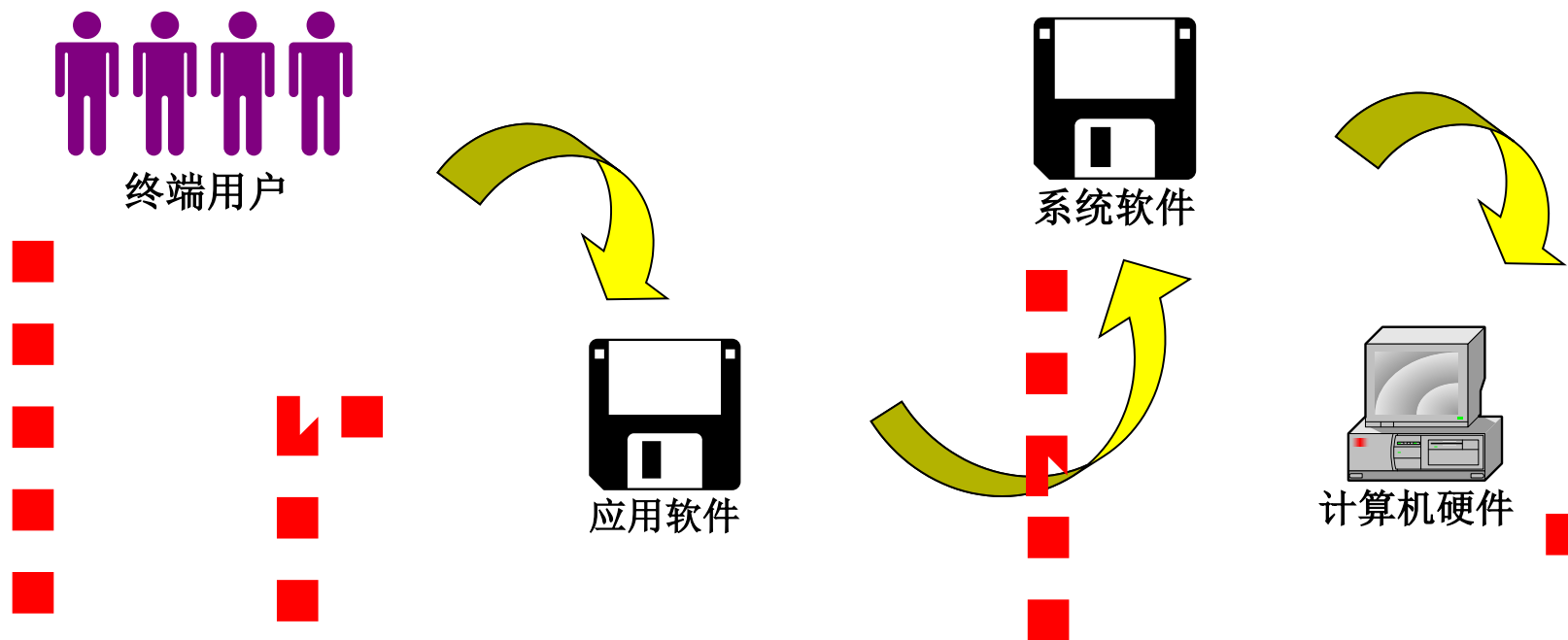
如果我们又在第一层软件上再覆盖上一层文件管理软件，则用户可利用该软件提供的文件存取命令，来进行文件的存取。此时，用户所看到的是台功能更强的虚拟机。如果我们又在文件管理软件上再覆盖一层面向用户的窗口软件，则用户便可在窗口环境下方便地使用计算机，形成一台功能更强的虚拟机。



- 图 1-2 I/O 软件隐藏了 I/O 操作实现的细节



软件的基本概念



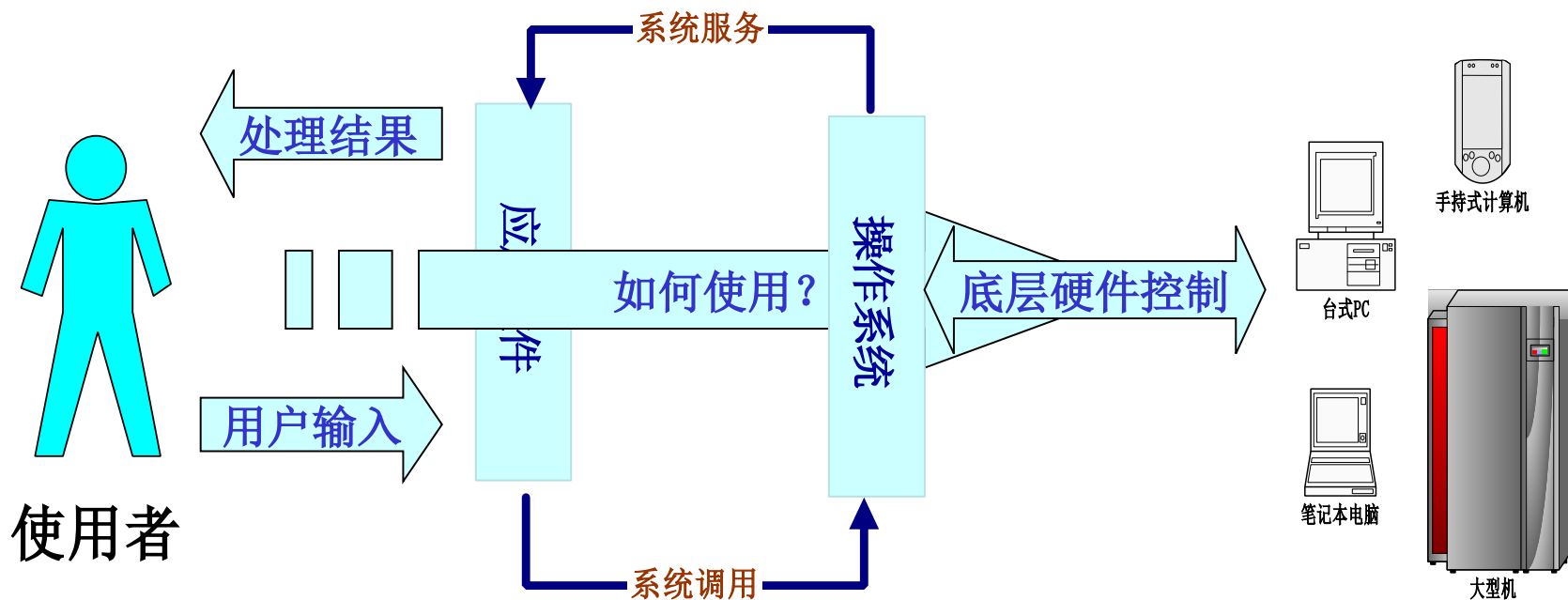
应用软件

系统软件

1. 创建、处理、分析信息并实现应用软件与计算机硬件的相互通信
2. 为计算机用户提供网络硬件设备
2. 应用软件的职责是对输入、处理数据、提供输出



操作系统的作用



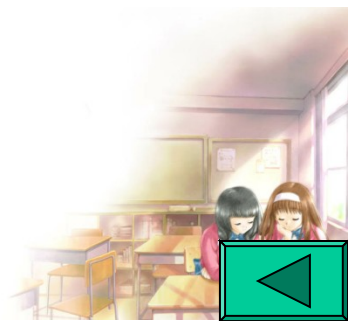
操作系统是控制和管理计算机系统的硬件和软件资源，合理的组织计算机工作流程以及方便用户使用的程序和数据的集合。

计算机硬件设备



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（如接口）
3. 器件的不断更新换代（如 CPU 芯片发展）
4. 计算机体系结构的不断发展（如多处理机、网络）
5. 不断提出新的应用需求（如多媒体、嵌入）



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

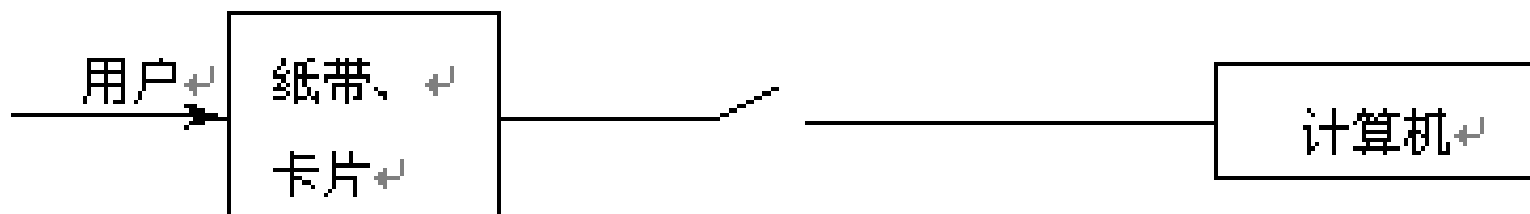
这种人工操作方式有以下两方面的缺点：

(1) 用户独占全机。 (2) CPU 等待人工操作。

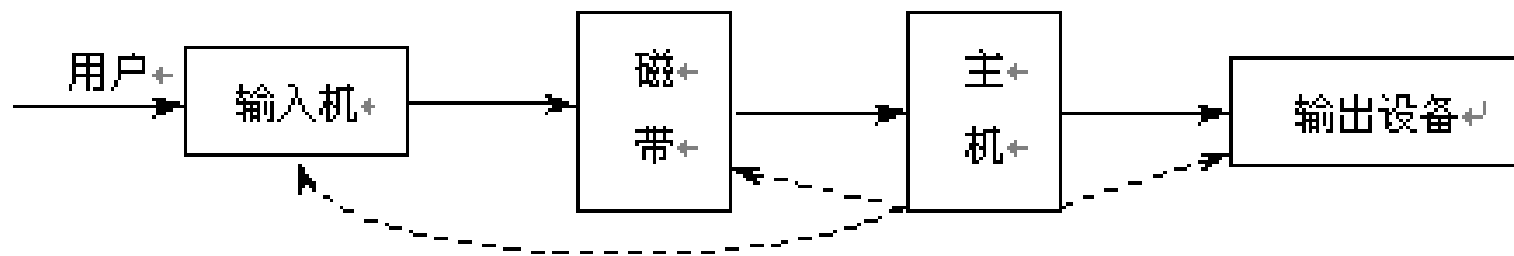
这时的计算机操作是由用户（即程序员）采用人工操作方式直接使用计算机硬件系统，即由程序员将事先已穿孔（对应于程序和数据）的纸带（或卡片）装入纸带输入机（或卡片输入机），再启动它们将程序和数据输入计算机，然后启动计算机运行。当程序运行完毕并取走计算结果后，才让下一个用户上机。



图例



手工时期的操作



联机批处理



1.2.1 无操作系统的计算机系统

1. 人工操作方式

2. 脱机输入 / 输出 (Off-Line I/O) 方式

这种脱机 I/O 方式的主要优点如下：

(1) 减少了 CPU 的空闲时间。

(2) 提高 I/O 速度。

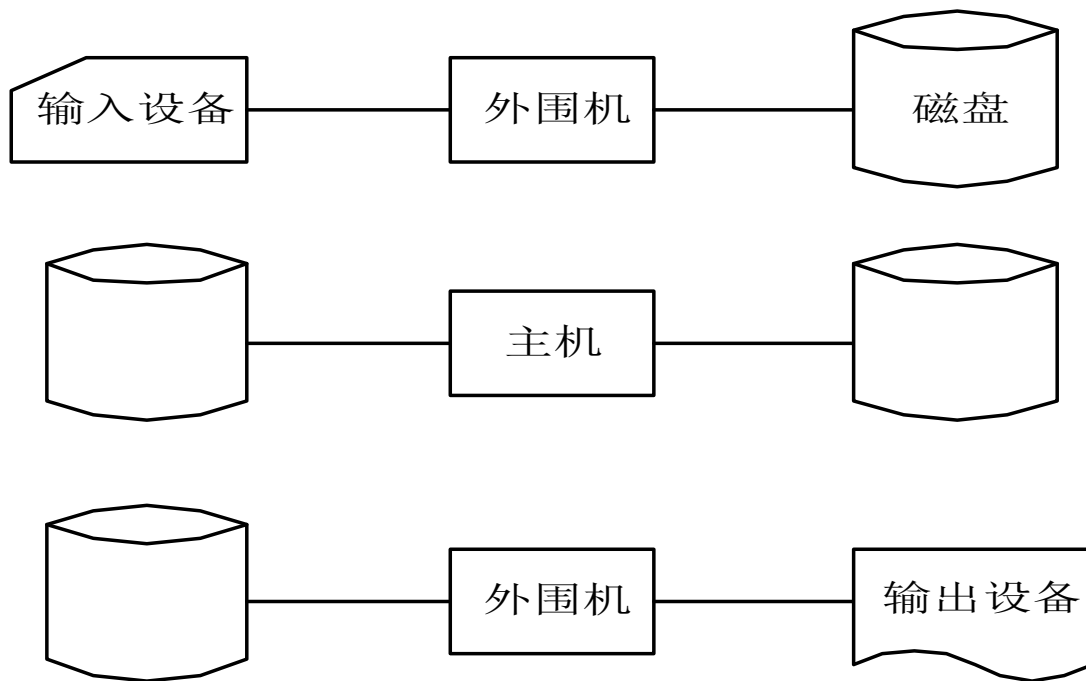


图 1-2 脱机 I/O 示意图



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

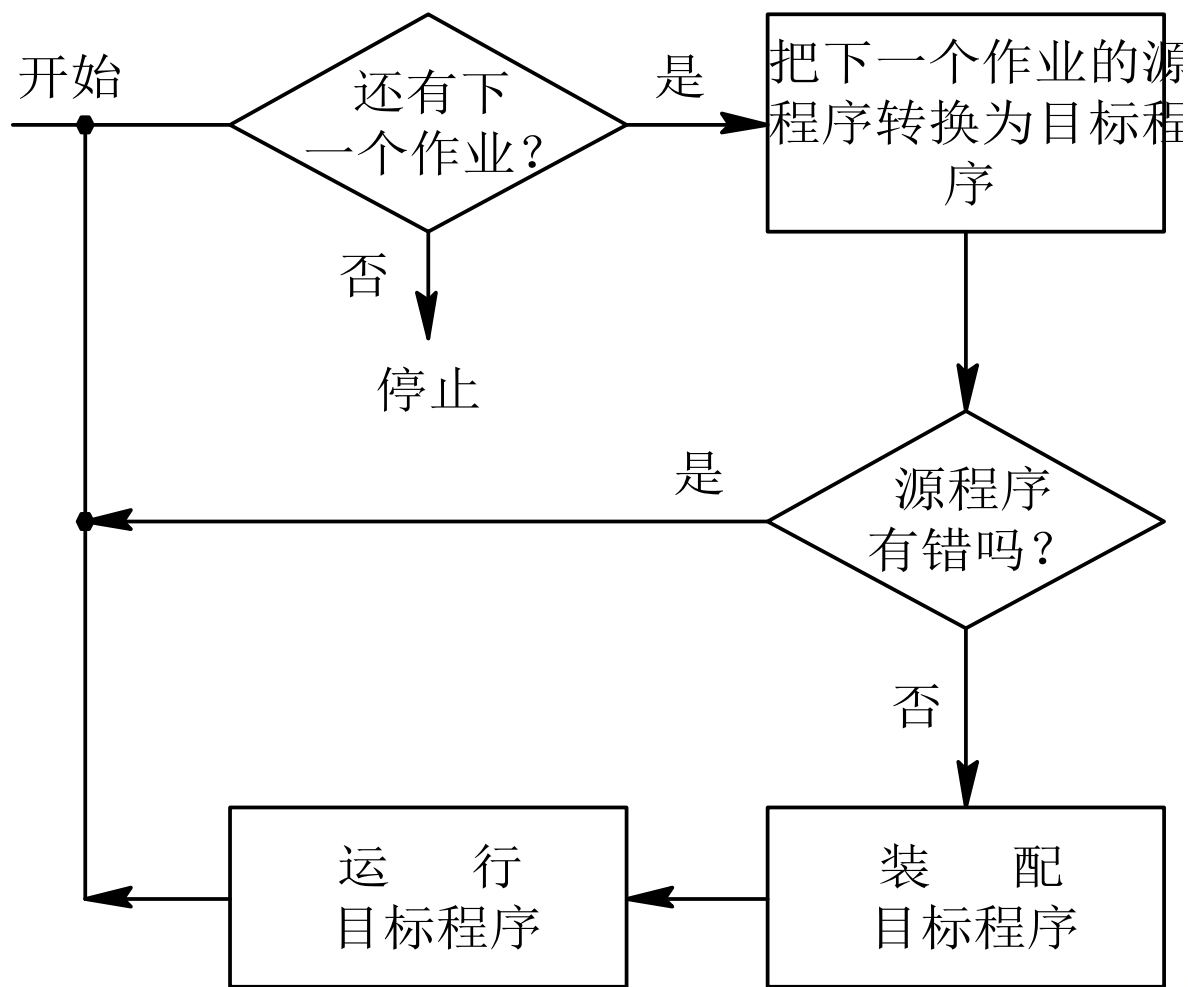


图 1-3 单道批处理系统的处理流程



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

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 的前身而并非是目前人们所理解的 OS。

尽管如此，该系统比起人工操作方式的系统已有很大进

该系统的主要特征如下：

- (1) 自动性。
- (2) 顺序性。
- (3) 单道性。



1.2.3 多道批处理系统

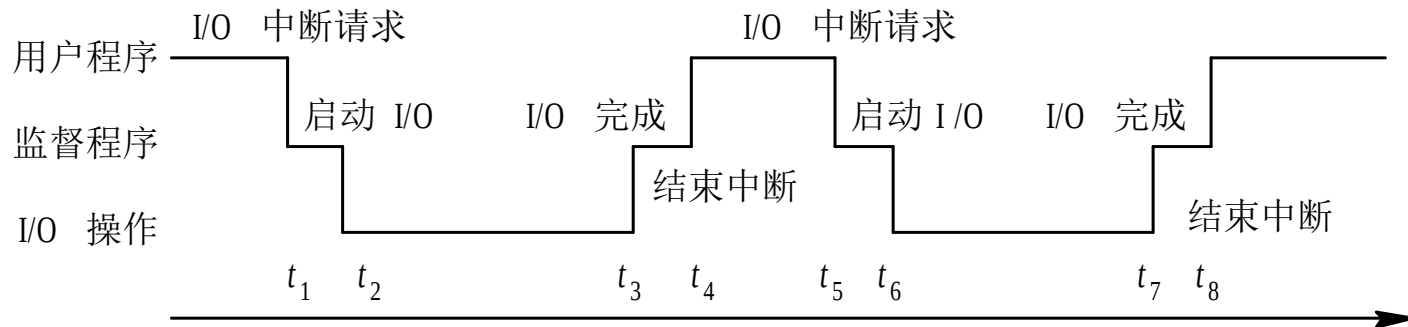
1. 多道程序设计的基本概念

在单道批处理系统中，内存中仅有一道作业，它无法充分利用系统中的所有资源，致使系统性能较差。为了进一步提高资源的利用率和系统吞吐量，在 60 年代中期又引入了多道程序设计技术，由此而形成了多道批处理系统 (Multiprogrammed Batch Processing System)。

在该系统中，用户所提交的作业都先存放在外存上并排成一个队列，称为“后备队列”；然后，由作业调度程序按一定的算法从后备队列中选择若干个作业调入内存，使它们共享 CPU 和系统中的各种资源。



第一章 操作系统引论



(a) 单道程序运行情况

图 1-4 单道和多道程序运行情况



在 OS 中引入多道程序设计技术可带来以下好处：

(1) 提高 CPU 的利用率。

在引入多道程序设计技术后，由于同时在内存中装有若干道程序，并使它们交替地运行，这样，当正在运行的程序因 I/O 而暂停执行时，系统可调度另一道程序运行，从而保持了 CPU 处于忙碌状态。



在 OS 中引入多道程序设计技术可带来以下好处：

(1) 提高 CPU 的利用率。

(2) 可提高内存和 I/O 设备利用率。

为了能运行较大的作业，通常内存都具有较大容量，但由于 80% 以上的作业都属于中小型，因此在单道程序环境下，也必定造成内存的浪费。类似地，对于系统中所配置的多种类型的 I/O 设备，在单道程序环境下也不能充分利用。如果允许在内存中装入多道程序，并允许它们并发执行，则无疑会大大提高内存和 I/O 设备的利用率。



在 OS 中引入多道程序设计技术可带来以下好处：

- (1) 提高 CPU 的利用率。
- (2) 可提高内存和 I/O 设备利用率。
- (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。



1.2.3 多道批处理系统

1. 多道程序设计的基本概念

2. 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。



1.2.3 多道批处理系统

1. 多道程序设计的基本概念
2. 多道批处理系统的优缺点

3. 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。



1.2.4 分时系统

1. 分时系统 (Time-Sharing System) 的产生

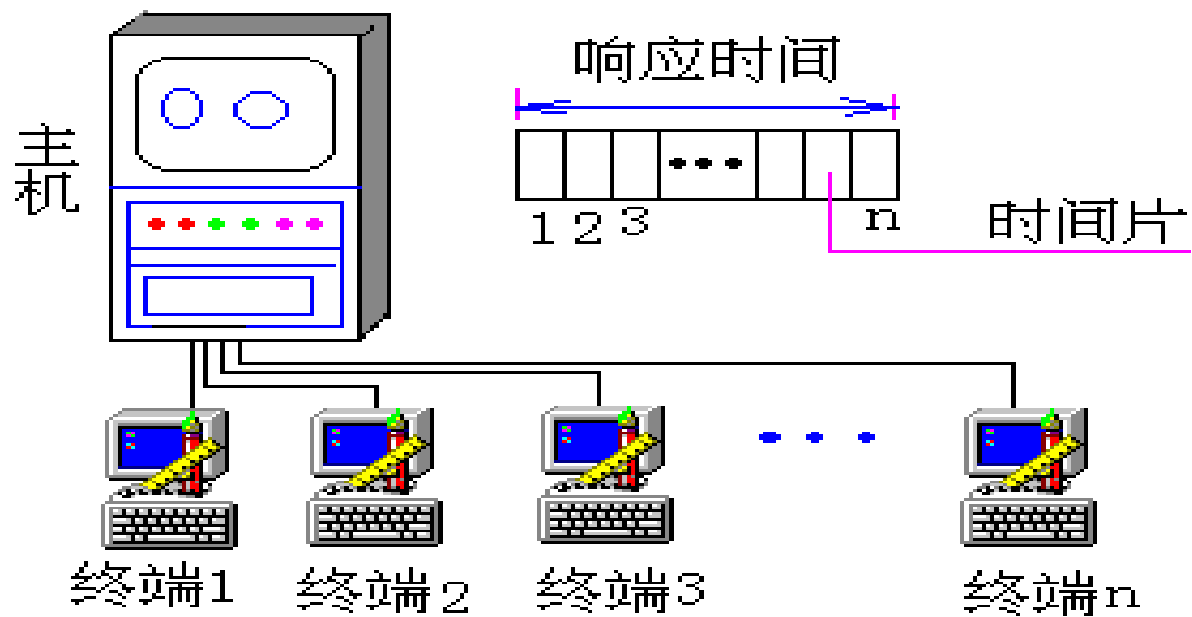
如果说，推动多道批处理系统形成和发展的主要动力，是提高资源利用率和系统吞吐量，那么，推动分时系统形成和发展的主要动力，则是用户的需求。或者说，分时系统是为了满足用户需求所形成的一种新型 OS。它与多道批处理系统之间，有着截然不同的性能差别。

用户的需求具体表现在以下几个方面：

- (1) 人一机交互。
- (2) 共享主机。
- (3) 便于用户上机。



分时系统的结构示意图



- (1) 人—机交互。
- (2) 共享主机。
- (3) 便于用户上机。



1.2.4 分时系统

1. 分时系统 (Time-Sharing System) 的产生

2. 分时系统实现中的关键问题

为实现分时系统，其中，最关键的问题是如何使用户能与自己的作业进行交互，即当用户在自己的终端上键入命令时，系统应能及时接收并及时处理该命令，再将结果返回给用户。此后，用户可继续键入下一条命令，此即人一机交互。应强调指出，即使有多个用户同时通过自己的键盘键入命令，系统也应能全部地及时接收并处理。

(1) 及时接收。

(2) 及时处理。



1.2.4 分时系统

1. 分时系统 (Time-Sharing System) 的产生
2. 分时系统实现中的关键问题

3. 分时系统的特征

- (1) 多路性。
- (2) 独立性。
- (3) 及时性。
- (4) 交互性。



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。



1.2.5 实时系统

1. 应用需求

2. 实时任务

1) 按任务执行时是否呈现周期性来划分

(1) 周期性实时任务。

(2) 非周期性实时任务。

外部设备所发出的激励信号并无明显的周期性，但都必须联系着一个截止时间 (Deadline)。它又可分为：

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。



1.2.5 实时系统

1. 应用需求

2. 实时任务

1) 按任务执行时是否呈现周期性来划分

(1) 周期性实时任务。

(2) 非周期性实时任务。

2) 根据对截止时间的要求来划分

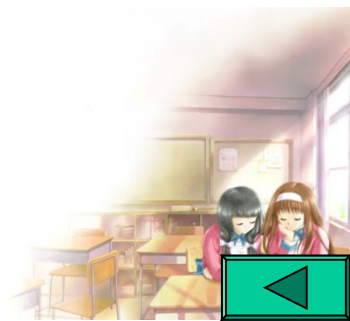
(1) **硬实时任务 (hard real-time task)**。系统必须满足任务对截止时间的要求，否则可能出现难以预测的结果。

(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。



1.2.6 微机操作系统的发展

- 微机操作系统按照字长分为
- 8 位、16 位、32 位、64 位；
- 按照运行方式分为：
 - 单用户单任务操作系统
 - 单用户多任务操作系统
 - 多用户多任务操作系统



1. 单用户单任务操作系统

- CP/M, 1975 年, Digital Reswarch 公司为 Intel8080 开发的。
- MS-DOS, 81 年微软为 IBM 开发的

2. 单用户多任务操作系统

1985 年推出 windows1.0, 2001 年 windows XP

3. 多用户多任务操作系统

1969-1970 年贝尔实验室 Ennis Ritchie (73 年 C 重写) 和 Ken Thompson (B 重写)

Solaris OS: SUN 公司 82 年推出运行在 Motorola680x0 平台上的 Unix OS;

Linux OS: 91 年芬兰 Linus Torvalds 为 386 开发的。



1.3 操作系统的基本特性

1.3.1 并发 (Concurrence)

1 . 并行与并发

并行性和并发性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。



1.3 操作系统的基本特性

1.3.1 并发 (Concurrence)

1. 并行与并发

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. 引入进程

程序为静态的，进程为程序的执行

3. 引入线程

进程的更小的可执行单位，一个进程可以包含若干个线程



1.3.2 共享 (Sharing)

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。由于资源属性的不同，进程对资源共享的方式也不同，目前主要有以下两种资源共享方式。



1.3.2 共享 (Sharing)

1. 互斥共享方式

系统中的某些资源，如打印机、磁带机，虽然它们可以提供给多个进程（线程）使用，但为使所打印或记录的结果不致造成混淆，应规定在一段时间内只允许一个进程（线程）访问该资源。

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。



1.3.2 共享 (Sharing)

1. 互斥共享方式

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。这里所谓的“同时”往往是宏观上的，而在微观上，这些进程可能是交替地对该资源进行访问。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。



1.3.2 共享 (Sharing)

1. 互斥共享方式
2. 同时访问方式

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。



1.3.3 虚拟 (Virtual)

操作系统中的所谓“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。物理实体（前者）是实的，即实际存在的；而后者是虚的，是用户感觉上的东西。相应地，用于实现虚拟的技术，称为虚拟技术。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。



1.3.3 虚拟 (Virtual)

1. 时分复用技术（即分时使用方式）

(1) 虚拟处理机技术

在虚拟处理机技术中，**是通过多道程序设计技术，让多道程序并发执行的方法，来分时使用一台处理机的。**此时，虽然只有一台处理机，但它能同时为多个用户服务，使每个终端用户都认为是有一个 CPU 在专门为他服务。亦即，利用多道程序设计技术，把一台物理上的 CPU 虚拟为多台逻辑上的 CPU，也称为虚拟处理机，我们把用户所感觉到的 CPU 称为虚拟处理器。



1.3.3 虚拟 (Virtual)

1. 时分复用技术（即分时使用方式）

(1) 虚拟处理机技术

(2) 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。



1.3.3 虚拟 (Virtual)

1. 时分复用技术（即分时使用方式）

(1) 虚拟处理机技术

(2) 虚拟设备技术

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。此外，也可以把一条物理信道虚拟为多条逻辑信道（虚信道）。

在操作系统中，虚拟的实现主要是通过分时使用的方法。显然，如果 n 是某物理设备所对应的虚拟的逻辑设备数，则虚拟设备的平均速度必然是物理设备速度的 $1/n$ 。



1.3.3 虚拟 (Virtual)

1. 时分复用技术 (即分时使用方式)

2. 空分复用技术

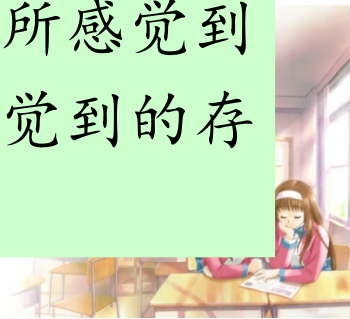
(1) 虚拟磁盘技术

将硬盘划分为若干个卷，机器上便会有若干个虚拟盘。

(2) 虚拟存储器技术

将一台机器的物理存储器变为虚拟存储器，以便从逻辑上来扩充存储器的容量。

此时，虽然物理内存的容量可能不大（如 32 MB），但它可以运行比它大得多的用户程序（如 128 MB）。这使用户所感觉到的内存容量比实际内存容量大得多，我们把用户所感觉到的存储器称为虚拟存储器。



1.3.4 异步性 (Asynchronism)

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，由于系统中只有一个处理机，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，由于打印机属于临界资源，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。**



1.3.4 异步性 (Asynchronism)

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，都是不可预知的。由于各用户程序性能的不同，比如，有的侧重于计算而较少需要I/O；而又有有的程序其计算少而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，此即进程的异步性。**尽管如此，但只要运行环境相同，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，是操作系统的一个重要特征。



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

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。



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

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。有两种协调方式：

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。。



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

2. 进程同步

为了实现进程同步，系统中必须设置进程同步机制。最简单的用于实现进程互斥的机制，是为每一个临界资源配置一把锁 W ，当锁打开时，进程（线程）可以对该临界资源进行访问；而当锁关上时，则禁止进程（线程）访问该临界资源。



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
2. 进程同步

3. 进程通信

在多道程序环境下，为了加速应用程序的运行，应在系统中建立多个进程，并且再为一个进程建立若干个线程，由这些进程（线程）相互合作去完成一个共同的任务。而在这些进程（线程）之间，又往往需要交换信息。



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
2. 进程同步
3. **进程通信**

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。输入进程负责将所输入的数据传送给计算进程；计算进程利用输入数据进行计算，并把计算结果传送给打印进程；最后，由打印进程把计算结果打印出来。**进程通信的任务就是用来实现在相互合作的进程之间的信息交换。**

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。



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
2. 进程同步
3. 进程通信

4. 调度

在后备队列上等待的每个作业，通常都要经过调度才能执行。在传统的操作系统中，包括作业调度和进程调度两步。作业调度的基本任务，是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）。



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
 2. 进程同步
 3. 进程通信
 4. 调度
- 在将它们调入内存后，便分别为它们建立进程，使它们都成为可能获得处理机的就绪进程，并按照一定的算法将它们插入就绪队列。**而进程调度的任务，则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，为此，须把就绪线程排成一个队列，每次调度时，是从就绪线程队列中选出一个线程，把处理机分配给它。



1.4.2 存储器管理功能

1. 内存分配

OS 在实现内存分配时，可采取静态和动态两种方式。

在静态分配方式中，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”；

在动态分配方式中，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”。



1.4.2 存储器管理功能

1. 内存分配

为了实现内存分配，在内存分配的机制中应具有这样的结构和功能：

- ① **内存分配数据结构**，该结构用于记录内存空间的使用情况，作为内存分配的依据；
- ② **内存分配功能**，系统按照一定的内存分配算法，为用户程序分配内存空间；
- ③ **内存回收功能**，系统对于用户不再需要的内存，通过用户的释放请求，去完成系统的回收功能。



2. 内存保护

内存保护的主要任务，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。系统须对每条指令所要访问的地址进行检查，如果发生越界，便发出越界中断请求，以停止该程序的执行。如果这种检查完全用软件实现，则每执行一条指令，便须增加若干条指令去进行越界检查，这将显著降低程序的运行速度。因此，越界检查都由硬件实现。当然，对发生越界后的处理，还须与软件配合来完成。



3. 地址映射

一个应用程序（源程序）经编译后，通常会形成若干个目标程序；这些目标程序再经过链接便形成了可装入程序。这些程序的地址都是从“0”开始的，程序中的其它地址都是相对于起始地址计算的；由这些地址所形成的地址范围称为“地址空间”，其中的地址称为“逻辑地址”或“相对地址”。此外，由内存中的一系列单元所限定的地址范围称为“内存空间”，其中的地址称为“物理地址”。

在多道程序环境下，**每道程序不可能都从“0”地址开始装入（内存），这就致使地址空间内的逻辑地址和内存空间中的物理地址不相一致。**使程序能正确运行，存储器管理必须提供地址映射功能，以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。该功能应在硬件的支持下完成



4. 内存扩充

存储器管理中的内存扩充任务，并非是去扩大物理内存的容量，而是借助于虚拟存储技术，从逻辑上去扩充内存容量，使用户所感觉到的内存容量比实际内存容量大得多；或者是让更多的用户程序能并发运行。这样，既满足了用户的需要，改善了系统的性能，又基本上不增加硬件投资。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：

- (1) 请求调入功能。
- (2) 置换功能。



1.4.3 设备管理功能

设备管理用于管理计算机系统中所有的外围设备，而设备管理的主要任务是，**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。**为实现上述任务，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。



1. 缓冲管理

CPU 运行的高速性和 I/O 低速性间的矛盾自计算机诞生时起便已存在。 而随着 CPU 速度迅速、大幅度的提高，使得此矛盾更为突出，严重降低了 CPU 的利用率。如果在 I/O 设备和 CPU 之间引入缓冲，则可有效地缓和 CPU 和 I/O 设备速度不匹配的矛盾，提高 CPU 的利用率，进而提高系统吞吐量。 因此，在现代计算机系统中， 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。

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

设备分配的基本任务，是根据用户进程的 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。

为了实现设备分配，系统中应设置设备控制表、控制器控制表等数据结构，用于记录设备及控制器的标识符和状态。据这些表格可以了解指定设备当前是否可用，是否忙碌，以供进行设备分配时参考。在进行设备分配时，应针对不同的设备类型而采用不同的设备分配方式。对于独占设备（临界资源）的分配，还应考虑到该设备被分配出去后，系统是否安全。设备使用完后，还应立即由系统回收。



3. 设备处理

设备处理程序又称为设备驱动程序。**其基本任务是用于实现 CPU 和设备控制器之间的通信**，即由 CPU 向设备控制器发出 I/O 命令，要求它完成指定的 I/O 操作；反之由 CPU 接收从控制器发来的中断请求，并给予迅速的响应和相应的处理。

处理过程是：设备处理程序首先检查 I/O 请求的合法性，了解设备状态是否是空闲的，了解有关的传递参数及设置设备的工作方式。然后，便向设备控制器发出 I/O 命令，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，并根据该中断请求的类型，调用相应的中断处理程序进行处理。对于设置了通道的计算机系统，设备处理程序还应能根据用户的 I/O 请求，自动地



1.4.4 文件管理功能

1. 文件存储空间的管理

由文件系统对诸多文件及文件的存储空间，实施统一的管理。其主要任务是**为每个文件分配必要的外存空间，提高外存的利用率，并能有助于提高文件系统的运行速度。**

为此，系统应设置相应的数据结构，用于记录文件存储空间的使用情况，以供分配存储空间时参考；系统还应具有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

为了使用户能方便地在外存上找到自己所需的文件，通常由系统为每个文件建立一个目录项。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。**即用户只须提供文件名，即可对该文件进行存取。其次，目录管理还应能实现文件共享，这样，只须在外存上保留一份该共享文件的副本。此外，还应能提供快速的目录查询手段，以提高对文件的检索速度。



3. 文件的读 / 写管理和保护

(1) 文件的读 / 写管理。该功能是根据用户的请求，从外存中读取数据；或将数据写入外存。在进行文件读（写）时，系统先根据用户给出的文件名，去检索文件目录，从中获得文件在外存中的位置。然后，利用文件读（写）指针，对文件进行读（写）。一旦读（写）完成，便修改读（写）指针，为下一次读（写）做好准备。由于读和写操作不会同时进行，故可合用一个读 / 写指针。

(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 操作系统与用户之间的接口

1. 用户接口

(1) 联机用户接口。

这是为联机用户提供的，它由一组键盘操作命令及命令解释程序所组成。当用户在终端或控制台上每键入一条命令后，系统便立即转入命令解释程序，对该命令加以解释并执行该命令。在完成指定功能后，控制又返回到终端或控制台上，等待用户键入下一条命令。这样，用户可通过先后键入不同命令的方式，来实现对作业的控制，直至作业完成。



(2) 脱机用户接口。

该接口是为批处理作业的用户提供的，故也称为批处理用户接口。该接口由一组作业控制语言 JCL 组成。批处理作业的用户不能直接与自己的作业交互作用，只能委托系统代替用户对作业进行控制和干预。这里的作业控制语言 JCL 便是提供给批处理作业用户的、为实现所需功能而委托系统代为控制的一种语言。用户用 JCL 把需要对作业进行的控制和干预，事先写在作业说明书上，然后将作业连同作业说明书一起提供给系统。当系统调度到该作业运行时，又调用命令解释程序，对作业说明书上的命令，逐条地解释执行。如果作业在执行过程中出现异常现象，系统也将根据作业说明书上的指示进行干预。



(3) 图形接口

用户虽然可以通过联机用户接口来取得 OS 的服务，但这时要求用户能熟记各种命令的名字和格式，并严格按照规定的格式输入命令，这既不方便又花时间，于是，图形用户接口便应运而生。图形用户接口采用了图形化的操作界面，用非常容易识别的各种图标 (icon) 来将系统的各项功能、各种应用程序和文件，直观、逼真地表示出来。用户可用鼠标或通过菜单和对话框，来完成对应用程序和文件的操作。此时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。



2. 程序接口

该接口是为用户程序在执行中访问系统资源而设置的，是用户程序取得操作系统服务的惟一途径。它是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。早期的系统调用都是用汇编语言提供的，只有在用汇编语言书写的程序中，才能直接使用系统调用；但在高级语言以及 C 语言中，往往提供了与各系统调用一一对应的库函数，这样，应用程序便可通过调用对应的库函数来使用系统调用。但在近几年所推出的操作系统中，如 UNIX、OS/2 版本中，其系统调用本身已经采用 C 语言编写，并以函数形式提供，故在用 C 语言编制的程序中，可直接使用系统调用。



1.4.6 现代操作系统的新功能

1. 系统安全

现代 OS 中采用多种有效措施来确保系统的安全，主要的技术有：

●认证技术

查验是否名副其实，防止入侵者进行假冒或篡改等。

●密码技术

对系统中所需存储和传输的数据进行加密。

●访问控制技术

设置用户存取权限。
通过文件属性设置访问控制。

●反病毒技术

预防、检测、清除。



1.4.6 现代操作系统的新功能

1. 系统安全

2. 网络的功能和服务

●网络通信

源机与目标机之间实现无差错的数据传输

●资源管理

对网络中的共享资源进行有效的管理

●应用互操作

即实现不同网络中信息的互通性和信息的互用性



1.4.6 现代操作系统的新功能

1. 系统安全
2. 网络的功能和服务
3. 支持多媒体

● 接纳控制功能

为了确保实时进程的截至时间，
对任务数目应该加以限制

● 实时调度

调度增加考虑实时、接纳度

● 多媒体文件的存储

硬盘数据的快速传输



1.5 操作系统的结构设计

所谓软件，是指当计算机运行时，能提供所要求的功能和性能的指令和程序的集合，该程序能够正确地处理信息的数据结构；作为规范软件，还应具有描述程序功能需求以及程序如何操作使用的文档。

软件工程是指运用系统的、规范的和可定量的方法，来开发、运行和维护软件；或者说，是采用工程的概念、原理、技术和方法，来开发与维护软件。

软件质量的评价指标有：功能性、有效性、可靠性、易使用性、可维护性、和易移植性。



操作系统是一个十分复杂的大型软件。为了控制该软件的复杂性，在开发 OS 时，先后引入了分解、模块化、抽象和隐蔽等方法。开发方法的不断发展，促进了 OS 结构的更新换代。这里，我们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。

