



中山大學  
SUN YAT-SEN UNIVERSITY

# 操作系统

## 笔记整理

姓名：刘斯宇

学号：17341110

## 目录

<b>1</b>	<b>计算机系统概述</b>	<b>4</b>
1.1	基本构成 . . . . .	4
1.2	微处理器的发展 . . . . .	4
1.3	中断 . . . . .	5
1.3.1	中断和指令周期 . . . . .	5
1.3.2	中断处理 . . . . .	6
1.3.3	多个中断 . . . . .	7
1.4	存储器的层次结构 . . . . .	7
1.5	多处理器和多核计算机组织结构 . . . . .	7
<b>2</b>	<b>操作系统概述</b>	<b>9</b>
2.1	操作系统的目标和功能 . . . . .	9
<b>3</b>		<b>10</b>
3.1	计算机系统的层次结构 . . . . .	11
3.2	操作系统的概念 . . . . .	11
3.2.1	用户观点 . . . . .	11
3.2.2	系统观点 . . . . .	12
3.2.3	进程观点 . . . . .	12
3.2.4	虚拟机观点 . . . . .	12
3.3	操作系统的特征 . . . . .	12
3.3.1	并发性 . . . . .	12
3.3.2	共享性 . . . . .	13

3.3.3	虚拟性	13
3.3.4	异步性	13
3.4	操作系统的主要功能	13
3.4.1	处理器管理	13
3.5	存储器管理	14
3.6	设备管理	14
3.6.1	文件管理	14
3.6.2	用户接口	14
3.7	操作系统的发展	15
3.7.1	无操作系统阶段	15
3.7.2	单道批处理系统	16
3.7.3	多道批处理系统	16
3.8	操作系统的分类	16
3.8.1	批处理操作系统	16
3.8.2	分时操作系统	17
3.8.3	实时操作系统	18
3.8.4	嵌入式操作系统	19
3.8.5	分布式操作系统	19
3.9	操作系统的运行环境	19
3.9.1	内核态与用户态	19
3.9.2	中断与异常	20
3.9.3	系统调用	20
3.10	操作系统的体系结构	21
3.10.1	模块组合结构	21
3.10.2	层次结构	21
3.10.3	微内核结构	22
<b>4</b>	<b>进程管理</b>	<b>22</b>
4.1	进程	23

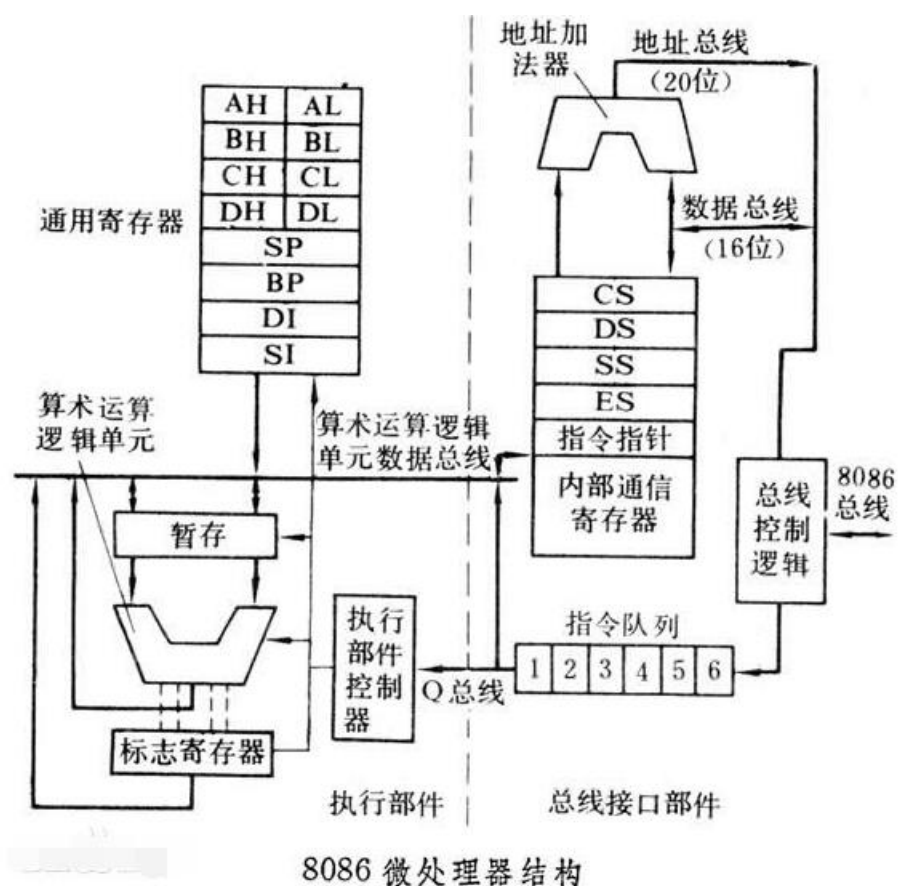
- ☒ 第一章--计算机系统概述
- ☒ 第二章--操作系统概述
- ☒ 第三章--进程描述和控制
- ☒ 第四章--线程
- ☒ 第五章--并发性：互斥和同步
- ☒ 第六章--并发：死锁和饥饿
- ☒ 第七章--内存管理
- ☒ 第八章--虚拟内存
- ☒ 第九章--单处理器调度
- ☒ 第十一章-I/O 管理和磁盘调度
- ☒ 第十二章--文件管瘤

# 1 计算机系统概述

表 1: 第一章关键术语汇总

名称	含义
处理器	控制计算机的操作，执行数据处理功能
内存	存储数据和程序。
输入/输出模块	在计算机和外部环境之间传送数据
系统总线	在处理器、内存和输入/输出模块间通信的设施。
微处理器	一种处理器，其所有组件小型化至一块或数块集成电路内。
中断	计算机允许其他模块中断处理器正常处理过程的机制

## 1.1 基本构成



## 1.2 微处理器的发展

**多核和多处理器的区别？** 多个处理器指的是多个单核处理器，多核处理器指的是单个芯片上有多个 CPU。我们可以看双核处理器和多处理器系统的概念图。

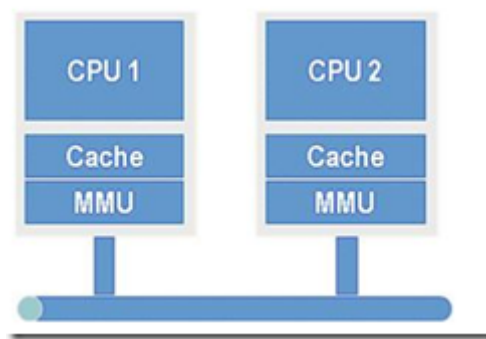


图 1: 多处理器

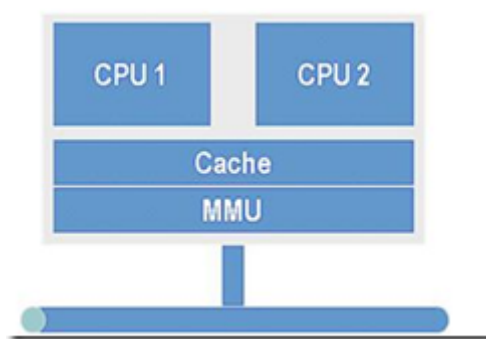


图 2: 双核处理器

### 1.3 中断

中断：计算机允许其他模块中断处理器正常处理过程的机制。

表 2: 中断的分类

程序中断	在某些条件下由指令执行的结果产生，如算术溢出、除数为 0、试图执行一条非法机器指令及访问用户不允许的存储器位置
时钟中断	由处理器内部的计时器产生，允许操作系统以一定的规律执行函数
I/O 中断	由 I/O 控制器产生，用于发信号通知一个操作的正常完成或各种错误条件
硬件失效中断	由诸如掉电或存储器奇偶校验错之类的故障产生

#### 1.3.1 中断和指令周期

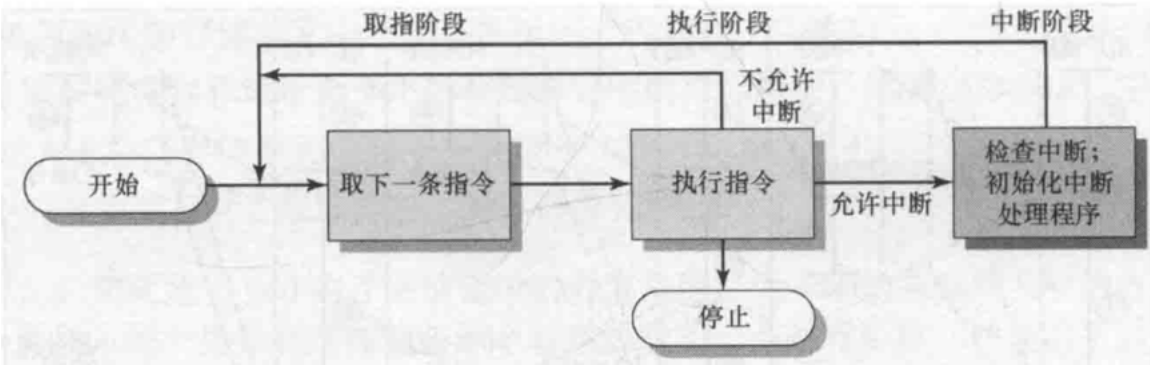


图 3: 中断和指令周期

### 1.3.2 中断处理

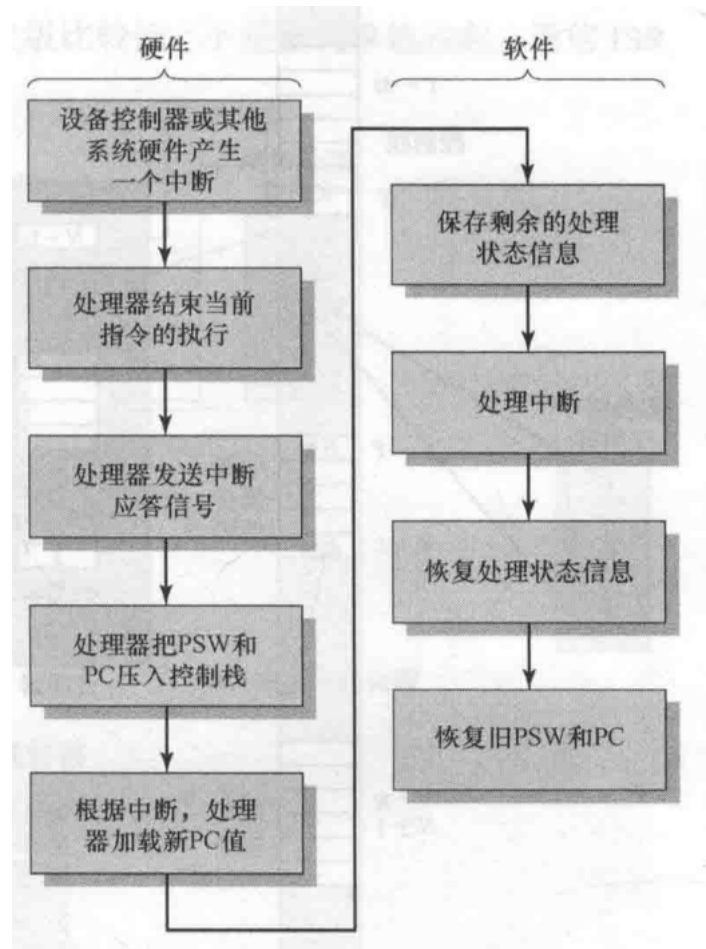


图 4: 简单中断处理

### 1.3.3 多个中断

也就是说中断里面套中断。

解决多个中断有两种方法，一种是不允许多个中断，另一种是第一中断优先级。

## 1.4 存储器的层次结构

## 1.5 多处理器和多核计算机组织结构

对称多处理器（SMP）：

- 1. 具有两个或两个以上可比性能的处理器。
- 2. 这些处理器共享内存和 I/O 设备，并通过总线或其他内部连接方式互连，因此每个处理器的访存时间大体相同。

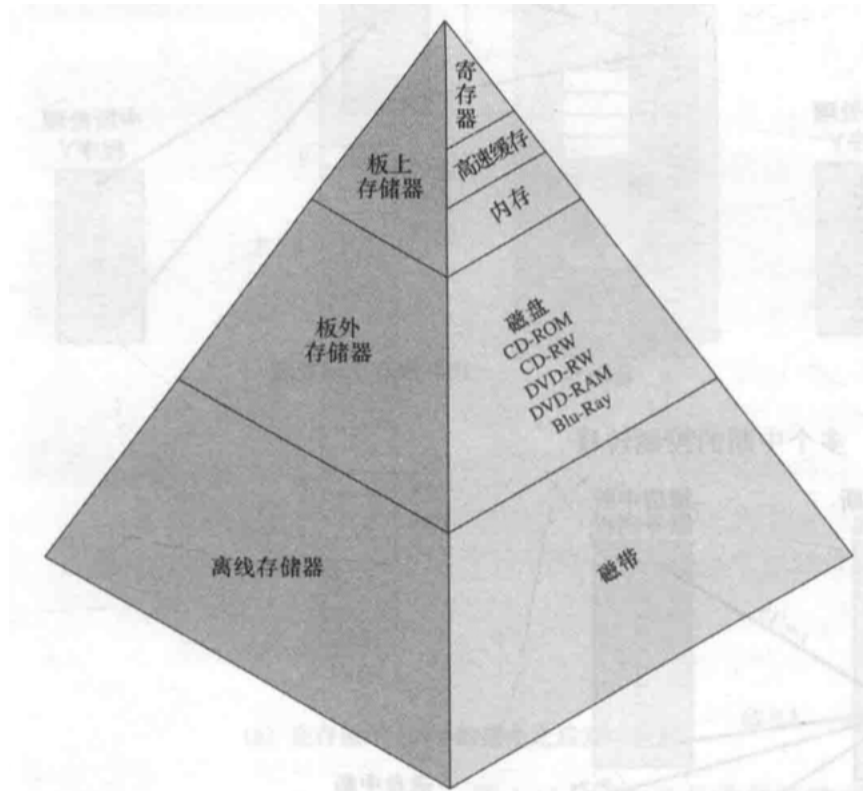


图 5: 存储器层次结构

- 3. 所有处理器共享对 IO 设备的访问，要么通过相同的通道，要么通过可以连接到相同设备的不同通道。
- 4. 所有处理器可以执行相同的功能 (因此是对称的)。
- 5. 整个系统由一个统一的操作系统控制，该操作系统为多个处理器及其程序提供作业、进程、文件和数据元素等各种级别的交互。



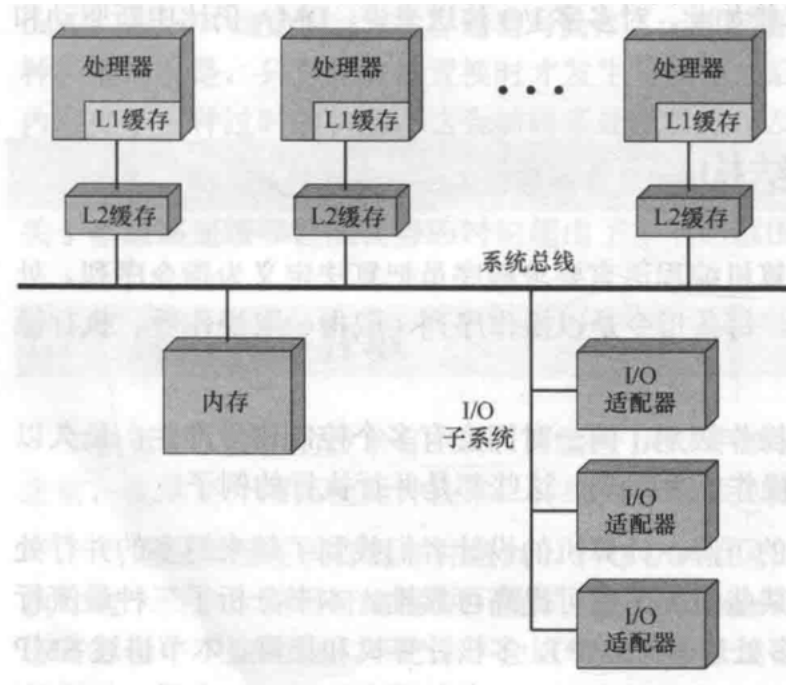


图 6: SMP 的组织结构

## 2 操作系统概述

### 2.1 操作系统的目标和功能

- 作为用户/计算机接口
- 作为资源管理器
- 扩展的功能

指令系统体系结构 (ISA): 定义了计算机遵循的机器语言指令系统, 该接口是硬件与软件的分界线。注意, 应用程序和实用程序都可直接访问 ISA, 这些程序使用指令系统的一个子集 (用户级 ISA)。操作系统能使用其他一些操作系统资源的机器语言指令 (系统级 ISA)。

应用程序二进制接口 (ABI): 这种接口定义了程序间二进制可移植性的标准。ABI 定义了操作系统的系统调用接口, 以及在系统中通过 ISA 能使用的硬件资源和服务。

应用程序编程接口 (API): API 允许应用程序访问系统的硬件资源和服务, 这些服务由用户级 ISA 和高级语言库 (HLL) 调用来提供。使用 API 能让应用软件更容易重新编译并移植到具有相同 API 的其他系统中。

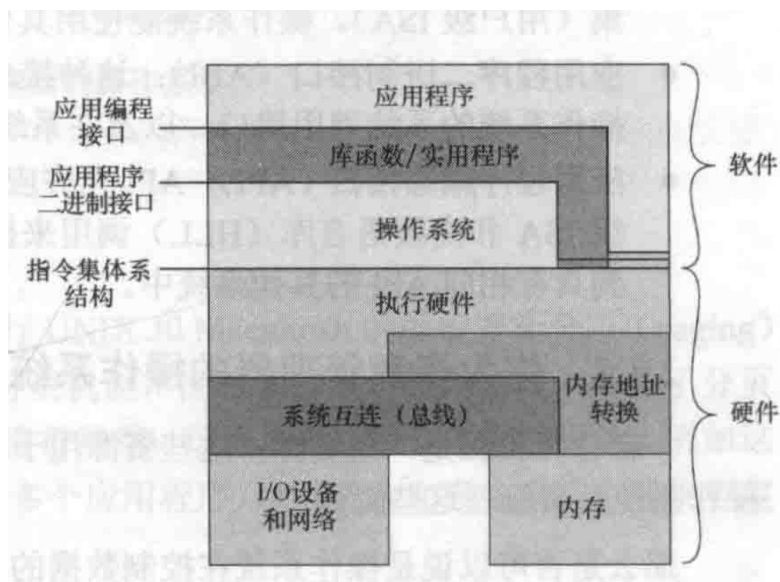


图 7: 计算机硬件和软件结构

### 3

1. 现代操作系统的两个最基本的特征是并发和共享。
2. 操作系统的五大功能是: 处理器管理、存储器管理、文件管理 (信息管理)、设备管理和提供用户接口 (有的书上还提到了作业管理)。
3. 从系统观点看, 操作系统是计算机系统中的一个系统软件, 它管理和控制计算机系统资源; 从用户观点看, 操作系统是用户与计算机之间的口; 从软件观点看, 操作系统是程序和数据结构的集合。
4. 允许多个用户以交互方式使用计算机的操作系统称为分时操作系统; 允许多个用户将多个作业交给计算机集中处理的操作系统称为批处理操作系统; 能及时处理过程、控制数据并做出相应操作的计算机系统称为实时操作系统。
5. 批处理操作系统的主要缺点是缺少交互性。虽然批处理操作系统提高了计算机系统的工作效率, 但用户不能直接干预作业的执行。
6. 实时操作系统应具有两个基本特征: 及时性和高可靠性, 即实时操作系统必须在规定时间内处理完来自外部的的事件并且保证系统的安全性和稳定性, 因此实时操作系统常用在订票系统、银行系统等对反馈时间及可靠性要求高的地方。
7. 操作系统中采用多道程序设计技术后, 提高了 CPU 和外部设备 (简称外设) 的利用率。多道运行的特征之一是宏观上并行, 其含义是同时进入系统的几道程序都处于运行状态, 即它们先后开始了各自的运行, 但均未运行完毕。

8. 多道程序设计技术是指将一个以上的作业放入主存, 并且使之同时处于运行状态。这些作业共享处理器和外设等资源。

9. 操作系统程序的运行状态称为内核态 (或管态), 用户程序的运行状态称为用户态 (或目态)。

10. 通常异常会引起中断, 而中断未必是由异常引起的。

11. 系统中与资源有关的操作, 都必须通过系统调用向操作系统提出请求, 由操作系统代为完成。

### 3.1 计算机系统的层次结构

计算机的硬件、软件以及软件的各部分之间是一种层次结构的关系。硬件在最底层, 其上层是操作系统, 通过操作系统提供的资源管理功能和方便用户使用的各种服务功能, 把裸机改造成功能更强大、使用更方便的机器 (通常称为虚拟机或扩展机)。而各种实用程序和应用程序在操作系统之上, 这些程序均以操作系统为支撑, 并向用户提供完成工作所需的各种服务。

操作系统是裸机上的第一层软件, 是对硬件功能的首次扩充。引入操作系统的目的是: 提供一个计算机用户与计算机硬件系统之间的接口, 使计算机系统更易于使用; 有效地控制和管理计算机系统中的各种硬件和软件资源, 使之得到更有效的利用; 合理地组织计算机系统的工作流程, 以改善系统性能。



图 8: 计算机系统的层次关系

### 3.2 操作系统的概念

#### 3.2.1 用户观点

操作系统的用户观点即根据用户所使用计算机的不同而设计不同类型的操作系统。比如, 大多数人使用的是个人计算机 (PC), 此类计算机主要包括主机、显示器、键盘等, 这种系统设计是为了使用户更好地进行单人工作, 因此操作系统要达到的目的就是方便用户使用, 资源利用率显得不是很重要。而有些用户使用的是大型机或者其终端等, 此类计算机用来完成大型计算或作为公共服务器等工作, 因此其操作系统的设计目的就是使资源利用最大化, 确保所有资源都能够被充分使用,

并且保障稳定性。而智能手机的操作系统所追求的则是界面友好、使用便捷及耗电量低等。

### 3.2.2 系统观点

从资源管理的角度来看，操作系统是计算机系统的资源管理程序。在计算机系统中有两类资源：硬件资源和软件资源。按其作用又可以将它们分为 4 大类资源：处理器、存储器、外设和信息（程序和数据）。这 4 类资源构成了操作系统本身和用户作业赖以活动的物质基础和工作环境。它们的使用方法和策略决定了整个操作系统的规模、类型、功能和实现。与上述 4 类资源相对应，操作系统可被划分成处理器管理、存储器管理、设备管理和信息管理（即文件系统），并分别进行分析研究。由此，可以用资源管理的观点组织操作系统的有关内容。

### 3.2.3 进程观点

这种观点把操作系统看作由若干个可以独立运行的程序和一个对这些程序进行协调的核心所组成的。这些运行的程序称为进程，每个进程都完成某一项特定任务（如控制用户作业的运行，处理某个设备的输入/输出....）。而操作系统的核心则是控制和协调这些进程的运行，解决进程之间的通信；它从系统各部分以并发工作为出发点，考虑管理任务的分割和相互之间的关系，通过进程之间的通信来解决共享资源时所带来的竞争问题。通常，进程可以分为用户进程和系统进程两大类，由这两类进程在核心控制下的协调运行来完成用户的要求。

### 3.2.4 虚拟机观点

虚拟机的观点也称为机器扩充的观点。从这一观点来看，操作系统为用户使用计算机提供了许多服务功能和良好的工作环境。用户不再直接使用硬件机器（称为裸机），而是通过操作系统来控制和使用计算机。计算机从而被扩充为功能更强大、使用更加方便的虚拟计算机。

## 3.3 操作系统的特征

### 3.3.1 并发性

并发性和并行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，并发性是指宏观上在一段时间内有多道程序在同时运行，但在单处理器系统中，每一时刻仅有一道程序在执行，故微观上这些程序是交替执行的。

### 3.3.2 共享性

资源共享是指系统中的硬件和软件资源不再为某个程序所独占，而是供多个用户共同使用。并发和共享是操作系统的两个最基本的特征，二者之间互为存在条件。一方面，资源的共享是以程序的并发执行为条件的，若系统不允许程序的并发执行，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的管理，也必将影响到程序的并发执行，甚至根本无法并发执行。

### 3.3.3 虚拟性

在操作系统中，虚拟是指把一个物理上的实体变为若干个逻辑上的对应物，前者是实际存在的，后者是虚拟的，这只是用户的一种感觉。例如，在操作系统中引入多道程序设计技术后，虽然只有一个 CPU，每次只能执行一道程序，但通过分时使用，在一段时间间隔内宏观上这台处理器能同时运行多道程序。它给用户的感觉是每道程序都有一个 CPU 为其服务。也就是说，多道程序设计技术可以把一台物理上的 CPU 虚拟为多台逻辑上的 CPU。此外还有虚拟存储器（从逻辑上扩充存储器的容量）、虚拟设备（独占设备变为共享设备）等技术。

### 3.3.4 异步性

在多道程序环境中，由于资源等因素的限制，程序是以“走走停停”的方式运行的。系统中的每道程序何时执行、多道程序间的执行顺序以及完成每道程序所需的时间都是不确定的，因而也是不可预知的。

## 3.4 操作系统的主要功能

### 3.4.1 处理器管理

处理器管理的主要任务是对处理器的分配和运行实施有效的管理。在多道程序环境下，处理器的分配和运行是以进程为基本单位的，因此对处理器的管理可归结为对进程的管理。进程管理应实现下述主要功能：

进程控制。负责进程的创建、撤销及状态转换。

进程同步。对并发执行的进程进行协调。

进程通信。负责完成进程间的信息交换。

进程调度。按一定算法进行处理器分配。

### 3.5 存储器管理

存储器管理的主要任务是对内存进行分配、保护和扩充。存储器管理应实现下述主要功能:

内存分配。按一定的策略为每道程序分配内存。

内存保护。保证各程序在自己的内存区域内运行而不相互干扰。

内存扩充。为允许大型作业或多作业的运行,必须借助虚拟存储技术去获得增加内存的效果。

### 3.6 设备管理

计算机外设的管理是操作系统中最庞杂、琐碎的部分。设备管理的主要任务是对计算机系统内的所有设备实施有效管理。设备管理应具有下述功能:

设备分配。根据一定的设备分配原则对设备进行分配。为了使设备与主机并行工作,还需采用缓冲技术和虚拟技术。

设备传输控制。实现物理的输入输出操作,即启动设备、中断处理、结束处理等。

设备独立性。即用户程序中的设备与实际使用的物理设备无关。

#### 3.6.1 文件管理

操作系统中负责信息管理的部分称为文件系统,因此称为文件管理。文件管理的主要任务就是有效地支持文件的存储、检索和修改等操作,解决文件的共享、保密和保护问题。文件管理应实现下述功能:

文件存储空间的管理。负责对文件存储空间进行管理,包括存储空间的分配与回收等功能。

目录管理。目录是为方便文件管理而设置的数据结构,它能提供按名存取的功能。

文件操作管理。实现文件的操作,负责完成数据的读写。

文件保护。提供文件保护功能,防止文件遭到破坏。

#### 3.6.2 用户接口

为方便用户使用操作系统,操作系统还提供了用户接口。通常,操作系统以如下 3 种接口方式提供给用户使用。

命令接口。提供一组命令供用户直接或间接控制自己的作业。主要有两种命令接口控制方式,即联机命令接口和脱机命令接口。

联机命令接口又称交互式命令接口,适用于分时或实时操作系统,它由一组键盘操作命令组成,用户通过控制台或终端输入操作命令,向系统提出各种服务要求,用户每输入完一条命令,控

制权就转入操作系统的命令解释程序，然后由命令解释程序对输入的命令解释并执行，完成执行的功能。之后控制权又转回到控制台或终端，此时用户又可以输入下一条命令。

脱机命令接口又称批处理命令接口，即适用于批处理系统，它由一组作业控制命令(或称作业控制语句)组成，脱机用户不能直接干预作业的运行，应事先用相应的作业控制命令写成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控制语句逐条解释执行从而间接地控制作业的运行。

程序接口。也称为系统调用，是程序级的接口，由系统提供一组系统调用命令供用户程序和其他系统程序调用。用户在程序中可以直接使用这组系统调用命令向操作系统提出各种服务要求，如使用外设、申请分配内存、磁盘文件的操作等。

图形接口。近年来出现的图形接口(也称图形界面)是联机命令接口的图形化。

由操作系统的功能可以知道操作系统提供哪些服务: 操作系统提供了一个用以执行程序的环境，提供的服务有程序执行、I/O 操作、文件操作、资源分配与保护、错误检测与排除等。

## 3.7 操作系统的发展

### 3.7.1 无操作系统阶段

在第一代计算机时期，构成计算机的主要元器件是电子管，计算机运行速度慢，没有操作系统，甚至没有任何软件，人们采用手工操作方式操作计算机。在手工操作方式下，用户一个接一个地轮流使用计算机，每个用户的使用过程大致如下: 先将程序纸带(或卡片)装入输入机，然后启动输入机把程序和数据送入计算机，接着通过控制台开关启动程序运行，当程序运行完毕后，由用户取走纸带和结果。由此可以推断，这种操作方式具有用户独占计算机资源、资源利用率低以及 CPU 等待等人工操作的特点。随着 CPU 速度的大幅提高，手工操作的慢速与 CPU 运算的高速之间出现了矛盾，这就是所谓的人机矛盾。此外，CPU 和 I/O 设备之间速度不匹配的矛盾也日益突出。为了缓和此矛盾，先后出现了通道技术和缓冲技术，但都未能很好地解决上述矛盾，直到后来引入脱机输入/输出技术，才获得了较为满意的效果。脱机输入/输出技术是为了解决 CPU 和 I/O 设备之间速度不匹配的矛盾而提出的，此技术减少了 CPU 的空闲等待时间，提高了 I/O 速度。其输入/输出方式如图 1-3 所示。为解决低速输入设备与 CPU 速度不匹配的问题，可以将用户程序和数据在一台外围机(又称卫星机)的控制下，预先从低速输入设备(纸带机)输入到输入带上，当 CPU 需要这些程序和数据时，再直接从输入带高速输入到内存，从而大大加快输入速度，减少 CPU 等待输入的时间，这就是脱机输入技术。

类似地，当程序运行完毕或告一段落，当 CPU 需要输出时，无须直接把计算结果送至低速输出设备(图 1-3 中为打印机)，而是高速地把结果送到输出带上，然后在外围机的控制下，把磁带上

的计算结果由相应的输出设备输出，这就是脱机输出技术。若输入/输出操作在主机控制下进行，则称为联机输入/输出。采用脱机输入/输出技术后，低速 I/O 设备上数据的输入/输出都在外围机的控制下进行，而 CPU 只与高速的输入带及输出带打交道，从而有效地减少了 CPU 等待慢速设备输入/输出的时间。详细说明本方法的目的在于使考生了解脱机输入/输出的模型，因为之后的缓冲区技术以及 SPOOLing 技术等，都是基于这种原理产生的。理解了 this 模型，对之后学习类似技术有较大的帮助。

### 3.7.2 单道批处理系统

单道批处理系统是最早出现的一种操作系统，严格地说，它只能算作是操作系统的前身而并非是目前人们所理解的操作系统。

早期的计算机系统非常昂贵，为了能充分利用，应尽量使系统连续运行，以减少空闲时间。为此，通常是把一批作业以脱机输入方式输入到磁带上，并在系统中配置监督程序 (管理作业的运行，负责装入和运行各种系统程序来完成作业的自动过渡)，在其控制下，先把磁带上的第一个作业传送到内存，并把运行的控制权交给第一个作业，当第一个作业处理完后又把控制权交还给监督程序，由监督程序再把第二个作业调入内存。计算机系统按这种方式对磁带上的作业自动地一个接一个进行处理，直至把磁带上的所有作业全部处理完毕，这样便形成了早期的批处理系统。

### 3.7.3 多道批处理系统

为进一步提高 CPU 的利用率，引入了多道程序设计技术，由此而形成了多道批处理系统。多道程序设计技术是“将一个以上的作业存放在主存中，并且同时处于运行状态。这些作业共享处理器、外设以及其他资源”。现代计算机系统一般都基于多道程序设计技术。

## 3.8 操作系统的分类

### 3.8.1 批处理操作系统

描述任何一种操作系统都安用到“作业”的概念。所谓作业，就定用户在一次解题或一个事务处理过程中要求计算机系统所做工作的集合，包括用户程序、所需的数据及命令等。单道批处理操作系统是早期计算机系统中配置的一种操作系统类型。其工作流程大致如下：系统操作员将用户作业收集起来，并将这些作业组成一批输入并传送到外存。批处理操作系统每次将其中的一个作业调入运行，同时只有一道作业处于运行状态，运行完成或出现错误而无法再进行下去时，输出有关信息并调入下一个作业运行。如此反复处理，直到这一批作业全部处理完毕为止。单道批处理操作系统大大提高了机器的利用率，减少了人工操作的时间。但是对于某些作业来说，当其发出



输入输出请求后，CPU 必须等待 I/O 的完成，这就意味着 CPU 空闲，特别是当 I/O 设备的速度较慢时，将导致 CPU 的利用率很低。为了提高 CPU 的利用率，引入了多道程序设计技术。在单道批处理操作系统中引入多道程序设计技术，就形成了多道批处理操作系统。在多道批处理操作系统中，不仅在主存（也称内存）中可以同时有多道作业运行，而且作业可随时（不一定集中成批）被接受进入系统，并存放在外存中形成作业队列，然后由操作系统按一定的原则从作业队列中调度一个或多个作业进入主存运行。多道批处理操作系统一般用于计算中心的大型计算机系统。

多道批处理操作系统的主要特点如下：

用户脱机使用计算机。用户提交作业之后，在获得结果之前几乎不和计算机交互。

成批处理。工作人员把用户提交的作业分批进行处理，由监督程序负责每批作业间的自动调度。

多道程序运行。按多道程序设计的调度原则，从一批后备作业中选取多个作业调入内存并组织其运行，成为多道批处理系统。由于多道批处理系统中的资源为多个作业所共享，作业之间自动调度执行，并且在运行过程中用户不干预自己的作业，从而大大提高了系统资源利用率和作业吞吐量。其不足之处是无交互性，一旦提交作业，用户就失去了对其运行的控制能力，使用不方便。

### 3.8.2 分时操作系统

在批处理操作系统中，用户以脱机操作方式使用计算机，在提交作业后，用户就完全脱离了自己的作业，在作业运行过程中，不管出现什么情况都不能加以干预，只有等待该批处理作业处理结束，用户才能得到计算结果，根据计算结果再做下一步处理。若作业运行出错，还要重复上述过程。这种操作方式对用户而言是极不方便的，人们希望能以联机方式使用计算机（即交互性更好），这种需求导致了分时操作系统的产生。所谓分时技术，就是把处理器的运行时间分成很短的时间片，按时间片轮流把处理器分配给各联机作业使用。若某个作业在分配给它的时间片内不能完成其计算，则该作业暂时停止运行，把处理器让给另一个作业使用，等下一轮时再继续运行。由于计算机速度很快，作业运行轮转也很快，给每个用户的感受都好像是自己独占一台计算机。在操作系统中采用分时技术就形成了分时操作系统。在分时操作系统中，一台计算机和许多终端设备连接，用户可以通过终端向系统发出命令，请求完成某项工作，而系统则分析从终端设备发来的命令，完成用户提出的要求，然后用户再根据系统提供的运行结果，向系统提出下一步请求，这样重复上述交互会话过程，直到用户完成预计的全部工作为止。实现分时操作系统有下述几种方法：简单分时操作系统。在简单分时操作系统中，内存只驻留一道作业，其他作业都在外存上。每当内存中的作业运行一个时间片后，便被调至外存（称为调出），再从外存上选一个作业装入内存（称为调入）并运行一个时间片，按此方法使所有作业都能在规定的时间内轮流运行一个时间片，这样，所有用户都能与自己的作业交互。

具有“前台”和“后台”的分时操作系统。为了改善系统性能，引入了“前台”和“后台”的概念。这里，把作业划分为“前台”和“后台”两类。“前台”存放按时间片调入/调出的作业流，其工作方式与简单分时操作系统相同；“后台”存放批处理作业。仅当“前台”正在调入/调出或无调入/调出作业流时，才运行“后台”的批处理作业，并给它分配更长的时间片。

多道分时操作系统。在分时操作系统中引入多道程序设计技术后，内存中可以同时装入多道作业，系统把所有具备运行条件的作业排成一个队列，使它们依次轮流获得一个时间片运行。

分时操作系统具有以下特征。多路性。指一台计算机与若干台终端相连接，终端上的这些用户可以同时或基本同时使用计算机。交互性。分时操作系统中用户的操作方式是联机方式，即用户通过终端采用人机会话的方式直接控制程序运行，同程序进行交互。独占性。由于分时操作系统采用时间片轮转的方法使一台计算机同时为许多终端用户服务（通常能在 2 3s 内响应用户请求），因此客观效果是这些用户彼此之间都感觉不到别人也在使用这台计算机，好像自己独占计算机一样。及时性。系统能够在较短时间内响应用户请求。

### 3.8.3 实时操作系统

实时操作系统是操作系统的又一种类型。对外部输入的信息，实时操作系统能够在规定的时间内处理完毕并做出反应。“实时”的含义是指计算机对于外来信息能够以足够快的速度进行处理，并在被控制对象允许的时间范围内做出快速反应。实时操作系统对响应时间的要求比分时操作系统更高，一般要求秒级、毫秒级甚至微秒级的响应时间。实时操作系统可以分成如下两类：实时控制系统。通常是指以计算机为中心的生产过程控制系统，又称为计算机控制系统。例如，钢铁冶炼和钢板轧制的自动控制，化工、炼油生产过程的自动控制等。在这类系统中，要求实时采集现场数据，并对它们进行及时处理，进而自动控制相应的执行机构，使某参数（如温度、压力、流量等）能按预定规律变化或保持不变，以达到保证产品质量、提高产量的目的。实时信息处理系统。在这类系统中，计算机及时接收从远程终端发来的服务请求，根据用户提出的问题对信息进行检索和处理，并在很短时间内对用户做出正确响应，如机票订购系统、情报检索系统等，都属于实时信息处理系统。实时操作系统的主要特点是提供及时响应和高可靠性。系统必须保证对实时信息的分析和处理的速度要快，而且系统本身要安全可靠，因为诸如生产过程的实时控制、航空订票等实时事务系统，信息处理的延误或丢失往往会带来不堪设想的后果。批处理操作系统、分时操作系统和实时操作系统是 3 种基本的操作系统。若一个操作系统兼有批处理、分时和实时系统或其中两者的功能，则称该操作系统为通用操作系统。

### 3.8.4 嵌入式操作系统

嵌入式操作系统是运行在嵌入式系统环境中，对整个嵌入式系统以及它所操作和控制的各种部件装置等资源进行统一协调、调度、指挥和控制的软件系统。嵌入式操作系统支持嵌入式软件的运行，它的应用平台之一是各种电器，该系统面向普通家庭和个人用户。由于快速发展的市场网络，使得家用电器的市场比传统的计算机市场大很多，因此嵌入式软件可能成为 21 世纪信息产业的支柱之一，嵌入式操作系统也必将成为软件厂商争夺的焦点，成为操作系统发展的另一个热门方向。

### 3.8.5 分布式操作系统

分布式系统是指多个分散的处理单元经互联网络连接而成的系统，其中每个处理单元既具有高度自治性又相互协同，能在系统范围内实现资源管理、动态分配任务，还能并行地运行分布式程序。配置在分布式系统上的操作系统称为分布式操作系统。分布式操作系统具有以下特征：统一性。即它是一个统一的操作系统。共享性。即分布式操作系统中的所有资源是共享的。透明性。是指用户并不知道分布式操作系统是运行在多台计算机上，在用户眼里整个分布式系统像是一台计算机，用户并不知道自己请求系统完成的操作是哪一台计算机完成的，也就是说，系统对用户来讲是透明的。自治性。即分布式操作系统中的多个主机都处于平等地位。分布式操作系统的优点之一是它的分布式：分布式操作系统可以用较低的成本获得较高的运算性能。分布式操作系统的另一个优点是它的可靠性：由于有多个 CPU 系统，因此当一个 CPU 系统发生故障时，整个系统仍旧能够工作。

## 3.9 操作系统的运行环境

### 3.9.1 内核态与用户态

为了避免操作系统及其关键数据 (如 PCB 等) 受到用户程序有意或无意的破坏，通常将处理器的执行状态分为两种：核心态与用户态。

核心态。核心态又称管态、系统态，是操作系统管理程序执行时机器所处的状态。它具有较高的特权，能执行包括特权指令的一切指令，能访问所有寄存器和存储区。

用户态。用户态又称目态，是用户程序执行时机器所处的状态，是具有较低特权的执行状态，它只能执行规定的指令，只能访问指定的寄存器和存储区。

划分核心态与用户态之后，这两类程序以及各自的存储空间被严格区分了，而且在 CPU 执行时有着完全不同的待遇。用户态程序不能直接调用核心态程序，而是通过执行访问核心态的命令，引起中断，由中断系统转入操作系统内的相应程序，例如，在系统调用时，将由用户态转换到核心态。

特权指令: 只能由操作系统内核部分使用, 不允许用户直接使用的指令, 如 I/O 指令、设置中断屏蔽指令、清内存指令、存储保护指令和设置时钟指令。

操作系统中一些与硬件关联较紧密的模块 (如时钟管理、中断处理、设备驱动等) 以及运行频率较高的程序 (如进程管理、存储器管理、设备管理等) 构成了操作系统的内核。内核的指令操作工作在核心态, 主要包括以下 4 个方面的内容。

1) 时钟管理。时钟是计算机的各部件中最关键的设备, 操作系统通过时钟管理, 向用户提供标准的系统时间。另外通过时钟中断的管理, 可以实现进程的切换, 如时间片轮转调度。

2) 中断机制。键盘或鼠标的输入、进程的管理和调度、系统功能的调用、设备驱动、文件访问等, 无不依赖于中断机制。中断机制中, 只有一小部分属于内核, 负责保护和恢复中断现场的信息, 转移控制权到相关的处理程序。这样可以减少终端的处理时间, 提高系统的并行处理能力。

3) 原语。原语是一些关闭中断的公用小程序, 主要有以下特点。处于操作系统最底层, 是最接近硬件的部分。

程序运行具有原子性, 操作只能一气呵成。这些程序的运行时间较短, 调用频繁。

4) 系统控制的数据结构及处理。操作系统中需要一些用来登记状态信息的数据结构, 如作业控制块、进程控制块、设备控制块、各类链表、消息队列、缓冲器、空闲登记区、内存分配表等。除此之外还应该定义对这些数据结构的一系列操作: 进程管理、存储器管理、设备管理。

### 3.9.2 中断与异常

中断与异常是一对类似但又有区别的概念。中断, 也称外中断, 是系统正常功能的一部分, 例如, 因进程调度使系统停止当前运行的进程转而执行其他进程, 或者因缺少所需资源而中断当前操作等待资源到达, 在系统处理完其他事情之后, 会继续执行中断前的进程。而异常, 也称内中断, 是由错误引起的, 如文件损坏、进程越界等。通常异常会引起中断, 而中断未必是由异常引起的。

### 3.9.3 系统调用

系统调用是操作系统提供的用户接口之一, 是由操作系统实现的所有系统调用所构成的集合, 即程序接口或应用编程接口 (Application Programming Interface, API), 是应用程序同系统之间的接口。

操作系统的主要功能是为应用程序的运行创造良好的环境。为了达到这个目的, 内核提供了一系列具备预定功能的内核函数, 通过一组称为系统调用 (System Call) 的接口呈现给用户。系统调用把应用程序的请求传给内核, 调用相应的内核函数完成所需的处理, 并将处理结果返回给应用程序。如果没有系统调用和内核函数, 用户将不能编写大型应用程序。操作系统提供的系统调用通常

包括进程控制、文件系统控制 (文件读写操作和文件系统操作)、系统控制、内存管理、网络管理、socket 控制、用户管理以及进程间通信 (信号、消息、管道、信号量和共享内存)。

操作系统执行系统调用的流程如图 1-6 所示。用户需要执行系统调用时, 首先准备并传递系统调用所需的参数, 通过陷入 (trap) 指令进入操作系统的系统内核, 此时将从用户态进入内核态: 之后执行相应的系统调用函数, 使用特定的系统内核功能; 最后将处理结果返回给用户进程, 此时将从内核态返回用户态。

## 3.10 操作系统的体系结构

### 3.10.1 模块组合结构

模块组合结构是软件工程出现以前的早期操作系统以及目前一些小型操作系统的体系结构。操作系统是一个有多种功能的系统程序, 可以看作一个整体模块, 也可以看作若干个模块按一定的结构方式组成的。系统中的每一个模块都是根据它们要完成的功能来划分的, 这些功能模块按照一定的结构方式组合起来, 协同完成整个系统的功能。优点: 结构紧密、接口简单直接、系统的效率相对较高。缺点: 首先, 这种结构的模块之间可以随意转接, 各模块相互牵连, 不容易把握好模块的独立性, 导致系统结构不清晰。其次, 这种结构的可扩展性较差。在更换一个模块或修改一个模块时, 要先弄清模块间的接口, 如果要按当初设计的模块接口来设计新的模块, 而当初设计的模块接口很可能是随意约定的, 那么要做这项工作就存在一定难度。最后, 这种结构系统的可适应性差。随着系统规模的不断增大, 采用这种结构构造的系统的复杂性会迅速增长, 所以它只适用于系统小、模块少、使用环境比较稳定的系统。

### 3.10.2 层次结构

要弥补模块组合结构中模块间调用存在的不足之处, 就必须改善模块间毫无规则的相互调用、相互依赖的关系, 尤其要清除模块间的循环调用。层次结构的设计就是从这一点出发, 力求使模块之间调用的无序性变为有序, 减少了模块调用的无规则性。按层次结构来设计操作系统, 就是将操作系统的所有功能模块按功能的调用次序排列成若干层, 使得功能模块之间只存在单向调用和单向依赖。优点: 模块间的组织和依赖关系清晰明了, 上层功能是建立在下层功能基础之上的, 系统的可读性、可适应性以及可靠性都得到了增强。此外, 对某一层进行修改或替换时, 最多只影响到邻近的两层, 便于修改和扩充。缺点: 操作系统的各个功能模块应该放在哪一层, 如何有效地进行分层是必须要考虑的问题。为了增强其适应性, 必须把与机器特点紧密相关的软件 (如中断处理、输入/输出管理等) 放在最底层; 其次, 要将最常用的操作方式放在最内层, 而把随着这些操作方式改变的部分放在外层。另外, 当前操作系统的设计都是基于进程的概念, 通常要将为进程提供服务

的系统调用模块放在系统的内层。

### 3.10.3 微内核结构

随着网络技术的普遍应用和发展,很有必要为用户提供一个符合处理分布式信息的分布式系统环境。因此,操作系统可以采用微内核结构。微内核的主要思想是:在操作系统内核中只留下一些最基本的功能,而将其他服务尽可能地从内核中分离出去,用若干个运行在用户态下的进程(即服务器进程)来实现,形成所谓的“客户/服务器”模式,即 C/S 模式。普通用户进程(即客户进程)可通过内核向服务器进程发送请求,以取得操作系统的服务。从微内核结构的主要思想可以看出,它非常适用于分布式系统。优点:首先,每个服务进程运行在独立的用户进程中,即便某个服务器失败或产生问题,也不会引起系统其他服务器和其他组成部分的崩溃,可靠性好;其次,系统具有很好的灵活性,只要接口规范,操作系统可以方便地增删服务功能;再次,便于维护,即修改服务器的代码不会影响系统其他部分;最后,这种结构的操作系统适合分布式处理的计算环境。缺点:这种结构的操作系统效率不高,因为所有用户进程都要通过微内核相互通信,所以微内核本身就成为了系统的“瓶颈”,尤其是通信频繁的系统。

“虚拟”体现在操作系统的各方面应用当中,请举出两个“虚拟”的例子。虚拟”体现在操作系统的各方面应用当中,请举出两个“虚拟”的例子。

由于一台计算机配置了操作系统和其他软件,因此比一台裸机功能更强大,使用更方便,称为虚拟机。由于操作系统自身包含了若干层软件,因此该计算机系统又可称为多层虚拟机。

如在多道分时系统中,利用多道程序设计技术可以把一台物理上的 CPU 虚拟为多台逻辑上的 CPU,而供多个终端用户使用。

虚拟存储器,仅把作业的一部分装入内存便可运行作业,从逻辑上对内存容量进行了扩充。又如在设备管理中虚拟设备技术的使用,可将一台物理设备变换为若干台逻辑上的对应物。

## 4 进程管理

1.进程是一个程序对某个数据集的一次运行活动。进程是动态的概念,而程序是静态的概念。  
2.进程的基本特征是:动态性、并发性、独立性、异步性和结构特征。3.程序段和数据段称为进程的实体,用 PCB 描述实体的存在和变化。从结构上讲,进程包括程序段、数据段和进程控制块(PCB)。4.在操作系统中引入线程概念的主要目的是减少程序并发执行时所需付出的时空开销,提高程序执行的并发程度。5.进程的基本状态有运行、就绪和阻塞。引起状态转化的事件通常有:时间片用完(运行转就绪)、等待事件(运行转阻塞)、等待条件发生(阻塞转就绪)以及进程调度(就绪转运行)。6.三级调度的运行频率:高级调度 < 中级调度 < 低级调度。7.进程的调度方法有两类:抢

占式与非抢占式。若要使当前运行进程总是优先级最高的进程，则应该选择抢占式优先级调度算法。时间片轮转调度方法也是抢占式调度方法。8.进程调度算法采用等时间片轮转法时，若时间片过大，则会使轮转法转化为先来先服务调度算法；若时间片过小，则会在进程切换时消耗过多的系统资源和时间，效率反而会降低。9.在所有调度算法中，短作业优先调度算法的平均等待时间最短。10.同时只能一个进程使用的资源称为临界资源。在进程中，访问临界资源的代码段称为临界区。为保证进程互斥访问临界资源，应在进程的临界区之前设置进入区，在临界区后设置退出区。11.访问临界资源应遵循的准则为：空闲让进、忙则等待、有限等待、让权等待。12.同步是指不同进程间相互合作、相互等待，互斥是指同类进程需要互斥使用资源。同步与互斥描述了进程间的两种不同制约关系。13.信号量的物理意义是：当信号量值大于零时，表示可用资源的数目；当信号量值小于零时，其绝对值为在该信号量上等待的进程个数。14.用P、V操作管理临界区时，任何一个进程在进入临界区之前应调用P操作，退出临界区时应调用V操作。15.死锁产生的4个必要条件是：互斥、请求与保持（部分分配）、不剥夺和环路等待。要防止死锁的发生，可以破坏这4个必要条件之一，但破坏互斥条件是不太实际的。16.在有m个进程的系统中出现死锁时，死锁进程的个数k应该满足的条件是 $2k \leq m$ 。某系统中有3个并发进程，都需要同类资源4个，该系统不会发生死锁的最少资源个数是10。因为无论如何分配，总能有一个进程获得足够资源运行结束并释放所持有的资源，使其他进程也得到足够的资源。17.不让死锁发生的策略可以分为静态和动态两种，死锁避免属于动态策略。18.在可共享系统资源不足时，可能出现死锁。但是，不适当的进程推进顺序也可能产生死锁。19.预先静态分配法破坏了死锁产生必要条件中的请求与保持条件，资源剥夺法和撤销进程的方法破坏了不剥夺条件，资源的按序分配策略可以破坏环路等待条件。

## 4.1 进程

在多道程序环境下，程序的并发执行破坏了程序的封闭性和可再现性，使得程序和计算不再一一对应，程序活动不再处于一个封闭系统中，程序的运行出现了许多新的特征。在这种情况下，程序这种静态概念已经不能如实地反映程序活动的这些特征，为此引入了一个新的概念——进程。1. 进程的定义进程的概念从提出之后，许多人都对进程有过各式各样的定义，这里给出几种比较容易理解又能反映进程实质的定义：进程是程序在处理器上的一次执行过程。进程是可以和别的进程并行执行的计算。进程是程序在一个数据集合上的运行过程，是系统进行资源分配和调度的一个独立单位。进程可定义为一个数据结构及能在其上进行操作的程序。进程是一个程序关于某个数据集合在处理器上顺序执行所发生的活动。上述这些描述从不同角度对进程进行了阐述，尽管各有侧重，但本质是相同的。2. 进程的特征进程具有以下几个基本特征：动态性。进程是程序在处理器上的一次执行过程，因而是动态的。动态特性还表现在它因创建而产生，由调度而执行，因得不到资源而暂停，最后因撤销而消亡。并发性。并发性是指多个进程同时存在于内存中，能在一段时间

内同时运行。引入进程的目的是使程序能与其他程序并发执行,以提高资源利用率。独立性。进程是一个能独立运行的基本单位,也是系统进行资源分配和调度的独立单位。异步性。异步性是指进程以各自独立的、不可预知的速度向前推进。结构特征。为了描述和记录进程的运动变化过程,并使之能正确运行,应为每个进程配置一个进程控制块 (Process Control Block, PCB)。这样从结构上看,每个进程都由程序段、数据段和一个进程控制块组成。3.进程和程序的关系进程和程序是两个密切相关但又有所不同的概念,它们在以下几个方面存在区别和联系。进程是动态的,程序是静止的。进程是程序的执行,每个进程包含了程序段和数据段以及进程控制块 (PCB),而程序是有序代码的集合,无执行含义。进程是暂时的,程序是永久的。进程是一个状态变化的过程,程序可以长久保存。进程与程序的组成不同。进程的组成包括程序段、数据段和进程控制块。通过多次执行,一个程序可以产生多个不同的进程:通过调用关系,一个进程可以执行多个程序。进程可创建其他进程,而程序不能形成新的程序。进程具有并行特性(独立性、异步性),程序则没有。

补充知识点:什么是进程映像?进程映像与进程的关系是什么?解析:由程序段、相关数据段和 PCB 三部分构成了进程映像,也叫进程实体。进程映像是静态的,进程是动态的,进程是进程实体的运行过程。

4. 进程和作业的区别作业是用户需要计算机完成某项任务而要求计算机所做工作的集合。一个作业的完成要经过作业提交、作业收容、作业执行和作业完成 4 个阶段。而进程是已提交完毕的作业的执行过程,是资源分配的基本单位。两者的主要区别如下:作业是用户向计算机提交任务的任务实体。在用户向计算机提交作业之后,系统将它放入外存中的作业等待队列中等待执行;而进程则是完成用户任务的执行实体,是向系统申请分配资源的基本单位。任一进程,只要它被创建,总有相应的部分存在于内存中。一个作业可由多个进程组成,且必须至少由一个进程组成,但一个进程不能构成多个作业。作业的概念主要用在批处理系统中。像 UNIX 这样的分时系统则没有作业的概念;而进程的概念则用在几乎所有的多道程序系统中。5. 进程的组成进程一般由以下几个部分组成:进程控制块 (PCB)。每个进程均有一个 PCB,它是一个既能标识进程的存在、又能刻画执行瞬间特征的数据机构。当进程被创建时,系统为它申请和构造一个相应的 PCB。程序段。程序段是进程中能被进程调度程序调度到 CPU 上执行的程序代码段,能实现相应的特定功能。数据段。一个进程的数据段可以是进程对应的程序加工处理的原始数据,也可以是程序执行时产生的中间或结果数据。系统根据 PCB 感知进程的存在。PCB 是进程存在的唯一标志。一般来说,根据操作系统的要求不同,PCB 所包含的内容多少会有些不同,但通常都包括下面所列出的内容。进程标识符 (PID)。每个进程都有唯一的进程标识符,以区别于系统内部的其他进程。在创建进程时,由系统为进程分配唯一的进程标识号。在 Windows 7 系统下,打开任务管理器,依次单击“查看”“选择列”,勾选“PID (进程标识符)”,即可在任务管理器中查看到进程 PID 信息,通常是纯数字。这里看到的 PID 是内部标识符,为了区别于外部标识符。进程当前状态。说明进程的当前状态,以作



为进程调度程序分配处理器的依据。进程队列指针。用于记录 PCB 队列中下一个 PCB 的地址。系统中的 PCB 可能组织成多个队列，如就绪队列、阻塞队列等。程序和数据地址。指出进程的程序和数据所在的地址。进程优先级。反映进程要求 CPU 的紧迫程度。优先级高的进程可以优先获得处理器。CPU 现场保护区。当进程因某种原因释放处理器时，CPU 现场信息 (如指令计数器、状态寄存器、通用寄存器等) 被保存在 PCB 的该区域中，以便该进程重新获得处理器后能继续执行。通信信息。记录进程在执行过程中与别的进程所发生的信息交换情况。家族联系。有的系统允许进程创建子进程，从而形成一个进程家族树。在 PCB 中，本进程与家族的关系是必须指明的，如它的子进程与父进程的标识。占有资源清单。进程所需资源及当前已分配资源清单。在一个系统中，通常存在着很多进程，有的处于就绪状态，有的处于阻塞状态，而且阻塞的原因各不相同。为了方便进程的调度和管理，需要将各进程的 PCB 用适当的方法组织起来。目前常用的组织方式有链接方式和索引方式。

为什么说 PCB 是进程存在的唯一标志? 首先来看 PCB 的作用: PCB 是系统为每个进程定义的一个数据结构，其作用是使程序 (含数据) 能独立运行; PCB 使一个在多道程序环境下不能独立运行的程序 (含数据) 成为一个能独立运行的基本单位，一个能与其他进程并发执行的进程，因此 PCB 是为了保证程序的并发执行。创建进程，实质上是创建进程的 PCB; 而撤销进程，实质上是撤销进程的 PCB。其次来解释为什么 PCB 是进程存在的唯一标志。在系统调度到某进程后，要根据其 PCB 中所保存的处理机状态信息，设置该进程恢复运行的现场，并根据其 PCB 中的程序和数据内存地址，找到其程序和数据; 进程在执行过程中，当需要和与之合作的进程实现同步、通信或访问文件时，也都需要访问 PCB; 当进程由于某种原因而暂停执行时，又需将其断点的处理机环境保存在 PCB 中。可见，在进程的整个生命期中，系统总是通过 PCB 对进程进行控制的，亦即系统是根据进程的 PCB 感知该进程的存在的，所以，PCB 是进程存在的唯一标志。