

高等职业教育教材丛书

GAODENG ZHIYE
JIAOYU JIAOCAI CONGSHU

操作系统 概论

实习指导与模拟试题

边莫英 主编
韩劼 王保旗 孙华志 编著



南开大学出版社

高等职业教育教材丛书

操作系统概论

实习指导与模拟试题

边奠英

主编

韩昉 王保旗 孙华志

编著

南开大学出版社

天 津

内容提要

本书是高等职业教育教材《操作系统概论》一书的配套辅导读物。全书分为三编。
第一编为课程基本要求,按照考试大纲的要求,叙述教材各章节的知识点、难点、考点。
第二编为实习指导,包括 DOS、Windows、Linux 三个操作系统学习中的配套实习。
第三编为模拟试题,提供了各章节对应的典型试题分析与模拟试题。
本书可供学习上述教材的读者自学、参考。

图书在版编目(CIP)数据

操作系统概论实习指导与模拟试题/韩勔,王保旗,
孙华志编著. —天津:南开大学出版社,2004.3
(高等职业教育教材丛书/边莫英主编)
ISBN 7-310-02047-2

I. 操... II. ①韩... ②王... ③孙... III. 操作系
统一高等学校:技术学校—教学参考资料 IV. TP316

中国版本图书馆 CIP 数据核字(2003)第 122189 号

出版发行 南开大学出版社

地址:天津市南开区卫津路 94 号 邮编:300071

营销部电话:(022)23508339 23500755

营销部传真:(022)23508542

邮购部电话:(022)23502200

出版人 肖占鹏

承印 天津蓟县宏图印务有限公司印刷

经销 全国各地新华书店

版次 2004 年 3 月第 1 版

印次 2004 年 3 月第 1 次印刷

开本 787mm×1092mm 1/16

印张 9.75

字数 240 千字

印数 1—5000

定价 15.00 元

前 言

本书是高等职业教育教材《操作系统概论》一书的配套读物。编写的目的是帮助读者更好地掌握教材各章节的重点，理解一些难点内容，并且对自己学习的效果进行自查，以便做好考试的准备。全书分为三编。

第一编为课程基本要求的描述。首先概要叙述了课程的性质、特点、要求和对教材内容安排的理解。然后与教材内容相对应地按章节逐一介绍每一节应当掌握的知识点、对难点内容的辅导和考点所在。由于原教材中的第5章（网络操作系统简介）属于大纲以外扩展的内容，不在考试范围之内，所以，本书未做讲解。

第二编为实习指导。分别对应教材的第2、3、4章，给出了DOS、Windows、Linux三个操作系统的若干实习题目（包括实习内容、要求和指导）。目的在于使读者能够较好地掌握这三个有代表性的微机操作系统各自的特点，了解它们的功能、配置和使用方法。同时加深对操作系统原理的理解。

第三编为模拟试题，提供了各章节对应的典型试题分析与模拟试题。目的在于使读者能够举一反三、深入复习，并有助于自我测试、开拓应考准备的思路。

参加本书编写的有（按章节顺序）：韩劼（第一编）、王保旗（第二编和第三编第2章及附录的相应部分）、孙华志（第三编第1章及附录的相应部分）。编写者根据自己多年来在本课程教学实践中的体会，在本书的内容安排、叙述方法上力图把有关知识条理化，并希望能对读者在复习时熟悉考试方式、感受考试气氛有所帮助。当然，由于我们水平有限，加之时间仓促，本书难免会有缺点和疏漏，还望读者批评指正。

编 者

2003年8月

目 录

第一编 课程基本要求	1
第 1 章 操作系统的基本概念	1
1.1 操作系统的地位与作用	2
1.2 操作系统的形成、发展与分类	2
1.3 操作系统的功能与工作原理	6
1.4 操作系统的结构与工作机制	26
第 2 章 微机磁盘操作系统 DOS	29
2.1 DOS 操作系统的基本特点	29
2.2 DOS 的文件管理	30
2.3 DOS 的设备管理	33
2.4 DOS 的存储管理	34
第 3 章 微机的多任务操作系统 Windows	37
3.1 Windows 操作系统的基本特点	37
3.2 Windows 的进程管理	39
3.3 Windows 的存储管理	41
第 4 章 多用户分时操作系统 Linux	44
4.1 Linux 操作系统的基本特点	44
4.2 Linux 的基本应用入门	45
4.3 Linux 的文件管理	46
4.4 Linux 的进程管理	49
4.5 Linux 的存储管理	54
第二编 实习指导	56
第 1 章 DOS 实习指导	56
实习 1 CONFIG.SYS 和 AUTOEXEC.BAT 文件的建立	56
实习 2 DOS 的存储管理	57
实习 3 DOS 的设备管理	58
实习 4 DOS 的文件管理	60
实习 5 DOS 命令的综合练习	64
第 2 章 Windows 实习指导	68
实习 1 磁盘驱动器的维护	68
实习 2 系统维护	69
实习 3 MSDOS.SYS 文件内容的修改与设置	70
第 3 章 Linux 实习指导	72

实习 1 Linux 简单操作.....	72
实习 2 账户的创建与维护.....	72
实习 3 X-Windows 的使用.....	73
实习 4 Linux 文件管理.....	74
实习 5 进程管理与存储管理.....	74
第三编 模拟试题	76
第 1 章 操作系统的基本概念.....	76
1.1 操作系统基础与处理机管理.....	76
1.2 存储管理.....	89
1.3 设备管理与文件管理.....	97
第 2 章 常用微机操作系统.....	107
2.1 微机磁盘操作系统 DOS.....	107
2.2 微机的多任务操作系统 Windows.....	115
2.3 多用户分时操作系统 Linux.....	118
附录：模拟试题参考答案.....	129

第一编 课程基本要求

课程性质

操作系统是计算机系统软件的核心，操作系统的理论和常用微机操作系统的系统管理技术是高等职业技术教育计算机技术与应用专业学生必须掌握的一门重要的专业基础课。

课程特点

讲解操作系统的知识，要有一定的理论内容，不能只是讲解常用微机操作系统的用户界面使用方法；但是，理论内容的深度和广度要适当，不但要比大学本科专业课的内容浅显，而且要与微机常用操作系统的实际相结合。注重于理解（而不是深入研究）这些微机操作系统是怎样管理有关的系统资源，以及作为系统管理人员应当怎样运用相应的命令和工具软件去配置系统、了解系统工作状况、优化系统。通过这样的学习，使学生具有对微机进行系统管理的能力，能对系统性能做简单分析并做必要的调整、优化，而不仅仅是在某种操作系统环境中工作的一般应用型用户。

要求

根据本课程的特点和高等职业技术教育的要求，教材的内容做了这样的安排：第一章较系统地讲解了操作系统的基本概念，重点运用资源管理观点，讨论处理机管理、作业管理、存储管理、文件管理、设备管理的基本知识。在此基础上，第 2、3、4 章分别讲解 DOS、Windows 和 Linux 这三种操作系统的基本特点和工作原理。这三章还各有侧重，对于 DOS 系统，重点在于文件管理和内存管理；对 Windows 系统，重点在于进程管理和虚拟存储技术；对于 Linux 系统，则把重点放在多用户联机工作的特点及与此有关的共享、安全保护等问题上。通过这些内容去加深和扩展一般操作系统原理中的有关概念。同时这三章也包括了一些具体的系统管理操作技术。考虑到大多数读者过去对 UNIX、Linux 很少接触，所以在第 4 章中还包括了 Linux 的常用操作知识。第 5 章简单介绍了网络操作系统的基本知识。这一章是选修内容，不属于课程大纲基本要求范围，本书也未做讨论。在学习时一定要注意操作系统理论这个整体和 DOS、Windows、Linux 系统各部分局部功能的关系，提倡把有关内容前后串联、对比、逐步系统化，切忌把知识割裂开来、变成一个个散块。

第 1 章 操作系统的基本概念

学习目的与总要求

掌握操作系统的概念、作用与分类、操作系统的资源管理观点、操作系统的层次结构。理解有关进程、并发、资源共享等概念。

1.1 操作系统的地位与作用

知识点

1. 操作系统在计算机系统中的地位

这个问题可以从两个角度去理解。

从计算机系统组成的角度看，计算机系统由硬件系统和软件系统组成，其中软件系统由系统软件和应用软件组成。操作系统属于系统软件中的一种，而且是系统软件的核心。

从计算机系统层次结构的角度看，从里向外（或说从最底层开始直到使用者——用户）分别是：硬件、操作系统、系统实用软件、应用软件。操作系统是最靠近硬件的一个层次。

2. 操作系统的作用

可以分为三点：管理系统资源以提高系统资源利用率、提供软件的开发运行环境和提供用户操作界面。

难点

关于操作系统的定义。

一般来说，学习一个概念总是需要首先搞清楚它的定义。但是要想给出“操作系统”的准确定义是很困难的，许多关于操作系统的论著中对此都有不同的提法，没有一个完全适当、普遍适用的定义。对于操作系统包括哪些部分、不包括哪些部分，也没有全世界统一的规定。可以说，要想说清楚“操作系统是什么”，还不如说清楚“操作系统是做什么的”来得方便。这也就是上面讲到的操作系统的三点作用。通常也就据此给出操作系统的“非形式化的定义”：

操作系统是一个大型系统软件，由大量的程序模块和数据结构集合而成，它全面地统一控制和管理着计算机系统的所有硬、软件资源，合理地组织计算机的工作流程，以便高效率地利用这些资源为用户服务，使用户有一个功能强大且可扩展的工作环境，向用户提供方便友好的操作界面。

考点

1. 操作系统在计算机系统中的地位：操作系统是硬件还是软件，操作系统是系统软件还是应用软件，操作系统在计算机系统层次结构中处于哪一层、与哪些层相邻。

2. 操作系统的三点作用：操作系统管理的对象是什么、管理的目的是什么。操作系统在软件的开发运行中起什么作用、在计算机与用户之间起什么作用。

1.2 操作系统的形成、发展与分类

知识点

1. 操作系统的形成

手工操作阶段。主要矛盾是人机工作速度的差异太大。

批处理阶段（早期的批处理，或者与以后发展起来的多道批处理相对比地称之为单道批处理）。出现了“监控程序”（Monitor）来对成批作业进行管理。主要矛盾是计算机与外部设备工作速度的差异太大。

执行系统。在硬件条件（主要是通道和中断技术）的支持下实现了主机和输入输出设备

的并行工作。主要矛盾是主机资源利用率不高。

2. 操作系统的发展

多道程序系统。在计算机内存中同时存在着多个互相独立的程序，它们各自在宏观上都在运行着，但是由于只有 CPU（在单处理机系统中），所以在微观上它们还是在轮流使用 CPU 执行各自的指令序列，这称为多道程序的并发。并发的多道程序可以充分地利用主机和外设的并行操作能力，所以使系统资源利用率大大提高。

在原先的批处理系统中采用多道程序设计技术就产生了多道批处理系统。

通用操作系统。按照多道程序的基本思想，把多道批处理系统、分时系统、实时系统综合起来。

个人计算机、多处理器计算机和计算机网络的出现，使操作系统进一步发展。

3. 操作系统的分类

多道批处理系统：系统资源利用率高，作业吞吐量大，没有与用户的交互能力，作业周转时间长。

（多道）分时系统：与用户有交互能力，多个用户同时工作（多路性），每个用户的作业独立工作（独占性）。

实时系统：在限定的时间内予以响应（高及时性），高可靠性。

通用系统：以分时作业或实时作业为前台作业、以批处理作业为后台作业。

个人计算机操作系统：用户界面要求友好、简单，以实时多任务方式支持多媒体处理。

网络操作系统：共享资源、用户通信、网络管理、安全控制等。

分布式系统：直接对全系统的资源进行全局调度、控制，实现分布式计算处理。需要解决系统重构、负荷分配、容错等问题，要有很高的可靠性和对用户的透明性。

多处理机系统有与分布式操作系统相似的功能：统一调度、控制 CPU（及其配套的缓存、内存等等）。

难点

学习“操作系统的形成与发展”时需要注意，不能是单纯地学历史，而是要掌握现代操作系统的基本思想。

1. 多道程序设计的思想

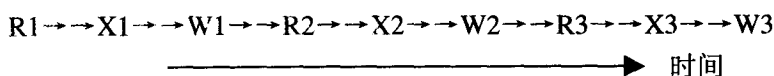
“多道程序”的思想可以说是操作系统理论的一个核心，由此才能真正理解操作系统最重要的概念：“进程”、“资源”和操作系统最基本的特征——“并发”、“共享”和“随机”。

冯·诺依曼结构的计算机主要的技术特点之一就是“程序的顺序存储与控制”。在一个处理机上多个程序的顺序执行具有“顺序性”、“封闭性”（一个程序执行的最终结果只取决于这个程序本身及其初始条件）和“可再现性”的特点。为了提高处理机的工作效率而引入多道程序的思想后，顺序程序的封闭性和可再现性不复存在，而出现了“并发性”、“共享性”和“随机性”。

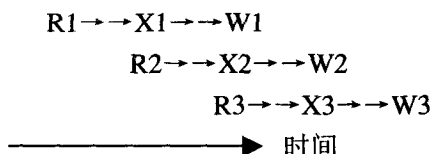
假设一个程序按照

输入数据（R）→计算处理（X）→输出结果（W）

这样几大部分的次序编制和运行，如果有三个这样的程序要顺序运行，每个程序在执行时都独占 CPU 和其他所有系统资源，执行结束后再把全部资源交给下一个程序。



而按照多道程序的思想, 不必等到第一个程序执行完了, 就可以开始第二个程序的运行。



这叫做多道程序的并发执行, 即通过计算机不同硬件部分并行工作, 使几个程序运行时间有重叠, 从宏观上看好像在同时工作。

(1) 并发性。计算机系统中同时有两个或两个以上程序存在, 它们都在同一个操作系统的管理下、处于已经开始运行而且都还没有结束运行。宏观上它们在同时运行, 但微观上, 同一个系统硬件 (例如处理机) 还是被几个程序轮流使用, 只不过处理机在执行完某个程序的某一条指令后, 并不一定就接着执行该程序的下一条指令, 而很可能转而去执行另一个程序的一条指令。如果使用系统的不同硬件, 则几个程序的不同类型的动作可能同时工作, 如上面示意的 W1、X2 和 R3。

(2) 共享性。即资源共享, 这时系统的各项资源都不再由一个程序独占, 而是由多个程序共同使用。不仅硬件资源 (处理机、内存、外设等) 被共享, 软件资源 (如编译、连接等实用软件、数据等等) 也是如此。有共享就有竞争, 例如处理机, 即使在多处理机系统中, 处理机的个数总是要少于多道程序的道数, 于是各个程序就要争相使用。

(3) 随机性。系统中运行的各个程序其行为和状态是随机的, 它们何时开始运行、何时做输入输出、何时由 CPU 执行其指令等等都无法在事先确切地知道, 因为它们在“并发”, 在竞争使用各种资源。系统中的硬件设备工作状态也有类似的情况。

多道程序的并发执行是现代操作系统最基本、最重要的技术。所以上述特点 (特别是并发性和共享性) 也就是现代操作系统的基本特征。并发性提高了系统资源的利用效率, 但另一方面, 资源的共享、竞争和状态的随机性也为多道程序设计带来了一系列必须加以解决的问题。

2. 操作系统运行的硬件环境

操作系统是最靠近硬件层的软件, 其工作是直接依赖硬件环境的。比如, 多道程序并发执行的前提条件, 是计算机系统的不同硬件部分能并行工作。与操作系统工作密切相关的硬件环境主要有:

(1) 通道。通道是一种硬件, 它相当于一个专用处理机, 与 CPU 可以并行工作、共享内存, 实现主机 CPU 和输入输出设备的并行工作。

(2) 中断技术。中断技术是各种硬件 (包括通道)、软件与主机 CPU 的一种通信手段。

CPU 为了接受和处理中断, 需要提供中断接口, 包括中断信号线的引脚和中断处理的有关指令。当有中断发生时, 处理过程是这样的:

各种“中断源”在需要时向主机 CPU 发出“中断请求” (硬件设备的中断信号或程序中的请求中断指令);

主机 CPU 在适当时机响应这个中断, 暂时中止自己当前正在执行的程序, 不再按照程序计数器 (PC) 的内容调入下一条指令;

把当前程序执行的现场信息（包括程序计数器和其他有关寄存器的内容等等）压入堆栈保存起来；

根据中断源找到相应的“中断处理程序”入口地址（各个中断处理程序的入口地址按照一定规律保存在内存的特定区域，称为“中断向量”），把该地址送入程序计数器；

启动该“中断处理程序”执行；

当中断处理程序执行结束后，再从堆栈中弹出当初保存的、被中止执行的那个程序的现场信息，恢复其现场并从断点继续执行，如图 1.1 所示。

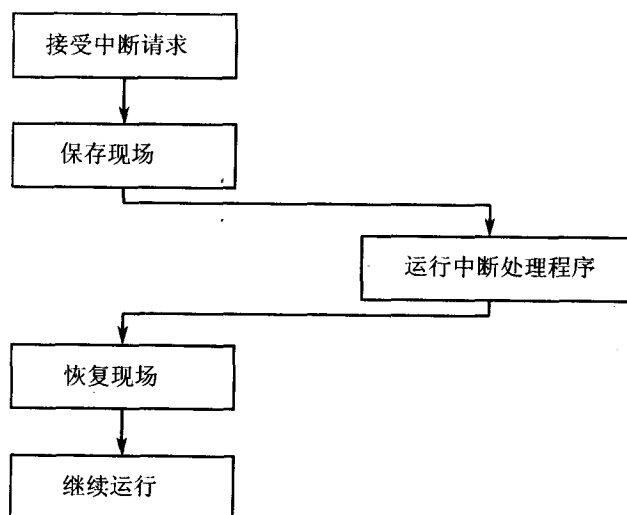


图 1.1 中断过程示意

（3）CPU 寄存器。程序运行过程中需要访问一些寄存器，特别是操作系统要使用许多专门的寄存器。从功能上看，寄存器可以分为两大类。一类是用户可编程的寄存器，如通用数据寄存器、各种地址指针寄存器等。另一类是状态和控制用寄存器，如程序计数器（PC，保存着下一个将被执行的指令的地址）、处理机状态字寄存器（PSW，它随时记录着 CPU 的工作状态，操作系统通过读取和修改 PSW 的内容来控制系统的工作）等。

（4）硬件保护机制。CPU 的指令分为“特权指令”和“非特权指令”两大类。特权指令就是那些只能由操作系统来使用的指令，比如，CPU 在中断响应过程中执行“保存现场”、“恢复现场”等等操作时所使用的指令就是只能由操作系统来使用而不能被用户使用的。相应地，处理机的工作状态（或称“工作模式”）也分为“特权状态”（又称管理态、核心态、管态等）和“普通状态”（又称用户态、目标态、目态等）两种。它们分别使用不同的堆栈，有不同的访问权限。在普通态下工作只能执行非特权指令，只能使用一部分寄存器，只能访问有限的内存空间；而在特权状态下工作，能执行包括特权指令在内的全部指令，能使用全部寄存器和内存空间。操作系统本身是一个包括许多程序模块的大型软件，其工作过程也就是这些程序模块的运行，事实上，操作系统的许多工作就是在特权状态下执行特权指令去完成各种中断处理和各种核心功能。

考点

1. 各种操作系统各有哪些主要特点？

2. 现代操作系统的基本思想是什么（多道程序设计的思想）？多道程序设计为什么能提高系统资源利用率？

3. 操作系统的基本特征是什么？程序并发执行的含义是什么？

4. 通道是硬件还是软件？通道的作用是什么？

5. 中断是一种什么技术？引起中断的原因有哪些（硬中断、软中断两类）？中断的处理过程有哪些步骤？为什么需要“保存现场”？

6. PC 寄存器和 PSW 寄存器的作用是什么？

7. 什么是特权指令？处理机在特权状态下能执行哪些指令？在普通状态下能执行哪些指令？

1.3 操作系统的功能与工作原理

由于这一节内容庞杂，下面分小节讨论。

1.3.1 研究操作系统的资源管理观点

知识点

1. 资源

操作系统的基本特征之一就是“共享”资源，包括计算机硬件资源和软件资源。资源总是有限的，而并发的多道程序要竞争使用它们，这就是操作系统要解决的基本矛盾。

对于各种资源从不同的角度上看，会分别具有不同的特点。这里最主要的就是“共享资源”和“独享（占）资源”的区分、“可抢占的资源”和“不可抢占的资源”的区分。

2. 资源管理观点

资源管理观点是从操作系统的功能角度进行分析的结果。按照这种观点，操作系统的功能可分为进程管理（或称为处理机管理）、作业管理、存储管理、设备管理、文件管理。

需要说明的是，中央处理机（CPU）是计算机硬件系统中最重要资源，操作系统对处理机的管理，可以进一步分为对进程的管理和对作业的管理。不过，作业及其管理的概念主要是从早期的多道批处理系统发展起来的，在目前的个人计算机或工作站系统中已经不大使用了。所以现在所说的处理机管理基本上就是进程管理。

3. 资源管理方法

主要就是对资源的分配、回收、协调程序之间的竞争与合作，以及相应地对分配给不同程序使用的资源施加保护、对“不够使用的”资源给予“虚拟地”扩充。显然，对不同特点的资源，管理的方法和重点也不同。

4. 资源管理中的死锁

死锁是发生在—组并发工作着的程序之间的现象，这些程序各自已经占有着一些不可抢占的独占资源，而又申请已被组内其他程序占有的这种资源，于是处于一种循环等待的僵持状态。

通常把这种不可抢占的、独享资源称为临界资源。硬件资源可以是临界资源（比如某个打印机、某个堆栈），软件资源也可以是临界资源（比如外存上的某个数据）。

死锁发生的条件与解决方法。

难点

主要是死锁问题。在一般教材中死锁问题是放在“进程管理”部分讲授的，本课程为了分散难点把它提前到了这里。

1. 死锁发生的条件

(1) 资源互斥：并发工作着的程序已经占有和申请的资源是独享的；

(2) 资源不可抢占：这些程序已经占有的资源只能由占用者自己把它释放，其他程序才能申请到；

(3) 资源部分分配：程序在已占有有一些资源且不释放的时候又申请别的资源；

(4) 循环等待：两个或两个以上并发程序对资源形成“循环等待”。

由于不可抢占的、独享资源又可以称为“临界资源”，所以上述(1)(2)两个条件也可以叙述成一个条件：所涉及的资源是“临界资源”。

2. 解决死锁问题的方法

(1) 在系统运行之前的“预防死锁”

就是说在设计操作系统时确定比较合理的处理原则和算法，破坏上述条件中的某一条，使死锁在系统运行的整个过程中都不可能发生。如：

采用“假脱机”技术——破坏“资源互斥”和“资源不可抢占”。

采用资源静态分配法——破坏“资源部分分配”。

采用资源有序分配法——破坏“循环等待”。

(2) 在系统运行之中的“避免死锁”

在操作系统工作过程中，进行动态判断。每当一个程序申请某一资源时，操作系统按照一定算法预测一下是否会进入某种有死锁危险的状态，若有危险则拒绝该申请。“银行家算法”就是避免死锁的一个著名算法，其基本原理是：申请资源的进程必须把自己对资源的最大需求量报告系统，而系统则遵循这样的原则：如果系统现有的剩余资源能够满足一个进程此后的最大需求，就满足该进程的申请，分配给它。这样总能保证有一个进程能得到它所需的所有资源并在有限的时间内执行完毕，然后释放它占有的全部资源，还给系统。系统能拿这些资源按照同样的原则再去满足其他进程。

(3) 在死锁发生后的“诊断与解除”

“诊断”(或称“检测”)即判断系统中当前是否已经发生了死锁。

如果已有死锁发生，常用的解除(或称“恢复”)措施有：

资源剥夺法。按一定顺序强行剥夺某些进程的资源并重新分配给已死锁的进程。

撤销进程法。按一定顺序强行撤销已死锁的那些进程并收回其占有的资源。

采用这两种方法时，首先要选择那些被剥夺或撤销而造成损失最小的进程。而且在每次收回资源后应立即再分配并再检测是否还有死锁。

由于各种死锁预防或避免算法的实现不但会严重地降低系统资源利用率，而且其本身又大大地增加了系统的开销，所以事实上很少采用。即使是“诊断”死锁，也要增加系统开销，加重系统负担，所以也只适合于系统中死锁发生率不高的情况。

考点

1. 什么是“共享资源”？什么是“独享(占)资源”？什么是“可抢占的资源”？什么是“不可抢占的资源”？

2. 操作系统的功能是什么（五大管理）？
3. 什么是“临界资源”？
4. 什么是“死锁”？死锁发生的条件有哪些？解决死锁问题的方法有哪些？

1.3.2 处理机管理

知识点

1. 进程的定义

一个进程，就是一个功能独立的程序针对某个数据集合（处理对象）的一次执行过程。

2. 进程的特征

(1) 动态性。进程是程序执行的动态过程，每个进程都有其生命期，动态地产生和消亡。进程在其生命期内，也是处于几种不同的状态中，动态地在这些状态之间转换，所占有的资源也是动态变化的。

(2) 独立性。进程是从系统获取资源的独立单位，并且独立地执行。

(3) 并发性和异步性。进程是并发执行的单位，系统中同一时刻会有多个进程同时存在，宏观上它们在同时工作，微观上它们交替地使用 CPU，各按各的具体情况、按照不可预知的速度或快或慢地“共同前进”。

(4) 相互制约性。虽然各个进程是相互独立的，但是它们既然在并发工作、竞争使用资源，其独立性就必然不可避免地要受到一定的制约，并且可以采用一定的通信手段互相协同。

3. 进程与程序的区别

(1) 程序是个静态的事物，是若干指令的有序集合；而进程是个动态的概念，是一条条指令按顺序执行的过程。

(2) 程序存放在一定的存储介质上，一般（只要不删除）是长期存在的；而进程不但占用存储空间还要占用 CPU，只有一个很短的存在时间（生命期）。

(3) 同一个程序可以被多个进程同时执行，而一个进程也可以包括执行多个程序。

4. 进程的组成（或进程的描述）

(1) 进程的地址空间。即进程所占用的存储空间，其中存放有进程所执行的程序——指令代码集合、进程所使用的数据集合、进程所使用的堆栈等。

(2) 处理机的现场场景。即各有关寄存器的内容。

(3) 进程控制块（PCB）。这是一个数据结构，是一个进程存在的基本标志。它包括：
进程描述信息，如进程名、进程标识码 PID、进程的主人和父进程等；
进程控制信息，如当前所处的状态与队列指针、当前的优先级、通信信息等；
进程资源信息，如所占用的内存空间、外设、缓冲区、文件等（相应的各种指针）；
处理机现场保护信息。

5. 进程的状态

进程在其生命期中始终处于不断变化的过程，可以分解为处于若干种状态。在一个系统中，进程至少有三种基本状态。

(1)（当前）执行状态（又称运行状态）。

(2) 等待状态（又称封锁状态、阻塞状态）。

(3) 就绪状态。

这三种状态之间可能存在的转换路径有：执行→等待，执行→就绪，等待→就绪，就绪→执行；不可能的转换路径则有：等待→执行，就绪→等待。

进程状态发生变化的原因，有进程本身执行造成的，也有外界条件造成的。

进程状态转换的过程，由操作系统对进程的控制与调度来实现。

6. 进程控制

包括创建进程、撤销进程、阻塞进程、唤醒进程等操作。

7. 进程调度

进程调度的任务是在处于就绪状态的进程中选择合适的进程送入执行状态。

进程调度的功能是按照调度策略选择“新”进程并进行“新”、“老”进程的场景切换。

进程调度的策略（算法）主要有：先进先出（FIFO）、最高优先级（HPF）、按时间片轮转（RR）等。

进程调度的时机取决于操作系统的类型和所使用的调度算法，通常可以有：

(1) 正在执行的进程结束，即其程序运行完成；

(2) 正在执行的进程由于多种自身原因转为等待状态（比如进程申请 I/O 操作、进程申请某个资源而尚未得到满足等等）；

(3) 正在执行的进程做完了一个时间片；

(4) 在可抢占 CPU 的系统中，就绪状态中出现了一个优先级比正在执行的进程还高的进程。

8. 进程间的制约关系与进程通信

并发进程之间存在两种制约关系：

间接制约——由于进程共享同一个公有资源而产生对各进程执行速度的影响。如果需要共享的资源是一种“临界资源”（即独享的、不可抢占的资源），操作系统必须采取“互斥”的管理手段来加以协调。

直接制约——进程之间由于其程序功能本身的需要而在设计时必须加以考虑互相协调、配合，必须按一定顺序协调一致地执行特定操作的互相制约关系。这样的进程称为“相关进程”。相关进程彼此之间必须采取“同步”措施。

进程间利用通信可以实现“互斥”、“同步”和其他信息交换任务。

“互斥”、“同步”与进程通信是进程管理的重要内容，但是本课程为了分散难点而把对这些内容的进一步讨论安排在第4章（即 Linux 系统）中。如果在教学中不包括第4章，也可以考虑把这一部分内容提到本章来。

难点

1. 进程的动态性

进程是程序的执行过程，在理解进程的各方面概念时（进而在理解整个操作系统的工作时）都必须特别注意进程的“动态性”这个根本特性。比如，在理解进程的描述时，诚如上面所讲，进程由进程的地址空间、处理机的现场场景和进程控制块组成，但千万不要把它们理解为静止的、一成不变的、像纸面上所写的程序那样。在 1.3.4 节将会讲到，进程使用的内存空间在绝大多数情况下是由系统动态分配的，某一条指令、某一个数据在内存中的具体位置在进程执行的过程中是经常变化的。处理机的现场场景更是这样，随着每条指令的执行，

各有关寄存器的内容都会有所变化。进程控制块虽然不像地址空间、现场场景那样抽象，是个很具体的数据结构（就像是操作系统为每个进程建立和填写的一张表格），但它的许多内容也是随时变化的。可以说，一个进程的动态状况基本上都体现在它的 PCB 的内容里。进程当前处于哪种状态要记载在 PCB 中；如果处于就绪状态，系统会通过 PCB 中相应的队列指针把该进程组织在某个队列中以便按照一定顺序得到下一次执行的机会，而随着系统一次次的调度，各就绪队列的情况也在变化，PCB 中记载的队列指针也就变化。处理机现场保护信息就是进程退出执行状态那一瞬间的被保存起来的硬件现场场景信息。如此等等，读者在以后学习存储管理、设备管理、文件管理等时都要注意把相应内容与这里讲到的进程、进程控制块联系起来，体会它们是怎样动态变化的。

2. 进程控制块的作用

进程控制块 (PCB) 是一个进程存在的基本标志。从概念上来说，系统管理的对象是进程，从实际操作上来说，系统管理的对象就是进程的 PCB。在一个实际的系统中，会同时存在着若干个 PCB（包括若干系统进程的 PCB 和若干用户进程的 PCB）。有些系统对于最大进程数（或最大的用户进程数）有限制，指的就是最多能同时建立多少个 PCB。

为了进行管理，操作系统把这些 PCB 以若干个线性表（顺序表或链表）的形式组织起来。操作系统对进程进行各种控制、调度，都离不开对 PCB 的读写。建立一个进程就是申请一个空闲的 PCB 空间，将该进程的各种初始信息填入这个 PCB，并把这个 PCB 插入相应的就绪队列中去，使该进程置为“就绪”状态。撤销一个进程的实质就是撤销其 PCB。进行进程调度时，系统要读取各有关进程的 PCB，以了解它们的状态、优先级、所属队列及在队列中的位置等等，按照调度算法选出下一个被送去执行的进程。“新”、“老”进程的场景切换操作也就是用专门的指令把原执行进程的现场场景写入其 PCB、用专门的指令把新进程 PCB 里的现场保护信息写入 CPU。

其他诸如存储管理、设备管理、文件管理等也都是面向进程展开的，也都离不开对 PCB 的读写。

3. 谁来做进程调度

进程的调度工作由操作系统中的进程调度程序完成，可以说这是操作系统中最核心的部分。在实际工作中，这个程序几乎每秒钟就要运行若干次，它是作为一个软中断的处理程序常驻内存的。

如前所述，中断可以由硬件（通过发出中断信号）引起，也可以由软件（通过程序中的相应指令）引起。每当到了前面讨论过的进程调度时机（这种时机的性质对具体的操作系统来说是确定的、可预期的），系统就会进入 CPU 特权状态（核心态）工作，执行一个软中断指令转入进程调度程序的工作。其实，操作系统中其他许多管理任务的实现，如下面几节将讲到的交换、（请求页式存储管理的）调页、打开文件、读写文件、各种 I/O 操作等等，也都类似于此。所以才有“操作系统（特别是它的核心功能）的工作建立在中断驱动基础上”的说法。

考点

1. 什么是进程？进程有哪些特征？进程的动态性指的是什么？进程与程序有什么联系与区别？

2. 什么是进程控制块？它的英文缩写是什么？它包括哪些内容？它与进程组成的其他部

分有什么关系？操作系统怎样通过进程控制块来管理进程？

3. 进程有哪三种基本状态？各种状态之间的转换路径中哪些是可能的，哪些是不可能的？转换的原因是什么？

4. 进程调度的任务是什么？常用调度算法有哪些？什么时候进行调度？什么是进程调度时的“场景切换”？

5. 什么是进程之间的间接制约？什么是进程之间的直接制约？什么是进程之间的互斥？什么是进程之间的同步？

1.3.3 作业管理

知识点

1. 作业的定义

作业是用户为了完成一项完整的计算或事务处理任务而要计算机所做的全部工作。

2. 作业与进程的关系

(1) 作业是用户提交的任务实体，而进程是完成该任务的执行实体；

(2) 一个作业可以包括多个（至少一个）进程，即一个根进程和若干个子进程，因为为了完成一个作业，显然需要执行至少一个、而且往往是多个程序。

3. 作业的类型

作业一般可分为两大类，即批量型作业和终端型作业。

4. 批量型作业的有关问题

(1) 批量型作业有完整的作业概念。较大型的作业往往要顺序经过几个步骤（称为作业步）才能完成，简单的作业也可能只包括一个作业步。一个作业，可以认为包括程序、数据、作业说明书这样三个部分。

(2) 批量型作业从被建立到完成撤销，在其存在过程中历经四种状态：提交、后备、运行、完成。

(3) 批量型作业的控制方式是脱机控制方式。其具体手段是由用户把对作业的全部控制意图，用系统提供的作业控制语言（JCL）或专门的命令程序设计语言，事先编写成作业说明书，连同组成作业的所有程序和数据等一起输入系统。作业说明书一般应包括：作业基本情况的描述、作业控制的描述和作业对资源需求的描述。

(4) 作业控制块（JCB）。系统在建立一个作业时要建立它相应的作业控制块。JCB 一般应包括：作业基本信息、作业对资源的要求及当前对资源的实际占有情况、作业当前状态等。作业结束时系统就撤销它的 JCB。

(5) 作业调度。作业调度的时机往往是当有新的作业建立或是内存中有作业已经完成的时候。作业调度的主要任务是：掌握已经进入系统的每个作业的情况，根据一定的作业调度算法从处于后备状态的作业中挑选一个或几个作业，并为其分配内存和外设资源、建立进程，使其投入运行，作业结束后撤销作业。调度算法主要有：先来先服务算法（FIFS）、最短作业优先算法（SJF）、最高响应比作业优先算法、优先级算法、均衡调度算法等。

(6) 批量型作业开始存在于早期的多道批处理系统中（其体现就是“作业说明书”）。在后来的执行终端型作业、实时处理中则可以认为不存在作业的概念。

5. 终端型作业的有关问题