



普通高等教育“十一五”国家级规划教材

高等学校规划教材

操作系统教程

(第4版)

◎ 陆松年 主编

◎ 翁 亮 潘 理 薛 质 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校规划教材

普通高等教育“十一五”国家级规划教材

操作系统教程

(第4版)

陆松年 主编

翁亮 潘理 薛质 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书为普通高等教育“十一五”国家级规划教材。本书以操作系统的基本原理、概念和应用为框架,以传统的 UNIX 为主线,分析了最近推出的 Linux 3.10.12 版本的存储管理、进程管理、进程通信、设备管理和文件系统的基本结构、算法、重要的数据结构和流程,详细给出了典型模块的源代码和注释,还对 Windows 7 和操作系统安全进行了分析。全书分 3 篇:第 1 篇介绍操作系统的基本原理及较常用的操作系统实例,如 UNIX、Linux、Windows 7/8 和 Windows Server 2012 等;第 2 篇介绍在 UNIX 和 Linux 系统下的系统程序设计、应用开发工具及开发技术;第 3 篇介绍 UNIX 系统管理和网络管理。

本书可作为高等学校计算机科学与技术 and 计算机应用专业以及通信工程、电子工程、信息安全、自动化控制和信息管理类等非计算机专业的教材和教学参考书,对于计算机软件开发人员、系统和网络管理人员,也是一本很好的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

操作系统教程 / 陆松年主编. —4 版. —北京:电子工业出版社, 2014.4

ISBN 978-7-121-22671-7

I. ① 操… II. ① 陆… III. ① 操作系统—高等学校—教材 IV. ① TP316

中国版本图书馆 CIP 数据核字 (2014) 第 052674 号

策划编辑:章海涛

责任编辑:章海涛

特约编辑:何 雄

印 刷:北京丰源印刷厂

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:26.75 字数:800 千字

印 次:2014 年 4 月第 1 次印刷

定 价:49.50 元



凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

第 4 版前言

相对 Linux 来说, UNIX 操作系统比较简洁, 源代码容易读, 较适合于教学。本书出版已经 14 年了, 期间 Linux 和 Windows 操作系统都有了较大的发展, 这次修订的第 4 版以操作系统的基本原理、概念和应用为框架, 以传统的 UNIX 为主线, 并反映了 Linux 和 Windows 操作系统的最新发展。

与第 3 版比较, 第 4 版的 Linux 操作系统部分有了很大的修改。第 4 版采用最近推出的 Linux 3.10.12 版本, 分析了存储管理、进程管理、进程通信、设备管理和文件系统的基本结构、算法、重要的数据结构和流程, 详细给出了典型模块的源代码和注释, 以供教师、学生和研究 Linux 内核的读者参考。

本书分析的部分 Linux 源代码很长, 为了节省篇幅, 非关键的部分有所省略; 有些结构说明中的常数也被截短了, 余下部分用符号 “*” 表示。

针对目前使用人数最多的 Windows 操作系统, 本书紧跟 Windows 操作系统的最新进展, 舍弃了 Windows 2000/XP/2003 的结构分析等内容。第 7 章中以 Windows 7 操作系统为主线, 全面分析了系统体系结构、存储管理、进程、线程及处理器管理、I/O 系统、文件系统和网络系统。对微软最新的 Windows 8 和服务器操作系统产品 Windows Server 2012, 重点分析了其对以前 Windows 版本的功能更新部分。第 7 章还对 x86 平台操作系统虚拟化技术的基本原理进行了介绍, 并以开源的 Xen 虚拟化技术为例说明了虚拟化的基本原理, 重点分析了 Hyper-V 的系统架构和处理过程。此外还介绍了可运行云服务器、数据中心、Web 和应用程序的云计算操作系统 Windows Azure。

随着全球计算机信息化的发展, 用户对信息服务的安全性提出了越来越高的要求。计算机信息系统的安全性主要依赖于其软件系统的安全性。由于操作系统是计算机软件系统中具有基础性、战略性和唯一控制硬件的关键软件, 其安全职能是其他软件安全的根基。没有操作系统的安全就不可能真正解决计算机系统安全, 进而解决整个网络信息系统的安全性问题。只有增强了操作系统的安全功能, 各种病毒、木马程序、网络入侵和人为非法操作才能被真正抵制。

Solaris 曾是使用最广泛、最成功的商业 UNIX 实现版本, 也是一个最安全和最稳定的商用操作系统。但自 2005 年 Sun 公司正式推出其开源项目 OpenSolaris 以来, 并没有取得关键的发展。本书第 3 版曾分析了 Solaris 操作系统, 但基于篇幅原因, 第 4 版删除了有关内容, 感兴趣的读者可参见第 3 版。

本书内容深浅适度, 安排系统、合理, 不仅介绍了操作系统的最新原理, 实用性也很强。本书可作为高等学校计算机科学与技术 and 计算机应用专业以及通信工程、信息安全、电子工程、自动化控制和信息管理类等非计算机专业的教材及教学参考书, 对于计算机软件开发人员、系统和网络管理人员, 也是一本很好的参考书。

全书分为原理篇、应用和开发篇、系统和网络管理篇 3 部分, 共 16 章。本书由陆松年、翁亮、潘理、薛质编写, 由陆松年统稿。

本书计划讲课学时为 80 学时，不同专业可根据需要删去或略讲书中某些章节，将学时压缩至 50~70。值得指出的是，操作系统是一门实践性很强的课程，因此不论在校学生或自学者都要进行一定数量的上机实验。与本书配套的还有由陆松年主编的《操作系统实验教程》（电子工业出版社，2010）。

本书在编写过程中得到了上海交通大学电子信息与电气工程学院领导对本书的大力支持，不少学生也参加了 Linux 分析工作，在此一并表示感谢。

由于作者水平有限，教学内容尚需要不断更新，特别是对 Linux 3.10.12 版本的分析，国内外鲜见论著发表，几乎完全是直接对源代码进行研究和分析，书中难免存在一些错误，恳切希望各位学者和读者批评指教，作者将不胜感谢。

本书为读者提供相关教学资料，可从华信教育资源网站（<http://www.hxedu.com.cn>）下载。

作 者

目 录

第 1 部分 原理篇

第 1 章 操作系统概论	3	3.7.2 线程的状态与功能	64
1.1 什么是操作系统	3	3.7.3 用户级和核心级线程	64
1.2 操作系统的发展历史	4	3.7.4 线程应用示例	66
1.3 现代操作系统类型	7	3.8 Linux 进程管理	66
1.4 操作系统的概念、特征和功能	11	3.8.1 Linux 进程控制块	67
1.5 UNIX 操作系统概述	14	3.8.2 进程调度	70
习题 1	16	3.8.3 进程创建	74
第 2 章 存储管理	18	3.8.4 Linux 的线程	82
2.1 存储管理基础	18	习题 3	83
2.2 基本存储管理方法	20	第 4 章 进程通信	84
2.3 可变分区存储管理	21	4.1 进程的同步与互斥	84
2.4 内存扩充技术	25	4.2 进程间互斥控制方法	85
2.5 纯分页的存储管理	27	4.3 信号量和 semWait、semSignal 操作	88
2.6 请求分页系统	30	4.4 信号量的应用	89
2.7 段式存储管理	33	4.5 进程间的数据通信	93
2.8 段页式存储管理	34	4.6 软中断和信号机构	95
2.9 Linux 存储管理	35	4.7 死锁	97
习题 2	43	4.8 Linux 的 IPC 机制	102
第 3 章 进程管理	44	4.8.1 Linux IPC 概述	102
3.1 进程概述	44	4.8.2 消息队列	104
3.2 进程控制块	46	4.8.3 共享内存	110
3.3 调度	50	4.8.4 信号量	113
3.3.1 调度概述	50	习题 4	118
3.3.2 进程调度策略	50	第 5 章 设备管理	120
3.3.3 进程调度算法	51	5.1 概述	120
3.4 UNIX 系统的进程调度	54	5.2 操作系统与中断处理	122
3.5 进程的控制	57	5.3 操作系统与时钟系统	126
3.6 进程的创建和映像改换	60	5.4 操作系统对 I/O 操作的控制	127
3.7 线程	62	5.5 设备管理的数据结构	129
3.7.1 多线程	62		

5.6 磁盘调度	130	5.9 Linux 的设备管理	143
5.7 UNIX 系统 V 的设备管理	134	5.9.1 概述	143
5.7.1 块设备管理的主要数据结构	134	5.9.2 Linux 字符设备管理	144
5.7.2 缓冲区管理	136	5.9.3 Linux 块设备管理	148
5.7.3 块设备管理	139	5.9.4 Linux 块设备缓冲区的管理	151
5.7.4 字符设备管理	140	5.9.5 网络设备的管理	154
5.8 设备分配	142	习题 5	154
第 6 章 文件系统	156	习题 6	214
6.1 概述	156	第 7 章 Windows 操作系统	216
6.2 文件目录	160	7.1 Windows 操作系统发展历程	216
6.3 文件存储资源分配	162	7.2 Windows 7	219
6.4 文件的系统调用	164	7.2.1 系统体系结构	219
6.4.1 文件的创建、打开、关闭和 取消	164	7.2.2 存储系统	222
6.4.2 文件的读写	166	7.2.3 进程、线程及处理器管理	228
6.4.3 调整文件读写位置 lseek	166	7.2.4 I/O 系统	232
6.4.4 创建任何类型文件 mknod	167	7.2.5 文件系统	235
6.4.5 其他文件系统调用	167	7.2.6 网络系统	238
6.5 文件的标准子例程	168	7.3 Windows Server 2012	246
6.6 UNIX 文件系统的内部结构	173	7.4 Windows 8	247
6.6.1 索引节点	173	7.5 操作系统虚拟化及云计算	249
6.6.2 文件索引结构	173	习题 7	257
6.6.3 目录结构	174	第 8 章 操作系统安全	259
6.6.4 打开文件结构	176	8.1 操作系统的安全性	259
6.6.5 文件系统存储资源管理	178	8.1.1 操作系统的安全需求	259
6.7 管道文件和管道通信	180	8.1.2 系统安全的评估与标准	260
6.8 Linux 虚拟文件系统	184	8.1.3 Linux 操作系统安全	262
6.8.1 虚拟文件系统	184	8.1.4 Windows 2000 操作系统安全	263
6.8.2 VFS 主要数据结构	188	8.2 操作系统安全机制	265
6.8.3 VFS 文件的打开	193	8.2.1 标识与鉴别	265
6.8.4 VFS 文件的读写	197	8.2.2 可信路径	266
6.9 Ext 文件系统	199	8.2.3 最小特权管理	267
6.9.1 Ext2 文件系统	199	8.2.4 访问控制	268
6.9.2 Ext2 的主要数据结构	202	8.2.5 隐蔽通道检测与控制	270
6.9.3 Ext2 文件的操作	207	8.2.6 安全审计	271
6.9.4 Ext3 文件系统	212	8.3 安全操作系统设计与实现	273
6.9.5 Ext4 文件系统	213	8.3.1 操作系统安全设计原理	273
		8.3.2 安全策略	273

8.3.3 安全模型	274	8.3.5 典型安全操作系统	278
8.3.4 安全体系结构	276	习题 8	280

第 2 部分 应用和开发篇

第 9 章 UNIX 使用基础	283	11.7 其他软件开发工具	337
9.1 登录和退出系统	283	习题 11	338
9.2 Shell 基础	284	第 12 章 UNIX 网络通信	339
9.3 文件系统基本知识	287	12.1 检查计算机的连接情况	339
9.4 UNIX 文件系统结构	289	12.2 获取远程用户信息	339
习题 9	290	12.3 用户间通话	340
第 10 章 UNIX 实用程序	291	12.4 远程登录和执行	341
10.1 目录操作命令	291	12.5 远程文件复制和传输	342
10.2 文件操作命令	292	12.6 网络服务命令	344
10.3 显示和打印命令	297	习题 12	347
10.4 过滤器	299	第 13 章 UNIX 系统和网络程序设计	348
10.5 设备操作	305	13.1 文件系统程序设计	348
10.6 用户信息与进程控制	307	13.2 用文件的系统调用实现 进程通信	351
10.7 压缩和解压缩	309	13.3 高级进程间通信	354
10.8 其他命令	310	13.4 远程进程间通信 Socket	360
习题 10	311	13.5 UDP 套接字编程	364
第 11 章 软件开发工具	313	习题 13	367
11.1 文本编辑器 vi	313	第 14 章 Shell 程序设计	368
11.1.1 屏幕编辑命令	314	14.1 Shell 程序和参数	368
11.1.2 底行命令	317	14.2 Shell 变量	369
11.1.3 在 vi 程序中执行 Shell 命令	319	14.2.1 用户定义变量	369
11.2 C 编译系统	320	14.2.2 系统定义变量	370
11.3 库的维护	324	14.2.3 Shell 定义变量	371
11.4 维护程序的程序 (Make)	326	14.2.4 参数替换	372
11.4.1 Make 使用初步	327	14.2.5 引号机制	372
11.4.2 Make 的内部规则	328	14.3 测试和求值	373
11.4.3 Make 中的宏	329	14.4 控制结构	374
11.4.4 用 Make 维护库	330	14.4.1 顺序控制结构	374
11.4.5 一个综合例子	330	14.4.2 if 语句	375
11.4.6 Make 命令选项和参数	332	14.4.3 case 语句	377
11.5 调试程序	333	14.4.4 for 语句	377
11.6 源代码控制系统 (SCCS)	334		

14.4.5 while 和 until 语句.....	378	14.6 Shell 内部命令	380
14.4.6 break、continue、exit 和 return 语句.....	379	14.7 Shell 环境	382
14.5 递归和 Shell 函数	379	14.8 Linux 的 BASH Shell	383
		习题 14.....	385

第 3 部分 系统和网络管理篇

第 15 章 UNIX 系统管理	389	第 16 章 UNIX 网络管理.....	400
15.1 系统的启动和关闭	389	16.1 配置 TCP/IP	400
15.2 用户管理	392	16.2 网络服务监控进程.....	402
15.3 刷新系统缓冲区	394	16.3 域名服务系统.....	403
15.4 检查和修复文件系统.....	394	16.4 网络文件系统 NFS	407
15.5 构造、安装和拆卸文件系统.....	396	16.5 电子邮政系统.....	409
15.6 定时运行程序	398	16.6 WWW 服务器的配置.....	412
习题 15.....	399	习题 16.....	417
		参考文献	418

第 1 部分 原理篇

CPU 是计算机系统的心脏，操作系统是计算机系统的大脑。半个世纪以来，操作系统吸引了世界上一大群最热情、最有智慧的杰出人才，集中了人类现代创造性思维活动的精髓。操作系统是软件世界的万花筒、博览会，是软件王国中的一顶璀璨的皇冠。能够有机会走进操作系统这个神奇、复杂而又充满诱惑的大千世界，理解其中的基本概念、结构和奥妙的管理机制是十分幸运的。

本篇内容包括：

- 操作系统概论
- 存储管理
- 进程管理
- 进程通信
- 设备管理
- 文件系统
- Windows 操作系统
- 操作系统安全

第 1 章 操作系统概论

计算机系统由硬件和软件两部分组成。硬件是指人所看得见、摸得着的各种计算机部件，包括存储器、处理器、输入/输出设备及电源、机箱等。软件是指存在于计算机系统中或外存储器中的程序及数据的集合。仅有硬件而没有配备软件的计算机如同一堆废物，只有在配备了软件后，计算机才能“活”起来，为用户解决各种各样的控制、处理和计算问题。

计算机软件分为系统软件和应用软件。系统软件是计算机厂商为了便于用户使用计算机而驻存在计算机硬件（如硬盘和软盘）内的系统支持程序，一般是连同计算机硬件一起出售的。主要的系统软件有操作系统、编译和解释程序、汇编程序、连接装入程序、编辑程序和设备驱动程序等。应用软件是用户为了专门的应用目的，向计算机厂商或其他软件商购买的或自己开发的解决某一类问题的软件，典型的有数据库管理软件、图像处理软件及各种 CAD、CAI、办公软件和杀毒软件等。

在所有的系统软件中，操作系统是一种首要的、最基本的、最重要的系统，也是最庞大、最复杂的系统软件。几十年来，软件界花费了大量的时间和金钱来研究、开发、扩展和完善计算机操作系统，使其获得了飞速的发展。

1.1 什么是操作系统

尽管“操作系统”（Operating Systems, OS）这个名称诞生至今已有了几十年的时间，计算机使用人员一般都知道它，但要对其下一个精确的定义并非轻而易举。很多论述操作系统的书籍从不同角度对操作系统下了不同的定义，综合起来，通常把操作系统定义为用以控制和管理计算机系统资源，方便用户使用的程序和数据结构的集合。

（1）系统观点——计算机资源管理

在计算机系统中，CPU 是计算机硬件的核心，是计算机系统的心脏；操作系统则是计算机软件的核心，是计算机系统的大脑，是整个系统的控制中心，是计算机或智能控制管理系统中首要的、最重要的、最复杂的系统软件。

计算机系统的主要硬件有 CPU、控制器、内存及磁盘、闪存、键盘、鼠标、显示器、打印机、USB 和网络接口等形形色色的外部设备。在计算机系统内运行的各个程序为了各自的“利益”，无时无刻地争夺这些有限而较为昂贵的资源。操作系统的作用类似于城市交通的决策、指挥、控制和调度中心，它组织和管理整个计算机系统的硬件和软件资源，在用户和程序之间分配系统资源，使之协调一致、高效地完成各种复杂的任务。

现代计算机硬件设备种类越来越多，功能越来越强，控制和操作起来也越来越复杂。如果一个程序员要直接与打印机、磁盘等 I/O 设备打交道，那么就要对每一种设备编制几千、几万条机器指令，这不仅是用户力所不及的，对系统存储的信息来说，也是极其不安全的。操作系统向用户提供了高级且调用简单的服务，掩盖了绝大部分硬件设备复杂的特性和差异，使用户可以免除大量的令人乏味的杂务，而把精力集中在自己所要处理的任务上。

（2）用户观点——用户使用计算机的界面

如果只有裸机，用户几乎不可能使用计算机。为了能够使用户方便、灵活、安全、可靠地使用计算机，就要改造它，充分发挥其潜能。如果在裸机之上覆盖一层 I/O 设备管理软件，就能使用户较方便地使用外部设备；如果在其上再覆盖一层文件管理软件，用户就很容易存取系统文件

和用户文件；每覆盖一层新的软件，就构造了一台功能更强的虚拟计算机。通过操作系统，计算机能提供种类更多、质量更高的服务。同样一台计算机硬件，安装 MS-DOS 操作系统或安装 Windows 2003 操作系统，呈现在用户面前的是完全不同的两类“虚拟”计算机，因此操作系统是用户与计算机硬件之间的接口。用户一般可用以下 3 种方式获得计算机系统提供的服务。

① 命令方式。用户可通过键盘输入有关命令来直接使用计算机，以获得计算机所能提供的服务，也可将若干个命令集中存于一个文件中，以批处理的方式连续运行命令，获得多种服务。

② 系统调用。用户可在应用程序中调用操作系统向用户提供的服务程序，以获得系统服务。

③ 图形界面。用户可以在窗口环境中通过鼠标、按键、菜单和对话框等方式操纵计算机，这是现代操作系统向多媒体转化的标志之一。

(3) 软件观点——程序和数据结构的集合

操作系统是直接和硬件相邻的第一层软件，是由大量极其复杂的系统程序和众多的数据结构集成的。在计算机中的所有软件中，操作系统起到了核心和控制的作用，其他软件的运行都要依赖它的支持。操作系统是在系统中永久运行的超级程序。

1.2 操作系统的发展历史

1. 早期的计算机和人工操作方式

在 1945 年，世界上第一台通用的数字计算机 ENIAC 诞生以后的早期真空管电子计算机时代，连汇编语言这样简单而基本的软件也没有，更不用说操作系统了。那时的计算机尽管十分庞大，但也可称其为“个人计算机”，因为计算机每次只能为单个用户服务，系统中所有的资源在这段时间内全部分配给该用户使用。

那时，程序员为了在计算机上算一道题，先要预约登记一段机时，到时他将预先准备好的表示指令和数据的插接板带到机房，由操作员将其插入计算机，并设置好计算机上的各种控制开关，启动计算机运行。程序和数据也可通过控制板上的开关直接送入计算机。假如程序员设计的程序是正确的，并且计算机也没有发生故障，若干小时后他就能获得计算结果，否则将前功尽弃，再约定下次上机时间。

汇编语言和高级语言的问世，以及程序和数据可以通过穿孔纸带或卡片装入计算机，改善了软件的开发环境，但计算机的操作方式并没有多大的改进。程序员首先将记有程序和数据的纸带或卡片装到输入设备上，拨动开关，将程序和数据装入内存；接着，程序员要启动汇编或编译程序，将源程序翻译成目标代码；假如程序中不出现语法错误，下一步程序员就可通过控制台按键设定程序执行的起始地址，并启动程序的执行。

在程序的执行期间，程序员要观察控制台上的各种指示灯以监视程序的运行情况。如果发现错误，并且还未用完所预约的上机时间，就可通过指示灯检查存储器中的内容，直接在控制台上进行调试和排错。如果程序运行正常，最终将结果在电传打字机等输出设备上打印出来。

总之，在早期的计算机系统中，每一次独立的运行都需要很多的人工干预，操作过程烦琐，占用机时多，也很容易产生错误。在一个程序的运行过程中，要独占系统的全部硬件资源，设备利用率很低。

2. 脱机输入/输出和批处理系统

(1) 脱机输入/输出

随着晶体管的使用和 CPU 速度的提高，计算机解题速度越来越快，但这种需要人工干预的由慢速设备读入程序和数据及由慢速输出设备打印结果的过程往往占用了用户使用计算机的大部

分时间。为了解决人工干预与 CPU 速度不匹配的矛盾，提高计算机的使用效率，在计算机中配备了“监控程序”。用户的控制命令和操作步骤可以写在源程序前或专门的控制卡片上。“监控程序”先读入控制命令，并按命令的指示一步一步自动地执行，这就是“操作系统”的雏形。

为了解决慢速输入/输出设备与 CPU 速度的不匹配问题，可将用户打在卡片上或纸带上的程序和数据通过外围小计算机预先输入到磁带上，运行时再从磁带上高速读入内存。输出也同样可通过磁带中转。由于使用了磁带作为输入/输出的中介，极大地提高了计算机的输入/输出速度。这种具体的输入/输出不需要在主计算机上进行的方式也称“脱机输入/输出”。图 1-1 描述了该系统的模型。



图 1-1 脱机输入/输出

(2) 批处理系统

脱机输入/输出进一步提高了计算机的运行效率，伴随着计算机处理速度不断的提高，在同样的时间内能解答更多的问题，但自前一个程序运行结束到启动后一个程序运行这段时间内，程序员或操作员还需要进行很多的人工干预。“批处理”是克服这个缺点的好方法。这种操作方法的基本思想是操作员取来一批作业，将它们输入到磁带中；操作系统先从磁带上将第一个作业读入内存，启动它运行，并将运行结果输出到另一条磁带上；当第一个程序运行完毕，操作系统能自动地从输入磁带上读入下一个作业，并予以运行和输出，如此直到整批作业全部处理完毕。

由于系统作业是成批地进行处理，但在内存中只能保持一个运行作业，故该类系统又称为单道批处理系统。批处理系统解决了高速计算机的运算、处理能力与人工干预之间的速度矛盾，实现了作业自动过渡。

3. 缓冲、中断和 DMA 技术

(1) 缓冲技术

脱机、批处理还没有完全解决 CPU 与外部设备速度的匹配问题，无论是从读卡机还是从磁带中读入，相对于 CPU 来说，读入数据的速度总嫌慢。进一步的解决方法是采用缓冲技术。缓冲技术的原理是将数据存放到一个特定的缓冲区中，当 CPU 从缓冲区中取得了数据，在对它进行运算之前，再启动输入设备以输入下一个数据至缓冲区。这样在 CPU 进行运算时，外部设备的输入工作也在同时进行。同样，输出也可采用缓冲技术。采用了带缓冲的输入/输出技术后，CPU 与外部设备能做到并行操作，减少了互相等待的时间，极大地提高了 CPU 与各种外部设备的使用效率。

(2) 中断技术

具有缓冲的输入输出也带来了新的问题。一个问题是处理机要知道 I/O 设备什么时候已完成了输入操作，以便处理输入数据并启动下一个 I/O。如果要靠 CPU 反复地查询输入设备的状态，就会浪费很多 CPU 的时间。中断技术的出现解决了这个难题。只要 I/O 设备一旦完成了输入/输出操作，它会自动向 CPU 发出中断信号。CPU 收到中断信号后，就暂停当前的处理工作，在做了一些必要的现场保护、中断处理工作后，转入中断服务程序。中断服务程序读出缓冲区数据，然后启动下一个 I/O 操作。从中断服务程序返回后，操作系统恢复被中断的运算过程。

中断处理机构需要在系统栈上保存中断返回地址，还要保护中断时的其他现场，如累加器和其他寄存器的值。在完成了中断服务之后，中断程序要恢复原先的中断现场，取得断口地址以使计算机继续原先的处理工作。CPU 在处理一个中断事务时，如果又收到另一个优先级更高的中断请求，就会暂停当前的中断服务，转而去处理更为紧迫的操作，这样就形成了中断的嵌套。

(3) DMA 技术

对于慢速的 I/O 设备（如键盘、串行口等），CPU 在执行有关的中断服务程序后，还可利用剩下的大部分时间来执行其他的计算工作。但对于像磁带、磁盘或高速网络通信接口，CPU 响应中断和处理数据所费的时间可以比数据到达的时间间隔更长。这样，即使处理机的时间全部用于处理中断和接收输入数据，也仍然会发生数据丢失的情况。

为了解决这个难题，产生了直接存储器存取（DMA）技术。一旦收到了 DMA 发来的中断请求后，CPU 在设置了缓冲区、指针和计数器后，DMA 就可以不需要 CPU 干预，在内存和设备之间传送整块数据。这样，通过 DMA 每传送一个数据块仅需要一次中断处理，而不是像低速设备那样每传送一个数据都需要一次中断处理。

4. SPOOLING

使用缓冲技术可提高 CPU 与外部设备工作的并行程度，如果 CPU 处理数据的速度比输入设备快得多，CPU 总是要等待输入设备将数据送入缓冲区后才能读取和处理数据，输出也会发生类似的情况。开始时 CPU 能全速运行，但不久所有的系统缓冲区都会被塞满，此后 CPU 必须等待输出设备取走缓冲区中的数据，以便可在其中存放新的输出结果。这种执行速度受到 I/O 设备限制的作业，称为受限于 I/O 的作业。另一方面，对于计算量很大的受限于 CPU 的作业，输入缓冲区经常是满的，而输出缓冲区经常是空的。因此，缓冲技术虽然是有用的，但在很多情况下，其作用并不十分明显。

磁盘系统的出现极大地改进了脱机输入输出的效果。磁带系统的问题是当 CPU 从磁带上读入数据时，读卡机等输入设备就不能在磁带的尾部写数据，所以慢速输入设备的数据不能通过磁带机被 CPU 联机地读入。磁盘设备消除了这个问题。磁盘的读写头很容易从磁盘中的一个区域移动到另一个区域，所以磁盘的读写位置能很快地从读卡机存入磁盘的区域移动到 CPU 需要读入下一个记录的区域。

在磁盘系统中，读卡机等设备将数据写到磁盘中，卡片数据的映像记录存放在由操作系统维护的一张表中。在一个作业执行期间需要请求读卡机输入数据时，实际读入的是存放在磁盘中的对应记录项。类似地，当作业要将输出送至打印机时，该输出实际上是通过系统缓冲区写到磁盘中，在该作业运行结束后，才由操作系统自动打印存储在磁盘中的输出结果。这种由操作系统将磁盘模拟为输入/输出设备的处理方式称为 SPOOLING（Simultaneous Peripheral Operating On Line），即“并行的外部设备操作联机”，也称为“假脱机”。SPOOLING 系统是以磁盘为几乎无限巨大的缓冲区来解决低速的 I/O 设备与高速 CPU 之间的速度匹配问题。

相比内存缓冲技术，SPOOLING 还具有其他优点：内存缓冲只能使作业的 I/O 与本身的计算工作重叠地进行，SPOOLING 能使多个作业的 I/O 与计算重叠地进行。使用 SPOOLING 技术，计算机能在执行一个作业时打印前面已完成了计算任务的作业的输出结果，还能读入尚未运行的作业，这样使得 CPU 和多台 I/O 设备能以很高速度并行地工作，提高了系统的吞吐量。

SPOOLING 还可提供一种很重要的结构——缓冲池。SPOOLING 通常把很多作业输入到磁盘中等待运行。操作系统就可根据系统当前的情况在这些作业中挑选下一个运行作业，以提高 CPU 和外部设备的利用率。当输入作业是放在读卡机或磁带中时，操作系统就不可能在这些作业中跳来跳去地以不同顺序运行作业，而只能按先来先服务的顺序运行作业。仅当作业是存放在像磁盘那样的直接存取设备中时，作业调度才成为可能。于是，操作系统就能使一些 CPU 受限作业和 I/O 受限作业互相搭配运行，以提高系统中各种设备的利用率。

5. 多道程序设计

脱机操作、缓冲和 SPOOLING 虽然能使 CPU 的计算与 I/O 设备的操作重叠地进行，却有一定的局限性。这些技术都不能使 CPU 和 I/O 设备时常保持忙碌状态。当一个作业必须等待 I/O 操作完成时（如从键盘读入一个数据），CPU 就不能执行下一步运算，往往只能空等。

作业调度使另一种新的计算技术——多道程序设计成为可能，能进一步提高 CPU 的利用率，使它几乎总有任务可执行，也能提高外部设备的利用率，使得多个作业的多种 I/O 操作可以并行地运行。

在多道程序设计系统中，操作系统可将多个作业存放在作业缓冲池中。在某一时刻，操作系统从缓冲池中挑选一个作业，并开始执行该作业。当执行中的作业因要等待用户键盘输入或等待其他设备 I/O 操作时，在单道程序设计中 CPU 就无事可干；在多道程序设计中，操作系统就可在缓冲池中挑选另一个作业，使其运行。当前一个作业结束了等待状态后就可再次获得 CPU，继续运行下去。只要系统中总是存在可执行的作业，CPU 就永远不会因无事可干而闲着。

这个思想很像一个厨师做菜。如果只有一个灶，那么在一段时间内他只能做一道菜，在热油或焖菜时厨师就无事可干。但厨师在一般情况下不只做一道菜，在他面前有好几个菜锅，当一些锅子里的菜正在焖或炖时，他就可以抽空炒某个锅里的菜，同时加热另一个锅里的油。只要有足够多的菜要做，厨师就不会闲着。

多道程序设计技术是相当复杂的。首先，为了支持多道程序设计，必须有足够大的内存以便同时容纳多个运行中的作业。假如内存不够，除了正在执行中的作业，其他作业就只能驻留在磁盘中，那么进行作业切换时，就要把内存中处于等待状态的作业调到外存去，再将外存中的作业调入内存执行，如此就会浪费很多的 CPU 时间。在内存中驻有多个作业就需要有比较复杂的存储管理和保护机构，同时还需要处理机调度机构，以决定在一些作业中选择哪一个占用 CPU，使其运行。除此之外，多道程序系统又需要提供各种外部设备的调度和管理功能。多道程序设计是现代操作系统的主旋律。典型的多道程序设计的操作系统有 IBM 的任务数固定的 OS/360MFT 系统和任务数可变的 OS/360MVT 系统。

1.3 现代操作系统类型

1. 分时系统

在批处理系统中开发和调试比较困难，因为在程序的执行期间用户不能与作业进行交互，他们必须预先编制好控制命令来规定作业该做些什么及处理各种可能发生的事件。在运行多步骤作业时，后继作业步骤的执行可取决于前面作业步骤的结果，对所有可能出现的情况，要完整地定义做什么和怎么做是非常困难的。

批处理系统适合于执行与用户很少交互的大作业，在一个作业的运行期间用户不能根据程序的运行状态和中间结果来调试和修改程序。用户提交作业后，在作业的运行时间内也不必等待，在隔了一个预定时间后，只要去取运行结果就是了。如果运行中间出错，一切只得从头做起。

交互式作业一般是由很多较短的动作序列组合而成，下一个命令的执行往往取决于当前命令的执行结果，而这往往是很难预见的。用户提交了一个命令后，要等待它执行结束，以便处理后继的任务，因此系统的响应时间应当较短。

其实，早期的系统也是交互式系统，因为整个系统是在程序员或操作员的直接控制之下运行的，这个特点使程序员能灵活、自由地开发和调试程序。但这样安排使处理机要等待程序员或操作者的命令，导致 CPU 大量的空闲等待时间。由于早期计算机的成本非常高，浪费 CPU 时间是不能容忍的。批处理系统的开发就是为了避免这个问题，从而改进了系统的运行效率，但在另一

方面也给用户调试和开发程序带来了极大的不便。

为了降低交互式系统的等待时间和运行时间的比率，系统通过多台终端同时向很多用户提供运行环境，这种分时系统就能以合理的成本向用户提供交互式使用计算机的方便。

操作系统使用多道程序设计技术来支持在一个计算机系统内运行多个用户的程序。每一个用户的程序都常驻在内存中，并按某一调度策略轮流运行。轮到某一用户程序运行时，它一次只能使用一段很少的时间，当分配给他的时间片用完或因等待 I/O 而不能继续运行下去时，就暂停该程序的运行，转而运行另一个用户的程序。当用户通过键盘命令与计算机交互时，即使输入的速度很快，但比起计算机来说还是极其缓慢的。计算机在所有用户之间快速切换，用户感觉不到需等待计算机要处理好别的用户的事务后才为自己服务。在分时系统中，用户觉得自己是在独自使用整个计算机系统。

由于同时有很多用户在使用计算机，当计算机刚处理好一个用户的输入后，就可利用等待用户做下一步动作的时间转而执行另一个用户的程序，这样 CPU 的空闲时间就减少了。即使大大减少了的等待时间也被众多的用户分担，从而降低了用户使用计算机的成本。

由于分时系统的突出优点，很多批处理的系统，如 IBM/360，被修改成也能同时支持分时系统。一些设计为分时系统的计算机也保留批处理的功能，以便在使用系统的用户数不多或晚上时，高效地运行一些不需用户干预的大作业，这样就可充分利用系统资源。现在，大部分系统能同时支持批处理和分时。

分时系统具有以下几个基本特征。

① 多路性，一台主机可连接多台终端，多个终端用户可以同时使用计算机，共享系统的硬/软件资源。

② 独立性，各用户操作互不干扰，每个用户都认为整个计算机系统被他所独占，为他服务。

③ 交互性，用户能与系统进行对话。在一个多步骤作业的运行过程中，用户能通过键盘等设备输入数据或命令，系统获得用户的输入后做出响应，显示执行的状况或结果。

④ 及时性，系统一般能在一秒钟内接受和响应用户的输入命令或数据，在数秒内显示命令的执行结果。

比较著名的分时系统有 CTSS 和 MULTICS。CTSS (Compatible Time Sharing System, 兼容分时系统) 是由美国麻省理工学院在 1963 年开发成功的最早的通用分时操作系统之一，其中一些关键的资源管理技术至今仍很有价值。MULTICS (MULTiplexed Information and Computing Service, 多路信息和计算系统) 是麻省理工学院、贝尔实验室和通用电气公司三家联合开发的分时系统。在该系统的管理中使用分页和分段的技术以及很多其他方面独特新颖的概念和思想，为现代操作系统的设计奠定了基础。

2. 实时操作系统

实时操作系统是一种能在限定的时间内对输入进行快速处理并做出响应的计算机处理系统。根据对响应时间限定的严格程度，实时系统又可分为硬实时系统和软实时系统。

硬实时系统主要用于工业生产的过程控制、航天系统的跟踪和控制、武器的制导等。这类操作系统要求响应速度非常快，工作极其安全可靠，否则有可能造成灾难性的后果。在一些重要的控制系统中，为了进一步提高系统的可靠性，除一台计算机控制系统工作外，还需要有一套后备系统。后备系统又可分为热备份和冷备份两种。

热备份就是除了一台当前工作的主控计算机外，另有一台相同的计算机与主控机同步运行，两者之间还定时交换运行状态的信息。当主控机发生故障时，控制立即被切换到同步运行的后备