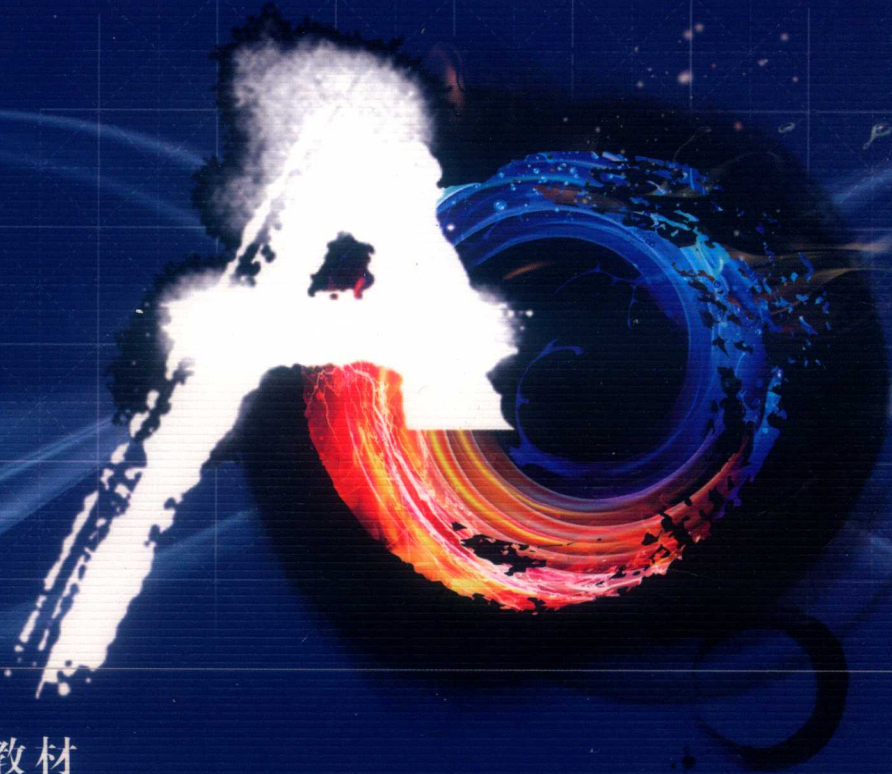




“十二五”普通高等教育本科国家级规划教材

国防科技大学国家精品课程教材·国家优秀教学团队教学成果



高等学校规划教材

操作系统 (第5版)

◎ 罗 宇 文艳军 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



“十二五”普通高等教育本科国家级规划教材

高等学校规划教材

国防科技大学国家精品课程教材·国家优秀教学团队教学成果

操作系统

(第5版)

罗 宇 文艳军 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是“十二五”普通高等教育本科国家级规划教材，也是国防科技大学国家精品课程教材和国家优秀教学团队教学成果。操作系统作为计算机系统的核心系统软件，负责控制和管理整个计算机系统的资源并支持用户以进程为单位高效协调地使用这些资源。本书以多道程序技术为基础，以通用操作系统主要功能部件为主线，介绍操作系统的概念、组成、功能、处理流程、设计等内容，包括：绪论，操作系统运行机制与用户界面，进程与处理机管理，进程同步与通信、进程死锁，存储管理，设备管理，文件系统，并行与分布式操作系统，保护与安全，系统虚拟机，Linux 操作系统实例。附录提供与课程配套的实验与课程设计参考资料。配套有实验教材《Linux 操作系统实验教程》，为任课教师免费提供电子课件和习题解答。

本书可作为高等学校计算机大类专业的教材或参考书，也可供从事计算机研究、开发、维护和应用的专业人员阅读。

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 / 罗宇，文艳军编著. —5 版. —北京：电子工业出版社，2019.6

高等学校规划教材

ISBN 978-7-121-36580-5

I. ①操… II. ①罗… ②文… 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 (2019) 第 096782 号

责任编辑：冉 哲

印 刷：三河市华成印务有限公司

装 订：三河市华成印务有限公司

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：20.25 字数：571 千字

版 次：2003 年 4 月第 1 版

2019 年 6 月第 5 版

印 次：2019 年 6 月第 1 次印刷

定 价：52.00 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, 88258888。

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。

本书咨询联系方式：ran@phei.com.cn。

前 言

操作系统是计算机中的核心系统软件,它负责控制和管理整个计算机系统的资源并支持用户高效协调使用这些资源,使计算机各部件在极大程度上并行运行。操作系统课程是计算机大类专业核心课程。随着计算机技术的发展,各类嵌入式系统得到广泛应用,其他相关专业也相继把操作系统作为一门重要的必修或选修课程。

本书阐述了操作系统的基本工作原理及设计方法,以多道程序技术为基础,以通用操作系统主要功能部件为主线,介绍操作系统的概念、组成、功能、处理流程、设计等内容。最后一章给出了当前流行的 Linux 操作系统设计实例。

本书是“十二五”普通高等教育本科国家级规划教材,也是国防科技大学国家精品课程教材和国家优秀教学团队教学成果。作者长期从事操作系统设计、开发和教学工作,根据自己 30 多年的科研和教学实践积累的经验,参考国内外近几年出版的教材和文献,结合科研开发工作对操作系统教学的要求,充分考虑当前我国计算机教育的现实情况,并参考了全国硕士研究生招生考试计算机科学与技术学科联考计算机学科专业基础综合考试大纲中的操作系统考查内容,编写了本书。作者剔除了传统操作系统教材已无实际使用价值的内容,增加了实用操作系统的典型处理方法,使本书的内容具有**先进性及实用性**;并且本着循序渐进的原则,采用通俗的语言和先进的实例,全面阐述操作系统的基本概念、原理、方法;既注重对操作系统经典内容的论述,又注意介绍操作系统的发展趋势及重要的研究开发成果,如第 10 章的系统虚拟机。修订后全书共 11 章,每章之后配有**小结及习题**,以加深理解。

第 1 章介绍什么是操作系统及操作系统的形成、发展与现状。

第 2 章介绍操作系统的运行机制与用户界面。

第 3 章介绍进程管理及线程的基本思想。

第 4 章介绍并发及死锁。

第 5 章介绍存储管理。

第 6 章介绍设备管理。

第 7 章介绍文件系统。

第 8 章介绍并行与分布式操作系统。

第 9 章介绍保护与安全。

第 10 章介绍系统虚拟机。

第 11 章介绍 Linux 操作系统实例。

另外,附录提供了与课程配套的实验与课程设计参考资料。

其中,前 7 章是操作系统的核心内容,可根据教学目的及课时安排选择增加第 8、9、10、11 章内容。本书适于 36~64 学时的课堂教学。建议在讲前 3 章时布置多进程(或多线程)编程、内核综合实验,穿插讲解习题及课程实验内容,也可利用本书**配套实验教材《Linux 操作系统实验教程》**安排实验。本书为任课教师**免费提供电子课件和习题解答**,可通过华信教育资源网 <http://www.hxedu.com.cn> 注册下载。本课程更多资源(含录像、课件、大纲、习题、问答、试题等)可参考网址 http://www.icourses.cn/sCourse/course_6560.html。

本书第9章及附录A~D由文艳军编写，其他部分由罗宇编写。陈燕晖、李思阳、晏益慧、张辉、许祥、张晨、韩明峰、邹鹏、邓胜兰为本书编写提供了素材或建议，在此表示感谢。

本书可作为高等学校计算机大类专业教材，对于具有高级程序设计语言初步知识和对计算机有一定了解的专业人士，亦是较全面的参考书。书中疏漏谬误之处恳请专家、读者指正。另特别感谢编辑冉哲以读者角度对本书的修改建议。

作者

于长沙·国防科技大学计算机学院

本书书号CIP数据核字(2019)第198782号

责任编辑

封面设计 荆门市成印务有限公司

印刷 荆门市成印务有限公司

出版 工业出版社

湖南分公司 湖南分公司

地址 湖南长沙 邮编 410000 电话 0731-88800000

定价 49.00元

ISBN 978-7-111-62025-5

本书为《Linux操作教程》的配套教材，主要介绍Linux操作系统的安装、配置、使用、维护等方面的知识。本书可作为高等院校计算机专业及相关专业的教材，也可供从事Linux系统管理的工程技术人员参考。

目 录

第 1 章 绪论	1
1.1 什么是操作系统	1
1.1.1 计算机系统的软件构成	1
1.1.2 操作系统作为特殊子程序	2
1.1.3 操作系统作为资源管理者	3
1.1.4 操作系统提供程序并发运行机制	4
1.2 操作系统的发展历史	4
1.2.1 监督程序	5
1.2.2 专用操作系统	8
1.2.3 多种方式操作系统	12
1.2.4 个人计算机操作系统、并行与分布式操作系统及其发展	12
1.3 主要操作系统介绍	14
1.3.1 Windows 系列及 MS DOS	14
1.3.2 UNIX 大家族	16
1.3.3 自由软件 Linux 和 freeBSD 等	19
习题 1	23
第 2 章 操作系统运行机制与用户界面	25
2.1 中断和异常	25
2.1.1 中断和异常的区别	25
2.1.2 中断分级	26
2.2 中断/异常响应和处理	28
2.2.1 中断/异常响应	28
2.2.2 中断/异常处理	29
2.3 操作系统运行模式	33
2.4 系统调用	35
2.5 人机界面	38
2.5.1 命令语言	39
2.5.2 图形化的用户界面	40
2.6 小结	42
习题 2	42

第 3 章 进程与处理机管理	44
3.1 进程描述	44
3.1.1 进程定义	45
3.1.2 进程控制块	46
3.2 进程状态	48
3.2.1 进程的创建与结束	49
3.2.2 进程状态变化模型	50
3.2.3 进程挂起	52
3.3 进程控制与调度	53
3.3.1 进程运行	53
3.3.2 进程调度	55
3.3.3 进程调度算法	58
3.4 作业与进程的关系	62
3.5 线程的引入	64
3.6 小结	66
习题 3	66
第 4 章 进程同步与通信、进程死锁	68
4.1 并行编程	68
4.1.1 并行编程方法	68
4.1.2 操作系统的支持	69
4.2 进程的同步与互斥	70
4.2.1 同步与临界段问题	71
4.2.2 解决临界段问题的硬件实现方法	72
4.2.3 信号量	74
4.2.4 管程	77
4.2.5 进程同步与互斥举例	77
4.3 消息传递	81
4.3.1 消息传递原理	82
4.3.2 消息传递示例	83
4.3.3 管道通信简介	84
4.4 死锁	85

4.4.1 死锁示例	85	6.3.1 常见存储外部设备	144
4.4.2 死锁定义	86	6.3.2 磁盘调度	147
4.4.3 死锁防止	88	6.3.3 磁盘阵列	150
4.4.4 死锁避免	90	6.4 小结	154
4.4.5 死锁检测	92	习题 6	154
4.4.6 死锁恢复	93		
4.4.7 死锁综合处理	94		
4.5 小结	94		
习题 4	95		
第 5 章 存储管理	99	第 7 章 文件系统	156
5.1 连续空间分配	99	7.1 文件结构	156
5.1.1 单道连续分配法	99	7.1.1 文件的概念	156
5.1.2 多道连续固定分区法	101	7.1.2 文件的逻辑结构	156
5.1.3 多道连续可变分区法	103	7.1.3 文件的物理结构	158
5.2 不连续空间分配	105	7.1.4 文件控制块	160
5.2.1 页式管理	106	7.2 文件目录结构	161
5.2.2 段式管理	109	7.2.1 一级目录结构	161
5.2.3 段页式管理	112	7.2.2 二级目录结构	162
5.2.4 改进的页式管理	113	7.2.3 树状目录结构	162
5.3 虚拟存储管理	114	7.2.4 无环图目录结构	163
5.3.1 页式虚存的基本思想	114	7.2.5 目录操作	164
5.3.2 页式虚存管理实现	115	7.3 文件存储器空间布局与管理	165
5.3.3 多级页表	117	7.4 文件访问接口	166
5.3.4 页替换策略	119	7.4.1 传统文件系统调用的实现	167
5.4 小结	125	7.4.2 存储映射文件访问	169
习题 5	126	7.5 文件保护	169
第 6 章 设备管理	129	7.5.1 文件访问保护	170
6.1 I/O 硬件概念	129	7.5.2 文件备份	171
6.1.1 常见外部设备分类	129	7.6 文件系统的基本模型	172
6.1.2 设备控制器 (I/O 部件)	130	7.7 FAT 文件系统磁盘布局	175
6.1.3 I/O 控制方式	131	7.8 小结	176
6.1.4 I/O 控制方式的发展过程	134	习题 7	177
6.2 设备 I/O 子系统	134	第 8 章 并行与分布式操作系统	178
6.2.1 设备的使用方法	134	8.1 并行操作系统	178
6.2.2 I/O 层次结构	136	8.1.1 对称多处理机	178
6.2.3 设备驱动程序	139	8.1.2 线程概念	180
6.2.4 缓冲技术	141	8.1.3 线程实现	186
6.3 存储设备	144	8.1.4 线程调度	190
		8.2 分布式系统	194
		8.2.1 分布式系统特点	195
		8.2.2 几种分布式系统应用模型	197
		8.2.3 分布式系统实现模型	200

8.2.4	分布式操作系统主要研究内容	202
8.2.5	分布式系统基础——通信协议层次简介	203
8.3	小结	206
习题 8		207
第 9 章	保护与安全	208
9.1	安全威胁	208
9.1.1	病毒	208
9.1.2	蠕虫	209
9.1.3	特洛伊木马	209
9.1.4	隐蔽通道	210
9.2	安全机制	210
9.2.1	硬件保护机制	210
9.2.2	标识与鉴别	211
9.2.3	存取控制	213
9.2.4	最小特权管理	214
9.2.5	安全审计	215
9.2.6	入侵检测	216
9.2.7	网络信息安全技术	217
9.2.8	软件漏洞防护技术	219
9.3	Linux 的安全机制	221
9.4	安全评测标准	223
9.4.1	TCSEC	223
9.4.2	GB 17859—1999	224
9.5	小结	225
习题 9		226
第 10 章	系统虚拟机	227
10.1	虚拟机概述	227
10.1.1	为什么需要虚拟机	227
10.1.2	虚拟机管理器的分类	228
10.1.3	系统虚拟化的优势及发展目标	228
10.1.4	虚拟机管理器的架构	229
10.2	CPU 的虚拟化	230
10.2.1	软件完全虚拟化	230
10.2.2	硬件辅助虚拟化	230
10.3	内存的虚拟化	232

10.3.1	软件实现内存虚拟化	232
10.3.2	EPT 页表辅助内存虚拟化	233
10.4	I/O 设备的虚拟化	233
10.4.1	完全的软件 I/O 设备虚拟化	233
10.4.2	半虚拟化设备模拟	234
10.5	小结	235
习题 10		235
第 11 章	Linux 操作系统实例	236
11.1	进程管理	236
11.1.1	进程与进程描述符	236
11.1.2	进程状态及切换时机	237
11.1.3	进程的调度算法	239
11.1.4	进程的创建与消亡	239
11.2	存储管理	242
11.2.1	物理内存的管理	242
11.2.2	进程地址空间的管理	244
11.3	文件系统	248
11.3.1	VFS	248
11.3.2	EXT2	253
11.3.3	主要文件系统中系统调用的处理流程	256
11.4	设备管理	258
11.4.1	设备文件的概念	258
11.4.2	设备模型基础	259
11.4.3	相关数据结构	260
11.4.4	块设备文件的 open() 和 read() 操作	262
11.5	中断、异常及系统调用	263
11.5.1	中断/异常的基本知识	263
11.5.2	异常处理函数	264
11.5.3	系统调用	264
11.5.4	中断的处理	265
11.5.5	软中断	267
11.6	SysV 进程间通信	268
11.6.1	共有的特性	268
11.6.2	信号量	270
11.6.3	消息队列	272
11.6.4	共享内存	273
习题 11		275

附录 A	bash 脚本编程简介	276
A.1	注释和简单命令	276
A.2	环境变量	276
A.3	控制结构	277
A.3.1	if 语句	277
A.3.2	case 语句	278
A.3.3	for 语句	278
A.3.4	while 语句和 until 语句	279
A.4	函数	279
附录 B	实现一个简单的 Linux 命令解释器	281
B.1	myshell 的语法	281
B.2	程序框架	281
B.3	命令行的语法分析	282
B.4	简单命令的执行	284
B.5	Makefile	284
附录 C	Linux 常用命令	285
C.1	用户终端命令	285

C.2	vi 编辑器的使用	292
-----	-----------	-----

附录 D	Linux 常用函数	295
------	------------	-----

D.1	进程管理函数	295
D.2	文件管理函数	296
D.3	进程间通信函数	298
D.4	多线程库函数	301

附录 E	Linux 内核综合实验指南	303
------	----------------	-----

E.1	实验目标与内容	303
E.2	实验基础知识	303
E.2.1	内核模块	303
E.2.2	系统调用	307
E.2.3	内存管理	309
E.2.4	其他	311
E.3	实验设计	313
E.4	实验步骤	314
E.5	常见问题	314

参考文献	316
------	-----

第 1 章 绪 论

计算机系统在国民经济和人们生活中起着越来越重要的作用。操作系统是计算机系统不可或缺的核心系统软件，是计算机系统的调度、控制中心。一方面，操作系统将裸机改造成为功能强大、各部件高效运行、使用方便灵活、安全可靠的虚拟机，为用户提供计算机系统的良好使用环境；另一方面，操作系统采用合理有效的方法组织多个用户任务共享计算机的各种资源，最大限度地提高资源的利用率。

自世界上第一台计算机 ENIAC 于 1946 年问世以来，计算机在运算速度、存储容量、元器件工艺及系统结构等方面都有了惊人的发展。以前，人们按照计算机元器件工艺的演变过程将计算机的发展划分为 4 个时代：电子管时代、晶体管时代、集成电路时代和大规模集成电路时代。与硬件发展相类似，人们也将操作系统的演变和发展过程划分为 4 个时代：单道批处理时代，多道批处理、分时和实时系统时代，同时具有多方面功能的多方式系统时代，并行与分布式系统时代。

本章将介绍什么是操作系统及操作系统在计算机系统中的地位和作用。并通过阐述操作系统历史的演变过程，使读者对操作系统的基本概念及技术的产生和发展等问题有一个直观的了解，从而使读者对不同类型操作系统的基本特征、今后的发展方向及目前流行的操作系统有更深刻的认识。

1.1 什么是操作系统

众所周知，处理机（CPU）、主存、辅存、终端、网络设备等硬件资源通过主板连接构成了看得见、摸得着的计算机硬件系统。为了能使这些硬件资源高效地、尽可能并行地被用户程序所使用，也为了给用户程序提供易用的访问这些硬件的方法，我们必须为计算机配备操作系统软件。操作系统的工作就是管理计算机的处理机、主存、外部设备等硬件资源，提供存放于存储设备中的文件等逻辑资源，并组织用户任务（如以进程的形式）使用这些资源。

操作系统是一种系统软件，是软、硬件资源的控制中心，它以尽量合理有效的方法组织单个或多个用户以多任务方式共享计算机系统的各种资源。

1.1.1 计算机系统的软件构成

计算机系统的软件层次及构成如图 1.1 所示。

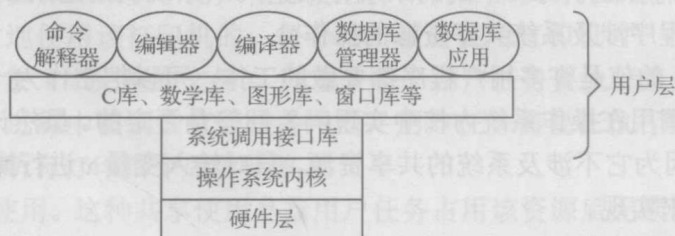


图 1.1 计算机系统的软件层次及构成

当用户在计算机中安装操作系统时，如图 1.1 所示的操作系统内核、命令解释器、编辑器、编译器，以及各种库程序，甚至数据库管理器、数据库应用等都从安装介质复制到计算机系统的磁盘中。

无论是安装 UNIX、Linux 还是 Windows 操作系统，命令解释器都是一个必不可少的程序。用户通过它来使用计算机。如果没有它，用户就无法操作计算机，无法输入命令让计算机去执行（注意，在窗口界面中，用户通过与命令解释器对应的程序管理器，如 Windows 的 explorer.exe，来使用计算机）。在现代操作系统实现中，命令解释器不是操作系统内核的组成部分，但它在使用计算机的过程中是不可缺少的。用户在终端输入的命令就是由命令解释器接收并解释执行的。其他的操作系统内核层之上的程序则是根据计算机的定位（服务器或工作站）而选择安装的。如果将计算机定位成开发程序用的工作站，那么用户必须安装编辑器进行程序编辑，并安装编译器进行程序编译。如果把计算机作为一个网络上的 Web 服务器，那么必须安装 Web 服务器程序。无论是用户自编的普通 C 语言程序还是数据库应用程序，都需要在安装并运行操作系统后再进行开发或安装。这些在操作系统内核层之上的程序，不管是命令解释器、Web 服务器，还是用户自编的程序，都要通过操作系统提供的进程机制来运行。

从狭义上看，操作系统只包含如图 1.1 所示的操作系统内核。它是一个非常重要的系统程序，管理着系统中所有的公共资源，并提供实现程序运行的进程机制。由于操作系统内核工作的重要性、特殊性，它必须在一种特殊的保护状态下运行，以免受到用户层程序的干扰和破坏，因此，它提供一组称为系统调用的接口，供上层程序调用，从而保证操作系统内核在特殊保护状态下运行，并且满足上层程序对系统资源的申请、使用、释放，以及进程的创建、结束等诸多功能的调用。

图 1.1 中的各种库程序实际上就是一些可以重用的、公用的子程序，它们提供形形色色的功能。系统提供这些库程序是为了方便用户编程。用户不必为了实现一个通用的功能再重写库程序代码，而只要引用库程序中的函数即可。库程序可以看成一些通用的、公共的程序集合，利用操作系统内核提供的简单的资源管理功能实现复杂的复合功能。这些通用的公共程序之所以不放到操作系统内核中实现，是因为它们不涉及系统公共资源的管理，也是为了控制操作系统内核的大小。

1.1.2 操作系统作为特殊子程序

从图 1.1 可以看出，操作系统内核位于计算机硬件之上。操作系统内核为用户层程序提供系统调用功能。系统调用与普通函数调用相似，可以看成对特殊的公共子程序的调用。因为这些程序提供了一些可以被任意用户层程序调用的公共功能，所以用户不需要再编写实现这些功能的程序，只要调用操作系统内核提供的相应“系统调用”即可。但是，要特别注意系统调用的特殊性，即系统调用处理程序运行在一种特殊的保护状态下。在这种状态下，程序可以执行一些特权指令，访问用户层程序访问不到的系统存储空间。系统调用之所以具有这样的特殊性，是因为系统调用处理程序涉及系统共享资源的操作。

举例来说，求 $\sqrt{x}$ 的值是许多用户程序都要做的工作，可以把它作为一个公共子程序实现。那么它需要作为系统调用在操作系统内核中实现吗？回答是否定的。虽然计算 $\sqrt{x}$ 需要许多条机器指令来实现，但是因为它不涉及系统的共享资源，只对输入变量 x 进行操作，所以可以把它作为数学库中的子程序来实现。

计算机都使用磁盘来存放系统或用户的程序及数据，存放在磁盘中的程序或数据称为文件。当一个用户程序要访问某个文件中的某段信息的时候，需要知道这段信息存放在磁盘哪个扇区中，需要向磁盘控制器发送读扇区的请求，需要查看扇区信息是否已经读入主存。如果这些操

作都交给用户来编程实现,不仅复杂,而且重复。因此,操作系统给用户提供一个简单的、统一的文件操作界面,即磁盘中可以包含许多文件,每个文件可以按照读/写方式打开,然后进行读/写操作,最后关闭文件。用户无须知道磁盘是怎么工作的、如何读/写数据,也不需要知道要读/写的文件放在磁盘的哪个扇区中,只需要知道读/写哪个文件的哪一段数据即可。利用这个简单的文件操作界面就可以与磁盘进行数据交换。至于确定文件信息在哪个扇区中、如何读/写这个扇区,则由操作系统的系统调用处理程序来实现。因为磁盘不是某个用户的私有资源,磁盘中的文件可以供多个用户访问,涉及磁盘和文件的管理数据都应该受到保护,所以文件操作以操作系统内核系统调用的形式实现。

1.1.3 操作系统作为资源管理者

计算机由处理机、主存、辅存、终端、网络设备等硬件资源组成。处理机提供程序执行能力;主存、辅存提供程序和数据的存储能力;终端提供人机交互能力;网络设备提供计算机间通信能力。这些硬件资源要被计算机用户高效地使用,必须有适合每种硬件资源特点的资源分配和使用机制。

为使硬件资源充分发挥作用,必须允许多用户或单用户以多任务方式同时使用计算机,以便让不同的资源由不同的用户任务同时使用,减少资源的闲置时间。例如,当一个用户任务将文件内容从辅存向主存的缓冲区读出时,另一个用户任务可以让自己的程序在处理机上运行。这样,处理机、主存、辅存同时工作,也就提高了资源利用率。

要让每种资源被多用户任务充分利用,就需要研究每种资源的特点。对于单处理机来说,它只能执行一个指令流。如果多个用户任务都要使用它,那只有让多个用户任务的程序分时地在处理机上运行,也就是说,处理机交替地运行多个用户任务中的程序。这意味着,操作系统要合理调度多用户任务来使用处理机。存储设备为程序和数据提供存储空间,只要多个用户的程序和数据按照规定的位置存放,互不交叉占用,它们就可以共存。操作系统要做的事就是管理存储空间,把适用的空间分配给用户的程序和数据使用;当用户任务访问这些程序和数据时要能够找到它们。

针对不同资源的特点,资源管理提供两种资源共享使用的方法:“时分”和“空分”。

① 时分是指由多个用户进程分时地使用该资源。除上述的处理机外,还有很多其他的资源也必须分时地使用,如外部设备控制器、网卡等,这些控制部件包含了控制 I/O 的逻辑,必须分时地使用。

② 空分是针对存储资源而言的。存储资源的空间可以被多个用户进程共同以分割的方式占用。

在时分共享使用的资源当中,有如下两种不同的使用方法。

① 独占式使用。独占表示某个用户任务占用该资源后,可以执行对资源的多个操作,使用一个完整的周期。例如,如果多个用户任务使用打印机,那么对打印机的独占式使用是指多个用户任务一定是分时地使用该打印机的;每个用户任务使用打印机时,执行了多条打印指令,打印了一个完整的对象(如完整的文件)。这里,每个用户任务要执行多条打印指令,为了不让多条打印指令在执行过程中被别的打印任务中断,用户任务需要在执行打印指令前申请独占该打印机资源,执行完所有打印指令后再释放。

② 分时式共享使用。这种共享使用是指用户任务占用该资源后无须使用一个逻辑上的完整周期。例如,对处理机的占用,用户任务随时都可以被剥夺处理机,只要运行现场保存好了,下次该用户任务再次占用处理机时就可以继续运行。再如,对磁盘的输入/输出,当一个用户任务让磁盘执行一个 I/O 请求后,其他用户任务也可向磁盘发出 I/O 请求,操作系统并不要求某个

用户任务的几个 I/O 请求之间不能插入其他用户任务的 I/O 请求。

操作系统应针对不同的资源类型，实现不同的资源分配和使用策略，并为资源分配、释放、使用提供相应的系统调用接口。

1.1.4 操作系统提供程序并发运行机制

用户可使用计算机进行科学计算、数据管理、通信、控制等工作。要实现所述的这些任务，必须执行相应的程序。用户使用处理机来执行程序，用程序驱动外部设备来进行数据交换，驱动网络设备来进行通信。用户的意图必须由程序及程序的输入参数表示出来。为了实现用户意图，必须让实现相应功能的程序得以执行；为了能让程序得以执行，需要由操作系统给程序及程序数据安排存放空间；为了能提高资源利用率，增加并发度，还必须让多个用户程序能分时占用处理机；为了能让一个程序还没运行完就允许另一个程序占用处理机运行，就必须保存上一个程序的运行现场。因此，必须要对实现各种用户意图的各个程序的执行过程进行描述和控制。

说明程序执行的状态、现场、标识等各种信息，有选择地调度某个程序占用处理机运行，这些工作必须由操作系统完成，这也是为了实现程序对处理机的分时占用。

操作系统一般用进程机制来实现程序的执行。

进程是指程序的执行，即程序针对某个数据集合的执行过程。操作系统的进程调度程序决定处理机如何在各执行程序间的切换。操作系统为用户提供进程创建和结束等的系统调用功能，使用户能够创建新进程以运行新的程序。

操作系统在系统初始化后，会为每个可能的系统用户创建第一个用户进程，用户的其他进程则可以由先前生成的用户进程通过“进程创建”系统调用陆续创建，以完成用户的各种任务。

在支持交互使用计算机的系统中，用户的第一个进程往往需要运行命令解释器程序（对于图形窗口终端用户而言，就是具有窗口界面的程序管理器，如 Windows 操作系统的 explorer.exe），这个程序会从终端获得用户输入的命令（或用户单击执行程序图标的信息），再进行相应的处理，可能会调用操作系统的“创建进程”系统调用，创建新进程去运行实现命令功能的程序。例如，在 Linux 操作系统控制的终端上输入：

```
$ cp /home/ly/test.c /home/wq/hello.c
```

那么，这一行字符串会由命令解释器程序获得，它会创建一个子进程，由子进程去运行 cp 实用程序，由 cp 实用程序建立一个新文件/home/wq/hello.c，并把/home/ly/test.c 文件中的内容读出来，写到/home/wq/hello.c 中。

1.2 操作系统的发展历史

在计算机刚刚诞生的 20 世纪 40 年代，计算机系统仅由硬件和应用软件组成。在这一时期，整个计算机系统是由用户直接控制使用的，所以又称为“手工操作”阶段。当时的计算机不仅速度慢、存储容量小，而且外部设备（简称外设）简单，主要使用磁带作为辅存，如图 1.2 所示。整个计算机系统由单个用户独占使用。当时用户使用计算机的大致方法是：将程序和数据以穿孔方式记录在卡片或纸带上，把卡片或纸带装在输入设备上；然后在控制台上形成输入命令，启动设备将卡片、纸带信息或磁带上的

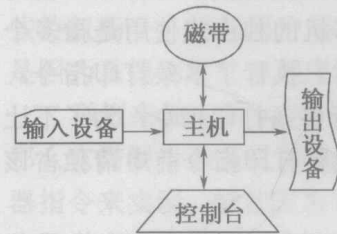


图 1.2 早期计算机系统

信息输入指定的主存单元；接着在控制台上指定主存启动地址，并启动程序运行；最后在打印机等输出设备上取得程序运行的结果。

显然,在这种使用方式下,用户在上机时独占全部资源,使用机器语言编写程序,并且对计算机各部分的工作直接实施人工干预,或者由用户自己所写的程序控制。在硬件各部分速度较低且程序较小的情况下,这种方式还能被人们所接受。但是,随着计算机速度的提高,以及 FORTRAN, COBOL 等高级程序设计语言的问世,这种方式势必使人无法忍受。

例如,用户要运行一个用 FORTRAN 语言编写的程序,必须先把存有 FORTRAN 编译程序的磁带安装在磁带机上,将 FORTRAN 编译程序和用户编写的 FORTRAN 源程序调入主存,并对 FORTRAN 源程序进行编译;然后再安装磁带上的“链接程序”,对编译好的程序进行链接,形成目标程序;最后启动目标程序运行。

由此可见,一批包括语言编译程序在内的系统软件的问世,使用户上机过程变得更繁杂,并增加了程序运行前的准备时间。由于计算机运算速度的提高,因此上述人工操作势必造成更大的资源浪费。为了缩短运行前的准备时间,提高计算机资源的利用率,人们提出了简单的改进措施,引入了“系统操作员”的概念。各用户将自己的程序及程序的运行步骤(控制意图)交给系统操作员,系统操作员将这种形式的一批用户作业按类进行划分,每次处理一类作业。

例如,将需要进行 FORTRAN 编译程序的作业组织成一类依次进行编译,并由系统操作员控制计算机运行用户程序。当然,这种使用计算机的方法仍旧停留在手工操作阶段,人的操作速度与机器的运行速度相比,仍存在极大的差距。由于人的操作缓慢,使得计算机资源在大部分时间被闲置,因此急需程序来代替人的手工操作。

1.2.1 监督程序

20 世纪 50 年代,为了减少系统操作员工作所花的时间,提高资源利用率,人们开始利用计算机系统中的软件来代替系统操作员的部分工作,从而产生了最早的操作系统——早期批处理系统。

1. 批处理系统

批处理系统的基本思想是:设计一个常驻主存的程序(监督程序, Monitor)。由系统操作员有选择地把若干用户作业合成一批,安装在输入设备中,并启动监督程序。然后,由监督程序自动控制这批作业运行。监督程序首先把第一道作业调入主存,并启动该作业;一道作业运行结束后,再把下一道作业调入主存启动运行。待一批作业全部处理结束后,系统操作员则把作业运行的结果一起交给用户。按照这种方式处理作业,各作业间的转换及各作业的运行完全由监督程序自动控制,从而减少了部分人工干预,有效地缩短了作业运行前的准备时间。

所谓作业(Job),是用户在一次上机活动中要求计算机系统所做的一系列工作的集合。从执行的角度看,作业由一组有序的作业步组成,如编译、运行分别称为不同的作业步。

当监督程序取代系统操作员的部分工作后,用户应以某种方式告知监督程序其作业的处理步骤。因此,在早期批处理系统中引出了“作业控制语言”和“作业控制说明书”的概念。作业控制说明书是使用作业控制语言编写的、用于控制作业运行的一段描述程序。在组织一道作业时,通常将作业控制说明书放在被处理的作业前面(或插入适当位置),由监督程序通过解释执行作业控制说明书中的语句来控制作业的运行。典型的卡片作业结构如图 1.3 所示。

这叠卡片中某些卡片表示了作业控制说明书中的语句。监督程序通过逐条解释、执行该作业控制说明书中的语句来自动控制作业运行,具体步骤说明如下。`$JOB` 语句说明该作业的名字、预计最大执行时间等信息。监督程序解释 `$FORTRAN` 语句的结果是,把 FORTRAN 编译程序调入主存,并启动其编译后面的源程序。编译结束后,控制返回给监督程序。监督程序解释 `$LOAD` 语句的结果是,通过链接程序把经过编译的程序链接起来,形成可执行程序。最后,解释 `$RUN` 语句,从而启动可执行程序运行。

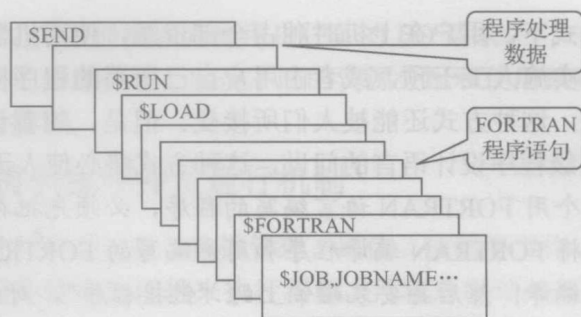


图 1.3 典型的卡片作业结构

监督程序内专设一个作业控制程序 (Job-Controller) 以控制作业的运行。作业的控制意图描述在作业控制说明书中。作业控制程序在控制某道作业运行时, 其实质性工作解释执行作业控制说明书中的语句, 实现对作业的控制。从逻辑上看, 作业由三部分组成: 源程序 (或程序)、数据及“加工”步骤。监督程序一旦接收到一道作业, 就开始根据“加工”步骤所规定的动作逐步完成对作业的加工活动。

如果用户可以使用全部的机器指令, 就可以直接控制和使用系统资源 (如主存、外部设备等), 但是用户编程中的错误往往可能导致各种预想不到的后果。为了避免这类错误发生, 人们将机器指令分为“普通指令”和“特权指令”, 并且引入了“模式/态” (Mode) 的概念。把有关 I/O 的指令、对特殊寄存器的访问等列为特权指令, 并且规定只有监督程序才有权执行特权指令, 用户程序只能执行普通指令。将 I/O 指令列为特权指令后, 用户便不能直接控制设备进行传输了。如果用户希望进行 I/O 操作, 则必须向监督程序提出请求; 监督程序通过调用系统内部的程序段来完成用户的 I/O 请求。由此引出了“系统调用” (System Call) 或称“广义指令”的概念。

监督程序为用户提供一系列分别完成各种不同功能的系统调用程序段。用户程序中可以用一条特殊的硬件转移指令 (不带转移地址, 直接转到操作系统与硬件约定的地址) 请求一次特定的 (通过编号区分) 系统调用。当处理机执行到用户程序中的系统调用指令时, 硬件通过产生“自陷” (Trap) 并借助转换机制将当前的用户模式转变为监督模式, 随之转入监督程序运行。监督程序根据用户提供的调用参数进行相应的处理, 如完成 I/O 操作等功能。处理结束后, 监督程序根据“自陷”前所保存的现场将模式改变为用户模式, 退回用户程序继续执行。

“系统调用”概念的引入提高了监督程序在整个系统中的地位, 丰富了监督程序的功能。监督程序不仅对作业的处理流程进行自动控制, 而且还负责为用户程序的运行提供各种功能的服务。系统调用的引入也为用户提供了使用计算机系统的新接口, 使用户从直接使用物理处理机的繁杂束缚中解脱出来, 呈现在用户面前的是一台功能强大、使用方便的虚拟处理机。引入系统调用后, 用户对系统内部各种资源的使用均由监督程序代为完成, 因而也使系统更加安全, 既避免了用户在编程使用资源时可能出现的某些错误, 也有利于提高资源利用率。

在手工操作阶段, 存储器全部由用户支配使用。引入监督程序后, 存储器不再由用户独占, 常驻主存的监督程序必须占用部分主存空间。通常, 监督程序占用主存的 $0 \sim k$ 单元, 而 $k+1 \sim n$ 单元由用户程序占用。监督程序所在的存储空间称为“系统空间”, 用户程序所在的存储空间称为“用户空间”。为了避免用户程序执行时有意或无意地对系统空间进行存取访问, 在硬件方面提供一个界地址寄存器, 用于存放系统空间与用户空间的分界地址。若系统处于用户模式, 每次访问主存时, 硬件会自动进行地址越界检查, 从而保证了监督程序不被破坏。这种保护称为“存储保护”。

在早期批处理系统中, 当系统动态运行时, 可能一段时期处于监督模式, 一段时期又处于用户模式。从用户模式进入监督模式主要是由用户程序中的系统调用引起的。例如, 用户请求

设备 I/O 或请求结束运行。但是，若在用户程序执行过程中永远不出现系统调用，或者永远不出现请求结束运行的系统调用（如用户程序进入了“死循环”），监督程序便失去了作用。为了防止这种情况发生，人们设置了“定时器中断”。

定时器（Timer）是一个硬件计数器，其计时长度可以根据需要而调整。计数器根据硬件的计时周期自动计时，计数器满后便发生定时器中断。用户程序执行时，若碰到定时器中断，则无条件进入监督程序。监督程序根据当前作业控制说明（或规定）的“最大运行时间”值来判断该程序是否进入了“死循环”，从而可以有效地防止某个用户程序长期垄断处理机的现象。

引出上述概念后，早期批处理系统中的监督程序工作流程如下。

- (1) 判断输入设备中是否有待输入的作业，如果没有，则等待作业输入。
- (2) 从输入设备输入一道作业。
- (3) 控制作业运行。

① 取一条作业控制说明书中的语句，解释执行。如果是一条“作业终止”语句，则删除该作业，转第（1）步。

② 如果当前是一条“执行性语句”（如请求编译、请求运行用户程序等），则在主存中建立相应程序的运行环境，并分配处理机，开始在用户模式下执行该程序。

③ 在用户模式程序的执行过程中，如果发生“中断”事件（如 I/O 中断、系统调用、程序执行错误等），将控制转入监督程序。当“中断”事件处理结束后，返回用户态，用户程序继续执行。

④ 用户程序执行结束后，进入监督程序，控制转第①步，取下一条作业控制说明书中的语句执行。

监督程序如同一个系统操作员，它负责批处理作业（即按照作业控制说明书运行的作业）的 I/O，并自动根据作业控制说明书，以单道串行的方式控制作业运行，同时在程序运行过程中通过提供各种系统调用的方式来控制计算机资源的使用。虽然监督程序并不能被称为操作系统（它与操作系统的本质差别在于监督程序不具有并发控制机制），但监督程序在系统中的地位和作用、实现的基本目标及管理资源的基本方法与操作系统类似。真正的操作系统就是在此基础上进一步发展和完善起来的。

与手工操作阶段相比，监督程序的引入有效地减少了人工干预时间和作业运行前的准备时间，相对提高了处理机的利用率。但是，在计算机运算速度大幅度提高的形势下，用这种方法管理计算机远不能满足需要。首先，在一个处理机上运行的程序启动 I/O 操作时，处理机被迫处于空闲状态或忙等待（Busy_Wait）状态，也就是说，处理机启动 I/O 操作后，再循环判断 I/O 操作是否完成，而没有做实质性工作，这将导致高速的处理机受到慢速外部设备的牵制，从而使处理机无法充分得到利用。

2. 利用脱机 I/O 技术改善系统性能

因为作业的 I/O 操作与作业的运行是串行的，所以受卡片机、光电机、打印机等慢速 I/O 设备的影响，处理机的利用率难以提高。为了进一步提高系统的工作效率，必须解决低速 I/O 的问题。磁带机的传输速度比卡片机、光电机和打印机的速度快，若用磁带机代替这类低速设备便可进一步缩小处理机与外部设备之间在速度上的差异。历史上，人们曾采用脱机 I/O 技术实现作业的 I/O 操作，其系统模型如图 1.4 所示。

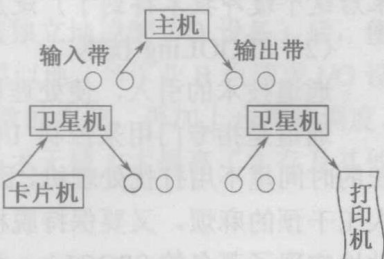


图 1.4 脱机 I/O 技术系统模型

在采用脱机 I/O 技术的系统中, 主机的所有 I/O 都是通过磁带机进行的。用户的作业由另一台能力较弱、价格较低的卫星机负责从卡片机传输到磁带(称为输入带)上, 然后系统操作员将输入带安装到与主机相连的磁带机上。主机在处理输入带上的作业时, 将产生的输出结果直接送到输出带上。系统操作员再将输出带安装到卫星机上, 由卫星机负责将输出带上的信息从打印机上输出。由于磁带机的速度比慢速 I/O 设备(如卡片机、打印机)的快, 因而按照这种脱机方式控制作业的 I/O 操作, 可以减少其所花费的时间, 有效地提高处理机的利用率。如果将一台主机与多台卫星机有机地组合, 使速度得到最好的匹配, 则可以大幅度提高系统的处理能力。从 20 世纪 50 年代末到 60 年代初, 这种脱机处理方式被广泛地应用于批处理系统中。

无论如何, 由于处理机与 I/O 设备是以串行方式工作的, 也就是说, 当处理机工作时, I/O 设备闲着; 当 I/O 设备工作时, 处理机处于忙等待, 这就限制了设备的利用率。另外, 从方便用户的角度来说, 采用这种批量处理的控制方法, 用户不能以交互方式使用计算机, 从而限制了计算机的灵活使用。随着对这些问题不断深入的研究和解决, 逐步形成了第 2 代操作系统。

1.2.2 专用操作系统

20 世纪 60 年代初, 计算机硬件有了很大的发展。例如, 主要元器件由电子管变成了晶体管, 出现了磁盘、通道、终端等部件。而这些硬件的发展为监督程序提出了新的研究课题, 也为操作系统的形成提供了重要的物质基础。这一时期是操作系统形成的重要时期。随着计算机应用的巨大牵引, 不仅批处理系统得到了充分的发展, 而且还出现了实时(Real Time)、分时(Sharing Time)等不同类型的系统。

1. 多道批处理系统

在早期批处理系统(也称单道批处理系统)中, 处理机与 I/O 设备以串行方式工作, 故两者的利用率较低。为了提高资源利用率, 人们开始使用 I/O 缓冲、SPOOLing 等技术, 尤其是引入了“多道程序设计”(Multiprogramming)的思想, 使单道批处理系统发展为多道批处理系统。

(1) 利用 I/O 缓冲实现异步编程

在单道批处理系统中, 作业的处理过程是单道串行的, 所以在监督程序的控制下, 处理机与外部设备也按串行方式工作。为改变这种串行工作方式, 人们首先采用了缓冲(Buffering)技术使两者在一定程度上并行操作。

例如, 在主存中建立两个长度相同的缓冲区: B0, B1。对于一批待输入的信息, 首先将其中的一个记录从外部设备中读入 B0, 读完后接着将下一个记录从外部设备中读入 B1, 与此同时, 处理机开始处理 B0 中的记录。待处理机的处理工作与输入工作均结束后, 则将下一个记录读入 B0, 处理机同时处理 B1 中的记录。如此重复, 直至将信息全部输入并处理完毕, 这种利用双缓冲区实现的 I/O 操作在一定程度上实现了处理机与外部设备并行工作。这类并行的实现要求 I/O 设备有较强的功能, 能够不依赖于处理机实现外部设备与主存独立地交换数据, 在引入通道技术后这个缓冲技术得到了广泛应用。

(2) SPOOLing 技术

通道技术的引入, 使处理机与外部设备并行操作成为可能。

通道是指专门用来控制 I/O 的硬件装置, 可以实现外部设备与主存直接交换数据, 在相当长的时间里不用打扰处理机, 因此这时处理机可以去干别的事情。为了能够消除脱机 I/O 带来的人工干预的麻烦, 又要保持脱机 I/O 系统中作业高速入/出主存的特点, 人们借助通道和磁盘成功地实现了著名的 SPOOLing 系统。通道也可看成专门的 I/O 处理机, 磁盘则是一种比磁带更快