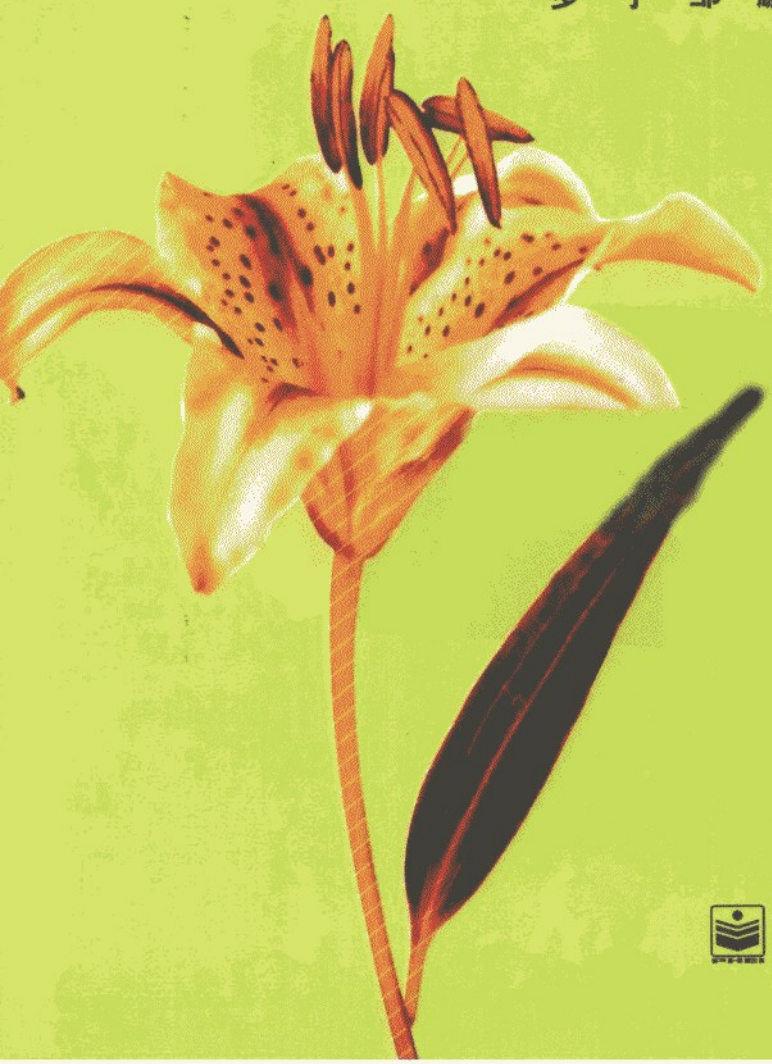
国家精品课程教材·国家优秀教学团队教学成果

操作系统

（第3版）

罗 宇 邹 鹏 邓胜兰 等编著



高等学校工程创新型"十二五"规划计算机教材

Engineering Innovation

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

操作系统（第3版）

高等学校工程创新型『十二五』规划计算机教材

大学计算机基础（第2版）
计算机应用基础
C语言大学实用教程（第2版）
C语言大学实用教程学习指导（第2版）
C++简明教程
C++语言程序设计
C++面向对象程序设计（第2版）
大学C/C++语言程序设计
大学C/C++语言程序设计实验教程
Java程序设计简明教程
双语版Java程序设计
计算机网络
网络协议工程
操作系统（第3版）
Linux操作系统实验教程
操作系统分析
计算机组成原理（第2版）
计算机组成原理学习指导与课程设计
计算机组成原理与汇编语言程序设计（第2版）
计算机组成技术教程
微机原理与接口技术
80X86/Pentium微型计算机原理及应用（第2版）
数据结构实用教程（C++版）
C++与数据结构（第2版）
数据库系统原理（第3版）
数据库原理与设计
数据库原理与设计实践教程
计算机图形学
汇编语言程序设计教程（第3版）
Win32汇编语言程序设计教程（第2版）
大型机系统应用基础
高级IT项目监理
射频识别（RFID）安全技术

软件工程及嵌入式方向

软件工程（第2版）
基于CMMI的软件工程（第2版）
Web程序设计基础教程（第2版）
Web项目开发
猜测、实证与构建——软件开发学习与提高
软件开发技术
嵌入式系统导论——知识体系、产业链及发展方向
嵌入式实时操作系统的设计与开发
嵌入式网络编程
——串口通信、工业总线、传感器网络应用开发
基于Java ME和Android的嵌入式应用开发
Java ME嵌入式程序设计
Java EE基础教程
计算机控制系统——工程实践的系统分析与设计

信息安全及网络安全方向

组合数学（第2版）
信息安全导论
信息安全协议理论与技术（第2版）
密码学基础（第2版）
网络与系统攻击技术（第2版）
PKI原理与技术（第2版）
计算机网络安全与防护（第2版）

计算机技术及应用

Excel数据处理与统计初步（第4版）
多媒体技术
Web程序设计——ASP.NET（C#）
网站规划与设计
数据挖掘基础
数据挖掘理论与技术
网络环境下文献信息资源检索与利用通用教材
Linux网络管理及应用（第2版）

ISBN 978-7-121-13613-9
9 787121 136139 >
定价：36.00元

策划编辑：童占梅
责任编辑：童占梅
封面设计：一克米工作室

高等学校工程创新型“十二五”规划计算机教材
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3版）

罗 宇 邹 鹏 邓胜兰 等编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是国防科技大学国家精品课程和国家优秀教学团队建设成果。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组织用户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、功能、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，Linux 操作系统实例。附录提供与课程配套的实验参考资料。配套出版了实验教材《Linux 操作系统实验教程》，为任课教师免费提供电子课件。

本书可作为高等学校计算机科学与技术、软件工程、通信与电子信息等相关专业教材和参考书，也可供从事计算机研究、开发、维护和应用的专业人员阅读。

图书在版编目（CIP）数据

操作系统 / 罗宇等编著. —3 版. —北京 ：电子工业出版社，2011.6
高等学校工程创新型“十二五”规划计算机教材
ISBN 978-7-121-13613-9

Ⅰ. ①操… Ⅱ. ①罗… 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（2011）第 091942 号

策划编辑：童占梅
责任编辑：童占梅
印　　刷：北京市顺义兴华印刷厂
装　　订：三河市双峰印刷装订有限公司
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：787×1 092　1/16　印张：19.75　字数：502 千字
印　　次：2011 年 6 月第 1 次印刷
印　　数：3 000 册　　定价：36.00 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，联系电话：（010）68279077；邮购电话：（010）88254888。

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

服务热线：（010）88258888。

前　言

补充。本次修订由罗宇、邹鹏、邓胜兰完成，罗宇负责统稿；陈燕晖、文艳军、晏益慧、张辉、许祥为本书修订提供了素材，做出了贡献。

本书可作为高等院校计算机及相关专业教材，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。

编著者

于长沙·国防科技大学计算机学院

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 …… 5
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 …… 13
1.2.4　PC 操作系统、并行与分布式操作系统及发展 …… 13
1.3　主要操作系统介绍 …… 14
1.3.1　Windows 系列及 MS DOS …… 15
1.3.2　UNIX 大家族（SVR4，BSD，Solaris，AIX，HP UX） …… 17
1.3.3　自由软件 Linux 和 freeBSD 等 …… 21
习题 1 …… 25
第 2 章　操作系统运行机制与用户界面 …… 26
2.1　中断和异常 …… 26
2.1.1　中断和异常的区别 …… 26
2.1.2　中断的分级 …… 27
2.2　中断/异常响应和处理 …… 29
2.2.1　中断/异常响应 …… 29
2.2.2　中断/异常处理 …… 31
2.3　操作系统运行模型 …… 34
2.4　系统调用 …… 36
2.5　人机界面 …… 40
2.5.1　命令语言 …… 40
2.5.2　图形化的用户界面 …… 42
2.6　小结 …… 44
习题 2 …… 44
第 3 章　进程与处理机管理 …… 45
3.1　进程描述 …… 45
3.1.1　进程定义 …… 46
3.1.2　进程控制块 …… 47
3.2　进程状态 …… 49
3.2.1　进程的创建与结束 …… 50
3.2.2　进程状态变化模型 …… 51
3.2.3　进程挂起 …… 53
3.3　进程控制与调度 …… 55
3.3.1　进程执行 …… 55
3.3.2　进程调度 …… 57
3.3.3　调度算法 …… 60
3.4　作业与进程的关系 …… 64
3.5　线程的引入 …… 66
3.6　小结 …… 68
习题 3 …… 68
第 4 章　进程同步与通信、进程死锁 …… 70
4.1　并发执行的实现 …… 70
4.1.1　并发编程方法 …… 70
4.1.2　并发执行的实现 …… 71
4.2　进程的同步与互斥 …… 72
4.2.1　同步与临界段问题 …… 73
4.2.2　解决临界段问题的硬件实现方法 …… 74
4.2.3　信号量 …… 76
4.2.4　管程 …… 79
4.2.5　进程同步与互斥举例 …… 80
4.3　消息传递原理 …… 84
4.3.1　消息传递通信原理 …… 84
4.3.2　消息传递通信示例 …… 85
4.3.3　管道通信简介 …… 87
4.4　死锁 …… 87
4.4.1　死锁示例 …… 87
4.4.2　死锁定义 …… 89
4.4.3　死锁防止 …… 92
4.4.4　死锁避免 …… 93
4.4.5　死锁检测 …… 95

4.4.6 死锁的恢复……97
4.4.7 死锁综合处理……97
4.5 小结……98
习题 4……99
第 5 章 存储管理……103
5.1 连续空间分配……103
5.1.1 单道连续分配……103
5.1.2 多道固定分区法……106
5.1.3 多道连续可变分区法……107
5.2 不连续空间分配……110
5.2.1 页式管理……110
5.2.2 段式管理……114
5.2.3 段页式管理……116
5.2.4 改进的页式管理……118
5.3 虚拟存储管理……118
5.3.1 页式虚存的基本思想……119
5.3.2 页式虚存管理实现……119
5.3.3 多级页表……122
5.3.4 页面替换策略……124
5.4 小结……130
习题 5……131
第 6 章 设备管理……134
6.1 I/O 硬件概念……134
6.1.1 常见外部设备分类……134
6.1.2 设备控制器（I/O 部件）……135
6.1.3 I/O 控制方式……136
6.1.4 I/O 控制方式的发展过程……139
6.2 设备 I/O 子系统……139
6.2.1 设备的使用方法……139
6.2.2 I/O 层次结构……142
6.2.3 设备驱动程序……144
6.2.4 缓冲技术……147
6.3 存储设备……150
6.3.1 常见存储外部设备……150
6.3.2 磁盘调度……153
6.3.3 磁盘阵列……156
6.4 小结……160
习题 6……161
第 7 章 文件系统……162
7.1 文件结构……162
7.1.1 文件概念……162
7.1.2 文件的逻辑结构……163
7.1.3 文件的物理存储……164
7.1.4 文件控制块……166
7.2 文件目录结构……167
7.2.1 一级目录结构……167
7.2.2 二级目录结构……168
7.2.3 树形目录结构……169
7.2.4 无环图目录结构……170
7.2.5 目录操作……171
7.3 文件存储器空间布局与管理……171
7.4 文件访问接口……173
7.4.1 传统文件系统调用的实现……173
7.4.2 存储映像文件访问……175
7.5 文件保护……176
7.5.1 文件访问保护……176
7.5.2 文件备份……178
7.6 文件系统的基本模型……179
7.7 FAT 文件系统磁盘布局……182
7.8 小结……183
习题 7……184
第 8 章 并行与分布式操作系统……185
8.1 并行操作系统……185
8.1.1 对称多处理机……185
8.1.2 线程概念……187
8.1.3 线程实现……193
8.1.4 线程调度……198
8.2 分布式系统……202
8.2.1 分布式系统特点……203
8.2.2 几种分布式应用模型……205
8.2.3 分布式系统实现模型……208
8.2.4 分布式操作系统主要研究内容……210
8.2.5 分布式系统基础——通信协议层次简介……211
8.3 小结……214
习题 8……215
第 9 章 保护与安全……216
9.1 安全威胁……216
9.1.1 病毒……216

9.1.2 蠕虫……217
9.1.3 特洛伊木马……217
9.1.4 隐蔽通道……218
9.2 安全机制……218
9.2.1 硬件保护机制……219
9.2.2 标识与鉴别……219
9.2.3 存取控制……221
9.2.4 最小特权管理……222
9.2.5 安全审计……223
9.2.6 入侵检测……224
9.2.7 网络信息安全技术……225
9.3 Linux 的安全机制……228
9.4 安全评测标准……230
9.4.1 TCSEC 橘皮书……230
9.4.2 中国国标 GB17859－1999……232
9.5 小结……232
习题 9……233
第 10 章 Linux 操作系统实例……234
10.1 进程管理……234
10.1.1 进程与进程描述符……234
10.1.2 进程状态及切换时机……235
10.1.3 进程的调度算法……237
10.1.4 进程的创建与消亡……238
10.2 存储管理……240
10.2.1 物理内存的管理……240
10.2.2 进程地址空间的管理……243
10.3 文件系统……246
10.3.1 VFS……246
10.3.2 EXT2 文件系统……252
10.3.3 主要文件系统中系统调用的处理流程……256
10.4 设备管理……258
10.4.1 设备文件的概念……258
10.4.2 设备模型基础……259
10.4.3 相关数据结构……259
10.4.4 块设备文件的 open()，read()操作……262
10.5 中断、异常及系统调用……263
10.5.1 中断/异常的基本知识……264
10.5.2 异常处理函数……264
10.5.3 系统调用……265
10.5.4 中断的处理……265
10.5.5 软中断……268
10.6 SysV 进程间通信……269
10.6.1 共有的特性……269
10.6.2 信号量……271
10.6.3 消息队列……273
10.6.4 共享内存……274
习题 10……277
附录 A bash 脚本编程简介……278
A.1 注释和简单命令……278
A.2 环境变量……278
A.3 控制结构……279
A.3.1 if 语句……279
A.3.2 case 语句……280
A.3.3 for 语句……281
A.3.4 while 语句和 until 语句……281
A.4 函数……282
A.5 小结……283
附录 B 实现一个简单的 Linux 命令解释器……284
B.1 myshell 的语法……284
B.2 程序框架……284
B.3 命令行的语法分析……285
B.4 简单命令的执行……288
B.5 Makefile……288
B.6 小结……288
附录 C Linux 常用命令……289
C.1 用户终端命令……289
C.2 vi 编辑器的使用……297
附录 D Linux 常用函数……299
D.1 进程管理函数……299
D.2 文件管理函数……301
D.3 进程间通信函数……303
D.4 多线程库函数……306
参考文献……308

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、系统各部件高效运行、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、主存、磁盘、终端、网卡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、外设等硬件资源，提供存放于存储设备的文件等逻辑资源，并组织用户任务（如以进程形式）使用这些资源。

操作系统是一种系统软件，是软、硬资源的控制中心，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。

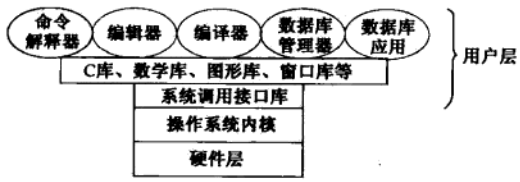


图 1.1 计算机系统的软件层次及构成

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、各种库程序，甚至数据库管理器、Web 服务器等都从安装介质复制到了计算机系统的磁盘上。

无论是安装 UNIX、Linux 还是 Windows 操作系统，命令解释器都是必不可少的一个程序，用户通过它来使用计算机系统。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，命令解释器程序没有作为操作系统内核的组成部分，但它在使用计算机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程序接收并解释执行的。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程序开发用的工作站，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，都在操作系统安装并运行后开发或安装。这些在操作系统内核层之上的程序，不管是命令解释器、Web 服务器或用户自编的程序，都是通过操作系统提供的进程机制来运行的。

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，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的需求，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、结束等诸多功能的需求。

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上述库程序代码，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去实现，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（又称广义指令）功能。系统调用与普通函数调用相似，可以看成是特殊的公共子程序，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，但因为它不涉及系统的共享资源，只对输入变量 x 进行操作，因此可以把它作为数学函数库中的子程序来实现。

以前的计算机都使用软盘，许多基于 Intel x86 的个人计算机都使用 NEC 的 PD765 软盘驱动器（简称软驱），该软驱支持 16 条命令，可以通过对软驱控制部件中的寄存器置不同的值来执行初始化、移动磁头、读/写数据等命令。其中，最基本的命令是读/写命令，需要 13 个参数，如磁盘块地址、每个磁盘的扇区数、物理介质中所用的记录模式、扇区间距等。当操作完成时，软驱控制部件中的状态寄存器中有一堆状态位，由驱动程序判定是正常完成还是异常结束。在启动读/写命令前还需要判定软驱电机是否已启动，若未启动还需要先启动电机。如果这些操作都交给用户编程实现，不仅复杂，而且每个用户都要重复编程，多个用户使用时还会引起混乱。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，即软盘上包含多个文件，每个文件可以按照读/写方式打开，然后进行读/写，最后关闭文件。用户无需知道电机如何启动、如何读/写数据，也不需要知道要读/写的数据放在软盘的哪个扇区，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软盘进行数据交换。这个文件操作界面由操作系统的系统调用实现，因为软驱不是某个用户的私有资源，软盘上的文件可以供多个用户访问，涉及到软驱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、终端设备、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；终端设备提供人机交互能力；网络设备提供机间通信能力。这些硬件资源要能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磁盘往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、磁盘同时工作，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使用处理机。存储设备为程序和数据提供存放空间，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，它们是可以共存的，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。

针对不同资源特点，资源管理包含两种资源共享使用的方法：“时分”和“空分”。

① 时分就是由多个用户进程分时地使用该资源，除了上述的处理机外，还有很多其他的资源也必须分时地使用，如外设控制器、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，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。独占表示某用户任务占用该资源后，执行对资源的多个操作，使用一个完整的周期。例如，如果多用户任务使用打印机，那么对打印机的独占式使用是指多用户

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无需使用一个逻辑上的完整周期。例如，对处理机的使用，用户任务随时都可以被剥夺 CPU，只要运行现场保存好了，下次该用户任务再次占用 CPU 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条 I/O 请求后，其他用户任务又可向磁盘发 I/O 请求，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执行；为了能让程序执行，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，还必须能让多个用户程序能分时占用处理机；为了能够让一个程序还没运行完就让另一个程序占用 CPU 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 CPU 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 CPU 的分时使用。

操作系统一般用进程机制来实现程序的执行。

进程是指运行当中的程序，也就是指程序针对于某一数据集合的执行过程。操作系统的进程调度程序决定 CPU 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 文件的内容读出来，写入 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，而且外部设备简单，辅存主要借助磁带，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信息输入到指定的主存单元；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。

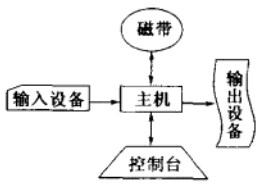


图 1.2 早期计算机系统

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，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量较小的情况下，这种方式还能被人们所接受。但是，随着计算机速度的提高和 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，用户如要想运行一个用 FORTRAN 语言编写的程序，必须首先把存有 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“连接程序”，对编译好的程序进行连接，形成目标程序；最后启动目标程序运行。

由此可见，由于一批包括语言编译器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。由于计算机速度的提高，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，使得计算机资源大部分时间闲置，因此急需用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），操作员有选择地把若干用户作业合成一批，安装在输入设备上，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，各作业间的转换以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。作业控制说明书是利用作业控制语言编写的，用以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），监督程序则通过解释执行“作业控制说明书”中的语句来控制作业运行。典型的卡片作业结构如图 1.3 所示。

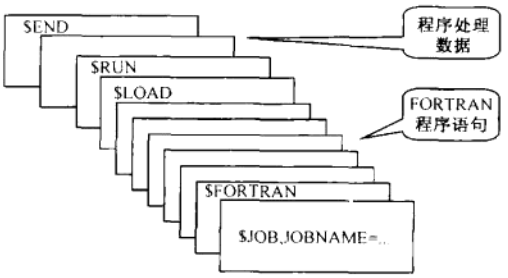


图 1.3　典型的卡片作业结构

这叠卡片中某些卡片表示了作业控制语句，监督程序通过逐条解释、执行该说明书中的作业控制语句自动控制作业运行。$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，并启动编译程序编译后面的源程序。编译结束后，控制返回到监督程序。监督程序解释 $LOAD 语句的结果是通过连接程序把经过编译的程序连接起来，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。

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。批处理作业的控制意图描述在“作业说明书”中。作业控制程序在控制某一道作业运行时，其实质性工作是解释执行“作业说明书”中的语句，实现对作业的控制。从逻辑上看，一个作业由三部分组成：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后，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，可以直接控制和使用系统资源（如主存、外部设备等），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，用户程序则只能执行普通指令。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。由此又引出了“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请求一次特定的系统调用。当处理机执行到用户程序的系统调用指令时，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，控制也随之转入监督程序。监督程序根据用户提供的调用参数进行相应的处理，完成设备 I/O 等功能。处理结束后，监督程序则根据“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界面，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、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，避免了用户在自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据部分主存空间。通常，监督程序占用主存的 $0\sim k$ 单元，$k+1\sim n$ 单元供用户程序占用。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，硬件提供一个界地址寄存器，用以存放系统空间与用户空间的分界地址。当系统处于用户模式时，每访问一次主存，硬件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，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而引起的。例如，用户请求设备 I/O 或请求结束运行。但是，若用户程序执行过程中永不出现系统调用，或者永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”），系统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，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系统处理机的现象。

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）判断输入设备上是否有待输入的作业，如果没有，则等待作业输入。

（2）从设备上输入一道作业。

（3）控制作业运行。

① 取“作业说明书”中的一条语句，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 CPU，开始在用户模式执行该程序。

③ 在用户模式的程序执行过程中，如果发生“中断”事件（如 I/O 中断、系统调用、程序执行错误等），硬件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，控制转步骤①，取下一条“作业说明书”语句执行。

监督程序如同一个系统操作员，它负责批作业的 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，控制使用计算机资源。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），但它与操作系统有许多相似的特征。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 CPU 的利用率。但是，在计算机速度大幅度提高的形势下，用这种方法管理计算机远不能适应需要。首先，在一个 CPU 上运行的程序启动 I/O 操作时 CPU 被迫处于空闲状态或忙等待（Busy_Wait）状态，也就是说，CPU 启动 I/O 操作后在循环判断 I/O 是否完成，而没有做实质性工作，这将导致高速的 CPU 受到慢速设备的牵制，从而使 CPU 无法充分利用。

2. 利用脱机 I/O 改善系统性能

由于作业的 I/O 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，CPU 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 CPU 与外设间速度上的差异。历史上人们曾采用脱机 I/O 技术实现作业 I/O，其系统模型如图 1.4 所示。

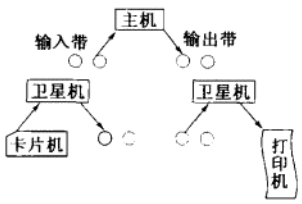


图 1.4 脱机 I/O 系统模型

在采用脱机 I/O 技术的系统中，主机的所有 I/O 操作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上（称为输入带），然后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。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。由于磁带机比慢速 I/O 设备（如卡片机、打印机）的速度快，因而按照这种脱机方式控制作业的 I/O，可以减少作业 I/O 所花的时间，有效地提高 CPU 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 CPU 与 I/O 设备是以串行方式工作的，也就是说，当 CPU 工作时，I/O 设备闲着；当 I/O 设备工作时，CPU 在忙等待，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，从而限制了对计算机的灵活使用。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，主要元件由电子管变成了晶体