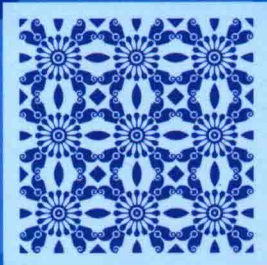


OPERATING SYSTEM CONCEPTS
SECOND EDITION

操作系统原理
第2版

孟庆昌 张志华 等编著



高等院校计算机教材系列

OPERATING SYSTEM CONCEPTS
SECOND EDITION

操作系统原理

第2版

孟庆昌 张志华 等编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

操作系统原理 / 孟庆昌, 张志华等编著. —2 版. —北京: 机械工业出版社, 2017.8
(高等院校计算机教材系列)

ISBN 978-7-111-58128-4

I. 操… II. ①孟… ②张… III. 操作系统 - 高等学校 - 教材 IV. TP316

中国版本图书馆 CIP 数据核字 (2017) 第 243323 号

本书全面而系统地介绍了现代操作系统的基本理论和最新技术, 并以 Linux 系统为实例介绍操作系统的设计与实现。全书共分 7 章: 第 1 章是操作系统概述; 第 2 ~ 6 章分别讲述进程管理、处理机调度、存储管理、文件系统和设备管理; 第 7 章简述现代操作系统的发展和安全管理机制。为强化操作系统课程的实践环节, 附录 A 给出了 8 个实验指导, 附录 B 和 C 分别给出 Linux 常用系统调用、库函数和常用命令, 供教师和学生参考。

本书可作为大学本科及专科计算机相关专业教材或考研参考书, 也可作为计算机工作者的自学用书。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 余 洁

责任校对: 殷 虹

印 刷: 北京瑞德印刷有限公司

版 次: 2017 年 11 月第 2 版第 1 次印刷

开 本: 185mm × 260mm 1/16

印 张: 22

书 号: ISBN 978-7-111-58128-4

定 价: 49.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

前 言

本书在修改之前，课程组征询和归纳了使用本教材的部分院校师生的反馈意见，参考硕士研究生入学考试大纲，并结合实际教学中的体会，研讨了操作系统理论、技术和应用的最新发展。在本次修订中，我们力求能够突出理论基本点，讲清技术发展脉络，导入最新知识应用，并专门增加了有关 Linux 的设计实现技术的内容。近来，“勒索”病毒在网上肆虐，运行 Windows 系统的机器纷纷中招，在全球造成了重大损失和严重恐慌，然而，在运行 Linux 系统的机器上并未出现这种灾难。可见，推广、使用和开发 Linux 系统是正确的选择。

与第 1 版相比，本次修订进行了一系列重要修改，主要包括以下几个方面：

1) 在第 1 章中，1.1.1 节增加了“看待操作系统的进程管理观点”的内容，对操作系统的运行环境做了补充，详细讨论了系统程序和系统调用之间的关系，借助图例进一步阐述了命令行接口和图形用户接口，详细讨论了微内核结构。

2) 在第 2 章中，采用了较规范化的进程定义，详细讨论了进程挂起状态，给出 Linux 中 `task_struct` 结构的简要定义；详细讨论了 Linux 有关进程操作的命令和系统调用使用示例；更严谨地描述了同步和互斥的含义，对信号量的类型做了界定，改写了“哲学家进餐问题”的算法；增加了“使用信号量的几点提示”内容，以答复实际授课中学生对信号量和 P、V 操作解决进程同步等问题时的疑惑，这仅是粗浅体会，希望起到“抛砖引玉”的效果；增加了 2.8 节；修订了对活锁概念的介绍。

3) 在第 3 章中，增加了 3.4 节，增加了关于“高响应比优先法”的例题。

4) 在第 4 章中，4.1 节增加了“用户程序的主要处理过程”方面的内容，将动态重定位的实现过程并入“动态重定位”，补充了“覆盖技术”的相关介绍，添加了空闲分区链的图示；将原 4.9 节调整为 4.5 节；增加了 4 个页面置换算法，即“第二次机会置换法”“时钟置换法”“最少使用置换法”和“页面缓冲算法”；删除了 4.9 节中有关“链接中断处理”的内容；增加了 4.10.3 节。

5) 在第 5 章中，补充了关于“UNIX 文件系统的 i 节点”的内容，增加了 5.7.3 节。

6) 在第 6 章中，增加了 6.1.3 节、6.5.2 节、6.5.3 节，以及补充了 6.5.4 等的内容。

7) 在第 7 章中，扩充了“嵌入式系统”和“分布式系统”方面的内容；增加了 7.2.7 节，以适应当前信息技术最热门应用的潮流。

8) 各章后面增加了有代表性的习题，附录 A 中增加了一个实验指导。另外，对书后给出习题参考答案的做法一直存在争议，褒贬不一。此次修订时我们取消了原书附录中的参考答案，意图是提升学生自主思考、分析、解决问题的能力，培养勤信作风，帮助教师了解学生对相关知识的真实掌握情况。为辅助教师备课，我们将另外提供这部分资料。

9) 最后, 对原书中不妥、不确切、不明了的表述做了修订。

全书共分 7 章:

第 1 章是操作系统概述, 主要介绍操作系统的概念、基本功能、主要特征、在计算机系统中的地位, 以及操作系统的基本类型及主要结构、UNIX 和 Linux 系统的核心结构。

第 2 章是进程管理, 主要介绍进程的概念、进程的状态和组成、进程管理、进程同步与互斥、进程通信、线程和管程、死锁的定义及各种对策。

第 3 章是处理机调度, 主要介绍调度的级别、各级调度的功能和模型、调度性能评价标准及常用调度算法、中断处理和系统调用、shell 基本工作原理。

第 4 章是存储管理, 主要介绍与地址空间有关的基本概念、分区管理、基本的分页技术和分段技术、虚拟存储器、请求分页和请求分段技术、Linux 中的存储管理技术。

第 5 章是文件系统, 主要介绍文件分类、文件系统的功能、文件的逻辑组织和物理组织、文件的目录结构、文件的存储空间管理、文件的可靠性、文件共享和保护、Linux 文件系统。

第 6 章是设备管理, 主要介绍设备管理的有关概念和功能、设备分配技术、I/O 软件构造原则、磁盘调度和管理、Linux 系统设备管理。

第 7 章是操作系统的发展和安全性, 主要介绍操作系统发展的动力、现代操作系统的发展、系统的安全性、系统性能评价。

书后三个附录分别给出了实验指导、Linux 常用系统调用和库函数、Linux 常用命令。

由于各学校课程设置、学时安排及学生程度等方面存在差异, 在应用本书授课时, 可以对内容进行适当取舍。下面列出的理论课学时安排建议是我们多年授课的体会, 仅供参考。

理论课学时安排 (建议)

总学时	课时分配						
	第 1 章	第 2 章	第 3 章	第 4 章	第 5 章	第 6 章	第 7 章
48	4	12	6	12	6	6	2
56	6	14	6	12	7	7	4
64	6	16	6	14	8	8	6

本次修订工作主要由孟庆昌、张志华完成, 参与编写、整理、录入工作的还有刘振英、牛欣源、路旭强、孟欣、马鸣远等。

由于编者水平有限, 时间仓促, 对广大读者的需求尚缺乏广泛深入的了解, 书中难免存在不妥甚至错误之处, 恳切期望广大读者给予批评指正, 并及时反馈用书信息。

作者

2017 年 7 月

于北京信息科技大学

目 录

前言

第 1 章 操作系统概述 1

1.1 操作系统概念 1

1.1.1 什么是操作系统 2

1.1.2 操作系统运行环境 4

1.1.3 系统初启一般过程 7

1.1.4 操作系统的构建目标和地位 8

1.1.5 操作系统提供的服务及其方式 10

1.2 操作系统的功能 12

1.3 操作系统的特征 17

1.4 操作系统的形成和基本类型 17

1.4.1 操作系统的形成和发展 17

1.4.2 操作系统的基本类型 20

1.5 操作系统的主要结构 23

1.5.1 单体结构 24

1.5.2 层次结构 24

1.5.3 虚拟机结构 25

1.5.4 微内核结构 25

1.5.5 客户-服务器结构 26

1.6 UNIX 和 Linux 系统的核心结构 27

1.6.1 UNIX 系统的核心结构 27

1.6.2 Linux 系统的核心结构 29

小结 30

习题 1 31

第 2 章 进程管理 33

2.1 进程概念 34

2.1.1 程序顺序执行及其特征 34

2.1.2 程序并发执行及其特征 34

2.1.3 进程概念的引入和定义 36

2.2 进程状态描述及组织方式 39

2.2.1 进程的状态及其转换 39

2.2.2 进程的组成 41

2.2.3 进程组织方式 43

2.3 进程管理和有关命令 44

2.3.1 进程图和进程管理 44

2.3.2 Linux 进程管理 48

2.3.3 有关进程操作的命令 51

2.3.4 有关进程管理的系统调用 55

2.4 线程概念 58

2.4.1 什么是线程 59

2.4.2 线程的实现方式 61

2.5 进程间的同步与互斥 62

2.5.1 进程间的关系 63

2.5.2 竞争条件和临界区 65

2.5.3 进程同步机制 66

2.5.4 信号量的一般应用 70

2.6 经典进程同步问题 72

2.7 进程通信 77

2.7.1 高级进程通信方式 78

2.7.2 消息缓冲通信 79

2.7.3 信箱通信 80

2.8 Linux 系统的进程通信 82

2.8.1 信号机制 82

2.8.2 管道文件 84

2.8.3 System V IPC 机制 85

2.9 管程 86

2.10 死锁 87

2.10.1 死锁概述 87

2.10.2 死锁的预防 91

2.10.3 死锁的避免 92

2.10.4 死锁的检测与恢复	96
2.10.5 饥饿和活锁	99
小结	100
习题 2	101
第 3 章 处理机调度	105
3.1 调度的作用和级别	106
3.2 作业调度	107
3.2.1 作业状态	107
3.2.2 作业管理和调度	108
3.3 进程调度	110
3.3.1 进程调度的功能和时机	110
3.3.2 两级调度模型	110
3.3.3 三级调度模型	111
3.4 线程调度	111
3.5 调度性能的评价	112
3.5.1 调度策略的选择	113
3.5.2 性能评价标准	113
3.6 常用调度算法	114
3.7 实时调度	121
3.8 Linux 系统中的进程调度	122
3.8.1 Linux 进程调度方式	122
3.8.2 Linux 常用调度命令	123
3.9 中断处理和系统调用	126
3.9.1 中断处理的一般过程	126
3.9.2 系统调用处理	132
3.10 shell 基本工作原理	136
小结	137
习题 3	138
第 4 章 存储管理	140
4.1 地址空间与重定位	141
4.1.1 用户程序的地址空间	141
4.1.2 重定位概念	143
4.1.3 覆盖技术	146
4.1.4 对换技术	147
4.2 分区管理技术	148
4.2.1 固定分区法	149
4.2.2 动态分区法	150
4.2.3 可重定位分区的紧缩	154
4.3 分页技术	155
4.3.1 分页的基本概念	155
4.3.2 分页系统中的地址映射	157
4.3.3 页的共享和保护	159
4.3.4 页表的构造	161
4.4 分段技术	164
4.4.1 分段的基本概念	164
4.4.2 分段系统中的地址映射	166
4.4.3 段的共享和保护	166
4.5 段页式结合系统	168
4.6 虚拟存储管理	169
4.6.1 虚拟存储器的概念	169
4.6.2 虚拟存储器的特征	171
4.7 请求分页技术	171
4.7.1 请求分页的基本思想	171
4.7.2 硬件支持及缺页处理	172
4.7.3 页面置换算法	175
4.8 内存块分配和抖动问题	182
4.8.1 内存块分配	182
4.8.2 抖动问题	184
4.8.3 工作集	185
4.9 请求分段技术	186
4.10 Linux 系统的存储管理技术	187
4.10.1 对换技术	187
4.10.2 请求分页技术	188
4.10.3 Linux 常用内存管理命令和函数	190
小结	195
习题 4	196
第 5 章 文件系统	199
5.1 文件系统概述	199
5.1.1 文件及其分类	200
5.1.2 文件系统的功能	203
5.2 文件的逻辑组织和物理组织	204
5.2.1 文件的逻辑组织	204
5.2.2 用户对文件的存取方法	205
5.2.3 文件的物理组织	207

5.3 目录文件	211	6.4.1 磁盘硬件	267
5.3.1 文件控制块和文件目录	211	6.4.2 磁盘调度算法	268
5.3.2 目录结构	213	6.5 Linux 系统设备管理	270
5.4 文件存储空间的管理	218	6.5.1 Linux 设备管理概述	271
5.5 文件系统的可靠性	221	6.5.2 设备驱动程序的接口	272
5.5.1 坏块管理	221	6.5.3 Linux 系统的缓冲技术	273
5.5.2 文件的备份和恢复	222	6.5.4 块设备管理	274
5.5.3 文件系统的一致性	224	6.5.5 字符设备管理	276
5.6 文件共享和保护	225	6.5.6 可安装模块	276
5.6.1 文件共享	225	小结	277
5.6.2 文件保护	227	习题 6	278
5.7 Linux 文件系统	229	第 7 章 操作系统的发展和安全性	280
5.7.1 一般文件系统的格式	230	7.1 推动操作系统发展的动力	280
5.7.2 虚拟文件系统	232	7.2 现代操作系统的发展	281
5.7.3 Linux Ext2 文件系统	235	7.2.1 个人机操作系统	281
5.7.4 对文件的主要操作	239	7.2.2 网络操作系统	282
小结	242	7.2.3 嵌入式操作系统	284
习题 5	243	7.2.4 多处理器系统	287
第 6 章 设备管理	245	7.2.5 分布式系统	289
6.1 设备管理概述	246	7.2.6 四种多机系统的比较	293
6.1.1 设备分类和标识	246	7.2.7 云计算系统	294
6.1.2 I/O 系统结构	247	7.2.8 操作系统发展展望	297
6.1.3 I/O 系统的控制方式	250	7.3 系统安全性	298
6.1.4 缓冲技术	253	7.3.1 信息安全问题	298
6.1.5 设备管理的功能	256	7.3.2 一般性安全机制	300
6.2 设备分配技术与 SPOOLing 系统	257	7.3.3 保护机制	302
6.2.1 设备分配技术和算法	257	7.4 系统性能评价	307
6.2.2 SPOOLing 系统	259	小结	311
6.3 I/O 软件构造原则	260	习题 7	312
6.3.1 I/O 软件目标	260	附录 A 实验指导	313
6.3.2 设备驱动程序	261	附录 B Linux 常用系统调用和 库函数	327
6.3.3 与设备无关的操作系统 I/O 软件	263	附录 C Linux 常用命令	333
6.3.4 用户空间 I/O 软件	265	参考文献	341
6.3.5 处理输入输出请求的步骤	265		
6.4 磁盘调度和管理	267		

第1章 操作系统概述

学习内容

你知道你所用的计算机上安装的是什么系统吗？它是怎样启动的？为什么要安装它？它是什么类型的操作系统？有什么功能？操作系统又是什么？……下面就介绍这些内容。

为什么要学习操作系统？因为计算机离不开操作系统，它是计算机系统的基本组成部分，是整个系统的基础和核心。操作系统的性能直接影响各行各业的应用。在当今网络时代，它关乎信息安全、产业发展乃至国家安全。学好操作系统是后继课程的需要，是社会应用的需要，是设计、开发具有自主知识产权国产核心软件的需要。

怎样学习操作系统呢？本书围绕着“操作系统是什么、操作系统干什么、操作系统如何干”等内容逐一讲述操作系统的基本概念、基本理论、基本技术以及 UNIX/Linux 系统的基本知识。

本章主要介绍以下主题：

- 什么是操作系统
- 操作系统的地位和特征
- 操作系统的主要功能
- 操作系统的基本类型
- 操作系统的主要结构
- UNIX 和 Linux 系统的核心结构

学习目标

了解：系统初启一般过程，看待操作系统的观点，操作系统的发展历程，操作系统在计算机系统中的地位，UNIX 和 Linux 系统核心结构。

理解：系统调用与系统程序，操作系统的基本类型，分时概念，分时和实时操作系统的特点，操作系统的特征，操作系统的主要结构。

掌握：多道程序设计，操作系统的定义，操作系统的主要功能。

1.1 操作系统概念

通常，一个完整的计算机系统是由硬件和软件两大部分组成的。硬件是指计算机物理装置本身，它是计算机软件运行的基础。从计算机的外观看，它由主机、显示器、键盘和鼠标等几个部分组成；软件是与计算机系统操作有关的计算机程序、过程、规

则以及相关的文档资料的总称。简单地说，软件是计算机执行的程序，如 Windows、Microsoft Office Word、Linux 以及 IE 等都属于软件范畴。在所有软件中，操作系统 (Operating System) 占有特殊的重要地位，它是配置在计算机硬件之上的第一层软件。它控制硬件的工作，管理计算机系统的各种资源，并为系统中各个程序的运行提供服务。众所周知的 Windows、Linux 都是当前最流行的操作系统。

1.1.1 什么是操作系统

大家几乎天天用到计算机，每次开机后都要启动系统，即引导操作系统。你的机器上或是安装了 Windows，或是 Linux，等等。它们有许多相同之处，又有众多差别。那么，什么是操作系统呢？

1. 操作系统的定义及其理解

操作系统是一类软件的总称。虽然操作系统已存在很多年，但至今仍没有一个统一的定义。通常情况下，可以这样定义操作系统：

操作系统是控制和管理计算机系统内各种硬件和软件资源、有效地组织多道程序运行的系统软件（或程序集合），是用户与计算机之间的接口。

怎样理解操作系统的定义呢？我们要注意以下几点：

第一，操作系统是软件，而且是系统软件，也就是说，它由一整套程序组成。例如，UNIX 系统就是一个很大的程序，它由上千个模块组成，有的模块负责内存分配，有的模块实现 CPU 管理，还有的完成读文件工作，等等。程序中还使用了大量的表格、队列等数据结构。

第二，它的基本职能是控制和管理系统内各种资源，有效地组织多道程序的运行。想象一下你编写的程序在计算机上执行的大致过程：程序以文件形式存放在磁盘上，运行之前计算机把它调入内存，然后在 CPU 上运行，产生的结果在屏幕上显示出来。这些工作都由操作系统完成。

第三，它提供众多服务，方便用户使用，扩充硬件功能。例如，用户可以使用操作系统提供的上百条命令或者图形界面完成对文件、输入输出、程序运行等许多方面的控制、管理工作；可以在一台机器上完成多项任务；甚至可以多个人同时使用一台机器。

2. 如何看待操作系统

出现“操作系统难以准确定义”这个问题，一方面由于操作系统实现两项相对独立的功能——扩展机器和管理资源，另一方面取决于从什么角度来看待操作系统——用户观点还是系统观点，以及静态角度还是动态角度。

(1) 作为扩展机器的操作系统

裸机（仅有硬件的计算机）提供的机器语言（即“0”“1”码）难记、难用，又难懂。在裸机之上安装操作系统之后，就把硬件细节与程序员隔离开。用户可以使用系统提供的各种命令，直接打开文件、读写文件、更改目录、将文件复制到 U 盘上，等等。在做

这些事情时，我们只关心自己要实现的目标，并未考虑硬件如何动作，从而隐藏了底层硬件的特性，实现简单的、高度抽象的处理。

抽象是管理复杂事物的一个关键。良好的抽象可以把繁杂的、难于管理的任务划分为可以管理的两部分，即有关对象的抽象定义和实现，以及应用这些抽象解决相关问题。可见，操作系统的实际客户是应用程序（当然是通过应用程序员），它们直接与操作系统及其抽象打交道。而终端用户是与用户接口所提供的抽象（如 shell 命令行或图形接口）打交道。

经过操作系统的加工，呈现在用户面前的机器是功能更强、使用更方便的机器。通常在裸机之上覆盖各种软件，从而形成功能更强的机器，称为扩展机器或虚拟机。

这种功能扩展可以重叠。在裸机之上覆盖一层软件后，得到第一层扩展；在此基础上再加一层软件，就得到第二层扩展，以此类推。

（2）作为资源管理器的操作系统

上述把操作系统看作向应用程序提供基本抽象的概念是一种自顶向下的观点。另外一种观点是自底向上的观点，它考察操作系统如何管理一个复杂系统的各个部分。大家知道，现代操作系统允许同时运行多道程序。所以，操作系统的功能就是管理系统中的硬件资源和数据、程序等软件资源，控制、协调各个程序对这些资源的利用，尽可能地充分发挥各种资源的作用。这就涉及资源共享问题，即时间复用（如 CPU 分时）和空间复用（如内存和磁盘的共用）。

因此，作为资源管理者，操作系统主要完成以下工作：

- 1) 监视各种资源，随时记录它们的状态。
- 2) 实施某种策略以决定谁获得资源，何时获得，获得多少。
- 3) 分配资源供需求者使用。
- 4) 回收资源，以便再分配。

总之，操作系统确实是计算机系统的资源管理器。当今看待操作系统作用的众多观点中，这种观点仍占主导地位。

（3）看待操作系统的用户观点和系统观点

从计算机用户的角度来看，操作系统处于用户与计算机硬件系统之间，有助于用户使用计算机系统的接口和各种资源。因此，操作系统应当使用方便、功能强、效率高、使用安全可靠、易于安装和维护，等等，当然价格应该便宜。这些看法反映了普通用户对操作系统的需求和期望，是从系统外部看待操作系统的作用。

另一种观点是系统观点，即从系统内部实现的角度来看待操作系统的作用。操作系统是硬件之上的第一层软件，它要管理计算机系统中各种硬件资源和软件资源的分配问题，如 CPU 时间、内存空间、文件存储空间、I/O 设备，等等，要解决大量对资源请求的冲突问题，决定把资源分配给谁、何时分配、分配多少等，使得资源的利用高效而且公平。这样，操作系统就是资源分配者。

另外，操作系统要对 I/O 设备和用户程序加以控制，保证设备正常工作，防止非法操作，及时诊断设备故障等。从这个意义上讲，操作系统就是控制程序。

(4) 看待操作系统的进程管理观点

上述两种观点是从静态的角度看待操作系统的。其实操作系统的内部活动是一个动态过程，很多并发执行的程序要占用系统资源，在活动过程中会直接或间接地产生相互制约的关系，从而引入“进程”概念。简单地说，进程就是程序在并发环境中的执行过程（详见 2.1.3 节）。进程是有生命周期的，从创建到终止要经历不同过程，处于不同状态；进程是有族系关系的，父进程创建子进程，子进程还可再创建子进程；进程是被调度、独立运行的单位，在其活动过程中会与其他进程发生联系或冲突，等等。用进程观点来研究操作系统就会明白其内部众多“生命体”是如何生存、活动、联系的。

还可以从其他角度来看待操作系统，这里不一一列举。

1.1.2 操作系统运行环境

操作系统是运行在计算机硬件之上的第一层软件，它控制和管理系统内各种硬件与软件资源。操作系统与承载它的计算机硬件之间有着密切联系。下面简要介绍操作系统运行的硬件环境。

1. 现代计算机体系结构

现代计算机体系结构基本上仍沿用 Von Neumann (冯·诺依曼) 体系结构，采用存储程序工作原理，即：把计算过程描述为由许多条命令按一定顺序组成的程序，然后把程序和所需的数据一起输入计算机存储器中保存起来，工作时控制器执行程序，控制计算机自动连续地进行运算。

大家知道，现代通用计算机系统由 CPU、内存和若干 I/O 设备组成。它们经由系统总线连接在一起，实现彼此通信，如图 1-1 所示。从功能上讲，计算机系统由五大功能部件组成，即运算器、控制器、存储器、输入设备和输出设备。这五大功能部件相互配合，协同工作。其中，运算器和控制器集成在一片或几片大规模或超大规模集成电路中，称之为中央处理器 (CPU)。

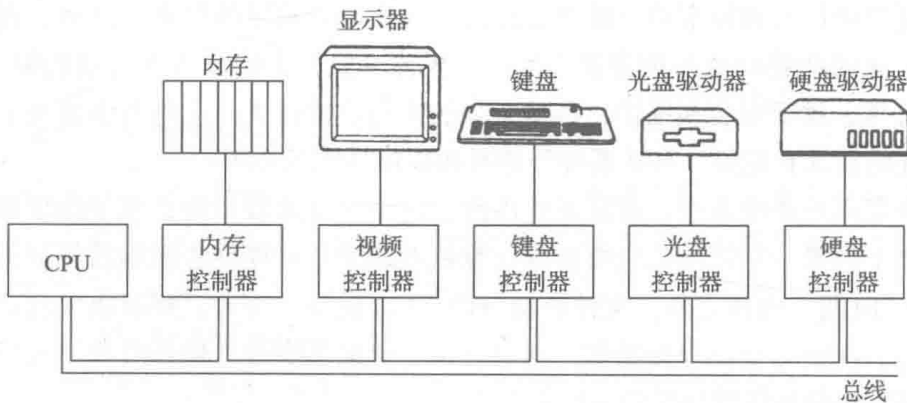


图 1-1 现代计算机硬件结构

请注意，图 1-1 中示出的控制器是设备控制器。每个设备控制器负责对特定类型的设备进行控制和管理，如硬盘控制器用来控制硬盘驱动器，视频控制器用来控制显示器，等等。CPU 和设备控制器可以并行工作，它们都要存取内存中的指令或数据。为保障对共享内存的有序存取，内存控制器对这些访问实施同步管理。

2. 与操作系统相关的几种主要寄存器

处理器中包含若干寄存器（因具体型号而异），它们交换数据的速度比内存更快，体积更小，但是价格也比内存单元贵。这些寄存器的功能分为两类：用户可编程寄存器及控制与状态寄存器。应指出，上面对寄存器的分类并不是很明确的，如在有些机器中程序计数器是用户可编程的，而在多数机器上是不允许的。所以，在不同的系统中，这些寄存器的功能和作用不尽相同。

(1) 用户可编程寄存器

程序员可以利用机器语言或汇编语言来对用户可编程寄存器进行操作，以减少对内存的访问。程序员也可以使用像 C 语言这样的高级语言对它们进行操作（如指定某些变量的类型为 register）。典型的用户可编程寄存器包括数据寄存器（data register）、地址寄存器（address register）、条件码寄存器（condition code register）。

数据寄存器又称通用寄存器，用来保存关键变量和中间结果。本质上是通用的，凡对数据进行操作的任何机器指令都可以访问它们。但在实际使用时往往受到限制，如指定它们仅用于浮点运算等。

地址寄存器一般用来存放数据和指令的内存地址或者计算完整地址时所用的入口地址。地址寄存器也往往用于特定的寻址方式，如作为索引寄存器、段指针、堆栈指针等。

条件码寄存器也称作标志寄存器，处理器硬件依据运算结果设置其二进制位，如算术运算的结果可能为正、为负、等于 0 或者溢出。不管运算结果本身是存放在一个寄存器中还是内存中，都要设置条件码寄存器。随后测试它的值，决定运行程序的哪个分支。

(2) 控制与状态寄存器

控制与状态寄存器可被处理器用来控制自身的操作，或者被有特权的操作系统例程用来控制相关程序的执行。处理器中有很多控制与状态寄存器，用来控制处理器的操作。它们往往是专用的。当然，在不同的机器上它们是有差异的。

最主要的控制与状态寄存器包括程序计数器（Program Counter, PC）、指令寄存器（Instruction Register, IR）、程序状态字（Program Status Word, PSW）寄存器。程序计数器中存放有要取出的指令的内存地址，在指令取出后，它就指向下一条；指令寄存器中存放有待执行的指令；程序状态字寄存器包含条件码位、CPU 优先级、运行模式（用户态或核心态），以及各种其他控制位。在系统调用和 I/O 中，PSW 的作用很重要。

所有的处理器都有若干寄存器，它们与操作系统有着非常直接和密切的关系，操作系统必须知晓所有的寄存器。

3. 特权指令和 CPU 运行模式

指令是控制计算机执行某种操作（如加、减、传送、转移等）的命令。一台计算机所能执行的全部指令的集合称作指令系统或指令集。不同型号的 CPU 有不同的指令集，也就是说，指令集与计算机系统密切相关，没有可移植性。

(1) 特权指令

特权指令是指计算机指令集中一类具有特殊权限的指令，它们只用于操作系统或其他系统软件，一般普通用户不能直接使用。它主要用于系统资源的分配和管理，包括改变系统工作方式，检测用户的访问权限，控制 I/O 设备动作，访问程序状态，修改虚拟存储器管理的段表、页表，完成任务的创建和切换，等等。普通用户能使用的指令是非特权指令，它们是指令集中除特权指令之外的指令，如算术运算指令、访管指令等。

(2) CPU 运行模式

为了使操作系统程序（特别是其内核部分）免受用户程序的干扰和损害，在多数 CPU 的设计中都提供两种运行模式：核心态（Kernel Mode，又称内核态、系统态、管态）和用户态（User Mode，又称目态）。

核心态指特权状态，是操作系统内核所运行的模式。此时，CPU 具有较高的权限，可以执行机器指令集中的全部指令，包括特权指令。这样，在核心态下，操作系统就具有对所有硬件的完全访问权，其代码可以不受限制地对系统存储、外部设备进行访问，从而实施有效地控制和管理。

用户态指非特权状态，是普通用户程序所运行的模式。当用户程序在机器上运行时，CPU 处于用户态，其权限较低，只能执行非特权指令，即执行的代码被硬件限定，不能进行某些操作（如写入其他进程的存储空间）。

在特定条件下，这两种模式可以相互转换：在用户程序运行过程中，当发生中断或者系统调用时，CPU 状态就转为核心态，这样就可以执行操作系统的程序了。而当中断或者系统调用的事件处理完成后，通常就转回用户态，以便继续完成用户的任务。

4. 中断和异常

现代计算机系统的一个重要特性就是允许多个进程同时在系统中活动，即并发。实施并发的基础是由硬件和软件结合而成的中断机制。中断对于操作系统非常重要，许多人称操作系统是由“中断驱动”的。

所谓中断是指 CPU 对系统发生的某个事件做出的一种反应，它使 CPU 暂停正在执行的程序，保留现场后自动执行相应的处理程序，处理该事件后，如被中断进程的优先级最高，则返回断点继续执行被“中断”的程序。

按中断事件来源，中断分为两类：

- 1) 中断。它是由 CPU 以外的事件引起的，如 I/O 中断、时钟中断、控制台中断等。
- 2) 异常（Exception）。它是来自 CPU 内部的事件或程序执行中的事件引起的过程，如 CPU 故障、程序故障等。

有关这部分的详细内容，请参见 3.9 节。

1.1.3 系统初启一般过程

当打开计算机电源以后,计算机就开始初启过程,即引导操作系统。系统初启过程的细节与所用计算机的体系结构有关,但对所有机器来说,初启的目的是相同的:将操作系统的副本读入内存,建立正常的运行环境。对于 Intel i386 系列来说,初启过程分为硬件检测、加载引导程序、初始化内核和实现用户登录。

1. 硬件检测

当计算机加电启动时,首先 CPU 进入实模式,开始执行 ROM-BIOS 起始位置的代码。BIOS 执行加电自检程序 (POST),完成硬件启动,然后对系统中配置的硬件(如内存、硬盘及其他设备)进行诊断检测,确认各自在系统中存在,并且处于正常状态。自检工作要经历约 2~3 分钟。自检完成后,按照预先在系统 CMOS 中设置的启动顺序,ROM-BIOS 搜索 CD-ROM、硬盘或者软盘等设备的驱动器,读入系统引导区(通常是磁盘上的第一个扇区)的程序,并将系统控制权交给引导装入程序。

2. 加载引导程序

整个磁盘的第一个扇区是引导扇区。加电后就从此处“引导”,所以它称为“主引导记录”(MBR)。MBR 中存有磁盘分区的数据和一段简短的程序,该程序并不直接引导操作系统,但它能根据盘区划分的信息找到“活动”分区,然后从活动分区中将引导程序读入内存;运行系统引导程序,它从硬盘中读入其他几个更为复杂的程序,由后者加载操作系统的内核。

内核加载完毕后,系统跳转到 setup 程序,并在实模式下运行,该程序设置系统参数(包括内存、磁盘等,由 BIOS 返回)、检测和设置显示器与显示模式等。最后进入保护模式,并转到内核映像的开头,执行内核初始化。

3. 初始化内核

系统初始化过程可以分为三个阶段。

第一个阶段主要是 CPU 本身初始化,如设置内核页表、启动页面映射机制、建立系统的第一个进程、初始化内核的全局变量和静态变量、设置中断向量表的初始状态等。

第二个阶段主要是系统中一些基础设施的初始化,如设置内存边界、初始化内存页面、设置各种处理程序入口地址、定义系统中最大进程数目、创建 init 内核线程等。

第三个阶段是对上层部分初始化,如初始化外部设备、加载驱动程序、创建核心线程、初始化文件系统并加载它等。

内核初始化工作完成后,就由初始化进程完成系统运行的设置工作,如设置操作系统启动时默认的执行级别、激活交换分区、检查磁盘、建立用户工作环境、显示登录界面及提示信息等。

4. 实现用户登录

在用户态初始化阶段,init 程序在每个 tty 端口上创建一个进程 login,用来支持用

户登录。login 进程接收用户输入的账号和密码，予以验证。合法用户通过验证后就可以进入系统，使用 shell 交互地执行用户命令，或者在桌面环境上操作。

1.1.4 操作系统的构建目标和地位

1. 操作系统的构建目标

操作系统是控制应用程序执行的程序，扮演计算机用户和计算机硬件之间的接口。设计操作系统时人们可以提出形形色色的要求，归纳起来，构建操作系统的目标主要有以下几点。

(1) 高效性

计算机系统中的所有软硬件资源都在操作系统的统一控制、管理下，通过合理的调度和分派，这些资源得到有效的利用，从而在有限的时间内完成更多的任务。

(2) 方便性

操作系统通过对外提供的接口，大大方便了用户的使用。例如，程序员可以在程序中利用系统调用直接对磁盘上的文件进行读写，终端用户可以通过输入命令或者点击鼠标来操纵计算机的动作。

(3) 安全性

当今社会处于信息时代，人们把大量的信息存放在计算机系统中，特别是随着网络技术的普及与应用，这既为信息交流带来了极大的方便，又产生了日趋严重的信息安全问题。操作系统应保护信息不被未经授权人员访问。

(4) 鲁棒性

鲁棒性 (robustness) 就是系统的强健性。众所周知，使用计算机的用户形形色色，在利用操作系统时会出现各种情况，如输入错误、磁盘故障、网络过载或有意攻击等，操作系统应在这些情况下做到不死机、不崩溃，这就是操作系统的鲁棒性。

(5) 移植性

计算机硬件平台千差万别，而操作系统的开发环境是有限的。当操作系统从一种硬件平台移植到另一种平台时，所做的修改应尽量少，而且要容易实施。

2. 操作系统的地位

如上所述，计算机系统是由硬件和软件组成的。软件“裹”在硬件之上。硬件是软件建立与活动的基础，而软件对硬件进行管理和功能扩充。没有硬件，就失去了计算机系统的物理基础，软件也就无法存在。反过来，若只有硬件而没有软件，则硬件就失去了“灵魂”，很难使用，没有活力，也就没有多大应用价值。硬件与软件有机地结合在一起，相辅相成，才使得计算机技术飞速发展，并在当今信息时代占据举足轻重的地位。

(1) 软件分类

按照所起的作用和需要的运行环境，软件通常可分为三大类，即系统软件、应用软件和支撑软件。

系统软件包括操作系统（如 Windows、Linux 等），编译程序（如 C/C++、Java 语言编译程序等），汇编程序（如 Intel 8080、8086 汇编语言等），连接装配程序（如 Loader）、数据库管理系统（如 SQL Server、Oracle 等），网络软件（如 IE、NetMeter、瑞星杀毒软件）等，这些软件对计算机系统的资源进行控制、管理，并为用户的使用和其他程序的运行提供服务。它们为计算机应用提供最基本的功能和共性服务，并不针对某一特定应用领域。

应用软件是为解决某一类应用需要或某个特定问题而设计的程序，包括图形软件（如 Photoshop、Flash 等），财务软件（如用友、金蝶等财务软件），软件包（如 All-in-One、RPM）等。与系统软件恰好相反，不同的应用软件根据用户和所服务的领域提供不同的功能。这是范围很广的一类软件。

支撑软件是辅助软件技术人员从事软件开发工作的软件，包括各种开发工具（如 JBuilder、Eclipse 等），测试工具（如 IBM Rational Robot、Microsoft Web Application Stress Tool 等）等，所以又称为工具软件，借以提高软件生产率，改善软件产品质量。

（2）操作系统的地位

计算机系统中硬件和软件以及各类软件之间是按层次结构组织的，如图 1-2 所示。

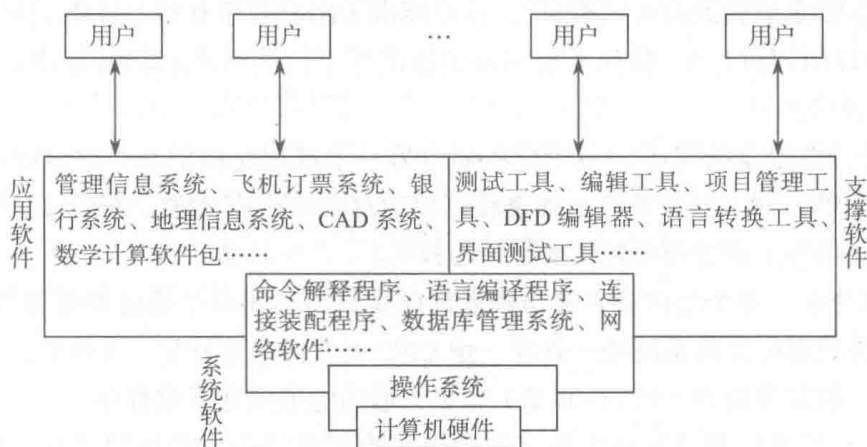


图 1-2 计算机系统的层次关系

由图 1-2 可以看出，操作系统是裸机之上的第一层软件，与硬件关系尤为密切。它不但对硬件资源直接实施控制、管理，而且其很多功能的完成是与硬件动作配合实现的，如中断系统。操作系统的运行需要有良好的硬件环境。这种硬件配置环境往往称作硬件平台。

操作系统是整个计算机系统的控制管理中心，其他所有软件都建立在操作系统之上。操作系统对它们既具有支配权力，又为其运行建造必备环境。因此，在裸机之上每加一层软件后，用户看到的就是一台功能更强的机器，通常把经过软件扩充功能后的机器称为“虚拟机”。在裸机上安装了操作系统后，就为其他软件 and 用户提供了工作环境，往往把这种工作环境称作软件平台。