

 高等教育规划教材

# Linux 操作系统 原理与应用

赵国生 王健 等编著



免费提供电子教案

下载网址 <http://www.cmpedu.com>



机械工业出版社  
CHINA MACHINE PRESS

高等教育规划教材

# Linux 操作系统原理与应用

赵国生 王健 等编著



机械工业出版社

本书以 Red Hat Enterprise Linux 6.2 为平台,介绍了 Linux 操作系统的基本原理以及应用实践,全面讲解了系统的基本概念和操作,以及系统在进程、存储、设备、文件等方面的运行原理,之后,对系统管理与网络配置管理进行了详解,结合实际操作步骤及完整的项目案例说明了在 Linux 平台下服务器的配置与应用,并配以系统安全的介绍,帮助读者完成一个层次递进、由浅入深的学习过程。

本书根据知识体系结构和读者特点的不同,在内容编写上遵循从理论到实践的过程,在基本理论基础,配以具体案例,加深对理论知识的理解。本书具有系统全面,结构递进,重点突出,操作性好,实用性强,语言简练流畅等特点。本书配有光盘,包括源代码、服务器配置等内容。

本书适合大中专院校的学生,可以作为计算机、通信等相关专业本科、研究生操作系统理论及应用课程的授课教材,也可作为相关专业技术人员的参考用书。

本书配套授课电子课件,需要的教师可登录 [www.cmpedu.com](http://www.cmpedu.com) 免费注册,审核通过后下载,或联系编辑索取(QQ: 2850823885,电话: 010 - 88379739)。

## 图书在版编目(CIP)数据

Linux 操作系统原理与应用 / 赵国生等编著. —北京: 机械工业出版社, 2015. 9

高等教育规划教材

ISBN 978-7-111-52080-1

I. ①L… II. ①赵… III. ①Linux 操作系统 - 高等学校 - 教材  
IV. ①TP316. 89

中国版本图书馆 CIP 数据核字 (2015) 第 266636 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 郝建伟 责任编辑: 郝建伟 张 恒

责任校对: 张艳霞 责任印制: 李 洋

三河市宏达印刷有限公司印刷

2016 年 2 月第 1 版 · 第 1 次

184mm × 260mm · 21.75 印张 · 534 千字

0001 - 3000 册

标准书号: ISBN 978-7-111-52080-1

定价: 49.90 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

服务咨询热线: (010) 88379833

读者购书热线: (010) 88379649

封面无防伪标均为盗版

网络服务

机工官网: [www.cmpbook.com](http://www.cmpbook.com)

机工官博: [weibo.com/cmp1952](http://weibo.com/cmp1952)

教育服务网: [www.cmpedu.com](http://www.cmpedu.com)

金书网: [www.golden-book.com](http://www.golden-book.com)

# 出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才,是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。

2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。

3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。

4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

# 前 言

Linux 是一种自由和开放源码的类 UNIX 操作系统，可安装在各种计算机硬件设备中，并可应用于系统管理和维护、系统开发、语言开发及嵌入式软件开发等领域。Linux 存在着许多不同的版本，但它们都使用了 Linux 内核。现阶段 Linux 平台的专业技术人员缺口很大，具有此方面专业技能的人员相对较少，但此领域发展方向很被看好。

本书从初级读者入手，按照知识的体系结构和读者的特点，逐步增加知识点。本书可以引导读者快速掌握 Linux 操作系统的基本原理，进而对 Linux 服务器的配置应用加以实现。教学内容设置由浅入深，同时结合实际操作步骤以及完整的案例项目，并附有示例代码，重点突出，注重理论与实践的结合，以帮助读者快速提高 Linux 的知识水平。

## 基本内容

在内容编写上，本书以 Red Hat Enterprise Linux 6.2 为平台，在讲述系统的基本概念、操作、原理等基础内容之后，对系统、网络及服务器的配置进行全面讲解，并配以系统安全的介绍，涵盖了 Linux 操作系统从初学到进阶的所有主要内容。

全书分为三大部分共 11 章，各章具体内容如下。

- 第 1 章：概括地介绍了 Linux 操作系统，包括操作系统的功能及分类、Linux 操作系统的发展历史及背景、Linux 操作系统的特点、Linux 的版本等。
- 第 2 章：主要讲解了 Linux 系统的安装与基本配置，包括安装前的准备、安装的步骤以及在 VMware 虚拟机下安装 Linux 系统等。
- 第 3 章：主要从整体上讲解了 Linux 系统的基本操作，包括 Linux 系统的基本操作、Linux 命令、Vi 编辑器的使用等。
- 第 4 章：主要讲解了 Linux 系统的进程管理，包括 Linux 系统中进程的概念、进程的组织方式、Linux 进程调度、进程间通信、Linux 线程等。
- 第 5 章：主要讲解了 Linux 系统的内存管理，包括内存管理概念与技术、存储管理方案、虚拟存储器、Linux 系统的存储管理机制等。
- 第 6 章：主要讲解了 Linux 系统的设备管理，包括设备管理的概念与相关技术、I/O 控制方式、设备的分配及处理、Linux 的设备管理机制等。
- 第 7 章：主要讲解了 Linux 系统的文件管理，包括文件的文件系统、逻辑结构、Linux 文件系统、虚拟文件系统、ext3 文件系统、文件系统的管理、文件的打开与读写等。
- 第 8 章：主要讲解了 Linux 操作系统接口与作业管理，包括 Shell 命令接口、X 图形界面接口、Linux 系统调用接口、Linux 中的用户接口与作业管理等。
- 第 9 章：主要讲解了 Linux 系统管理，包括系统管理概述、用户管理、系统备份与监控、软件安装等。
- 第 10 章：主要讲解了 Linux 系统的网络配置与管理，包括网络配置的相关概念、Linux 网络配置、Samba 服务器的配置与应用、DHCP 服务器的配置与应用、DNS 服务器的配置与应用等。
- 第 11 章：概括地介绍了 Linux 系统安全，包括操作系统安全性概述及安全机制、Linux



系统的安全设置、Linux 系统的防火墙管理等。

### 主要特点

本书作者多年来一直从事 Linux 相关课程的讲授及理论研究工作，并在多个项目中对 Linux 内核、系统安全等内容进行了深入研究，有着丰富的教学实践和编著经验。

本书采用最新版的 Red Hat Enterprise Linux 6.2 作为学习平台，在整体内容编排上遵循从理论到实践的过程，采用梯度层次化结构由浅入深地系统介绍 Linux 操作系统的原理及应用。每章会有“本章小结”，在学习本章后对所学的内容进行梳理，达到知识的强化作用。对有实践操作要求的章节，配有详实完整的案例，以加深对 Linux 操作系统理论的理解，实现理论知识的实践应用，提高教学效果，能够使读者快速、真正地掌握 Linux 操作系统。

具体地讲，本书具有以下鲜明的特点。

- 从零开始，轻松入门。
- 图解案例，清晰直观。
- 图文并茂，操作简单。
- 实例引导，专业经典。
- 学以致用，注重实践。

### 读者对象

- 学习 Linux 的初级读者。
- 具有一定 Linux 基础知识，希望进一步深入掌握 Linux 系统配置与管理的中级读者。
- 大中专院校计算机相关专业的学生。
- Linux 平台的专业技术人员。

### 配套资源

为了方便读者学习，本书配有多媒体教学光盘，其中包含了本书多媒体教学课件及服务器配置步骤演示等。

同时，本书得到以下项目支持：国家自然科学基金项目“基于认知循环的任务关键系统可生存性自主增长模型与方法（61403109）”，国家自然科学基金项目“可生存系统的自主认知模式研究”（61202458），高等学校博士点专项基金项目“任务关键系统可信性增强的自律机理研究”（20112303120007）。

本书主要由哈尔滨师范大学赵国生和哈尔滨理工大学王健编写，宋一兵主审。赵国生老师编写第 1~6 章的内容，王健老师编写 7~11 章的内容。参与编写的还有管殿柱、王献红、李文秋、赵景波、谈世哲、曹立文、初航。

由于时间仓促，书中难免有疏漏之处，请读者批评指正，并提出宝贵意见。

编者

# 目 录

出版说明

前言

第 1 章 Linux 操作系统概述 .....	1	2.5 思考与练习 .....	47
1.1 认识操作系统 .....	1	第 3 章 Linux 操作基础 .....	48
1.1.1 操作系统的诞生 .....	1	3.1 Linux 基本操作 .....	48
1.1.2 操作系统的目的及作用 .....	4	3.1.1 图形界面登录 .....	48
1.1.3 操作系统的主要功能 .....	6	3.1.2 修改密码 .....	49
1.1.4 操作系统的分类 .....	8	3.1.3 Linux 运行级别 .....	50
1.2 Linux 概述 .....	11	3.1.4 系统启动过程 .....	52
1.2.1 Linux 成长的历史背景 .....	11	3.1.5 注销系统与关机 .....	54
1.2.2 Linux 的特点 .....	13	3.2 Linux 命令 .....	56
1.2.3 GNU 与 Linux .....	15	3.2.1 命令的格式 .....	56
1.2.4 Linux 的版本 .....	16	3.2.2 命令的启动与退出 .....	57
1.2.5 Linux 的应用与发展 .....	19	3.2.3 命令的分类 .....	58
1.3 本章小结 .....	20	3.2.4 基本命令 .....	60
1.4 思考与练习 .....	20	3.2.5 命令行帮助 .....	72
第 2 章 Linux 的安装与配置 .....	21	3.3 Vi 编辑器的使用 .....	77
2.1 Linux 的安装准备 .....	21	3.3.1 认识 Linux 的文本编辑器 .....	77
2.1.1 获取 Linux 的安装程序 .....	21	3.3.2 Vi 编辑器的启动和退出 .....	77
2.1.2 明确系统硬件信息 .....	22	3.3.3 Vi 编辑器的 3 种工作模式 .....	78
2.1.3 选择安装方式 .....	22	3.3.4 命令模式操作命令 .....	79
2.1.4 硬盘的组织结构 .....	23	3.3.5 输入模式操作命令 .....	81
2.1.5 Linux 分区方案 .....	24	3.3.6 末行模式下的基本操作 .....	81
2.2 Linux 的安装 .....	24	3.3.7 Shell 切换 .....	82
2.2.1 Linux 的安装步骤 .....	24	3.4 本章小结 .....	83
2.2.2 Linux 首次运行的设置步骤 .....	33	3.5 思考与练习 .....	84
2.3 VMware 虚拟机下安装 .....		第 4 章 进程管理 .....	85
Linux 系统 .....	36	4.1 进程概述 .....	85
2.3.1 VMware 简介 .....	36	4.1.1 进程的概念 .....	85
2.3.2 VMware Workstation 网络的 .....		4.1.2 程序和进程 .....	86
工作模式 .....	37	4.1.3 进程的结构 .....	87
2.3.3 VMware Workstation 的下载 .....		4.1.4 进程实例 .....	88
和安装 .....	39	4.1.5 Linux 中的进程 .....	89
2.3.4 VMware 虚拟机下安装运行 .....		4.2 进程控制块 .....	89
Linux .....	42	4.2.1 进程状态 .....	89
2.4 本章小结 .....	47		

4.2.2	进程标识符	91	5.3.1	分区存储管理	120
4.2.3	进程之间的关系	91	5.3.2	分页存储管理	126
4.2.4	进程控制块的存放	92	5.3.3	分段存储管理	128
4.3	进程的组织方式	93	5.3.4	段页式存储管理	129
4.3.1	散列表	93	5.4	虚拟存储器	130
4.3.2	双向循环链表	94	5.4.1	虚拟存储的概念	130
4.3.3	可运行队列	95	5.4.2	请求分页存储管理	131
4.3.4	等待队列	96	5.4.3	请求分段存储管理	132
4.4	进程的互斥与同步	96	5.5	Linux 的存储管理	132
4.4.1	互斥的定义	97	5.5.1	Linux 存储器管理概述	133
4.4.2	同步的定义	97	5.5.2	Linux 的分页管理机制	133
4.4.3	信号量机制	98	5.5.3	Linux 物理内存空间的管理	137
4.4.4	用 P、V 操作实现进程的互斥	99	5.5.4	内核态内存的申请与释放	140
4.4.5	用 P、V 操作实现进程的同步	100	5.5.5	用户态内存的申请与释放	141
4.4.6	死锁	101	5.5.6	存储管理系统的缓冲机制	141
4.5	进程调度	102	5.6	Linux 地址映射实例	142
4.5.1	进程调度的基本原理	102	5.7	本章小结	146
4.5.2	Linux 进程调度	103	5.8	思考与练习	146
4.5.3	调度策略	104	第 6 章	设备管理	147
4.5.4	调度函数	106	6.1	设备管理概述	147
4.6	进程间通信	107	6.1.1	I/O 系统的组成	147
4.6.1	进程通信的方式	107	6.1.2	设备的分类	148
4.6.2	Linux 信号通信原理	108	6.1.3	设备管理的功能	149
4.6.3	Linux 管道通信原理	109	6.2	设备管理的相关技术	149
4.7	线程	110	6.2.1	中断技术	149
4.7.1	线程的概念	110	6.2.2	缓冲技术	150
4.7.2	线程与进程的比较	110	6.2.3	DMA 技术	153
4.7.3	Linux 中的线程	111	6.3	I/O 控制方式	154
4.7.4	线程的实现	112	6.3.1	程序 I/O 方式	154
4.7.5	线程的状态及转换	113	6.3.2	中断方式	155
4.8	本章小结	115	6.3.3	DMA 方式	155
4.9	思考与练习	115	6.3.4	通道方式	156
第 5 章	存储管理	116	6.4	设备的分配	157
5.1	存储管理概述	116	6.4.1	设备分配策略	158
5.1.1	存储管理的概念	116	6.4.2	设备分配程序	159
5.1.2	存储管理的功能	116	6.4.3	SPOOLing 技术	160
5.2	覆盖和交换技术	119	6.5	设备的处理	161
5.2.1	覆盖技术	119	6.5.1	设备处理程序的功能与 处理方式	161
5.2.2	交换技术	120	6.5.2	设备处理程序的处理过程	163
5.3	存储管理方案	120			



6.6	Linux 设备管理	164	7.7.1	打开文件	197
6.6.1	Linux 设备的分类	164	7.7.2	读/写文件	198
6.6.2	Linux 的 I/O 控制	165	7.8	本章小结	199
6.6.3	Linux 的设备文件	168	7.9	思考与练习	199
6.6.4	Linux 设备驱动	168	<b>第8章</b>	<b>操作系统接口及作业管理</b>	<b>200</b>
6.6.5	Linux 的设备管理命令	169	8.1	操作系统接口概述	200
6.7	本章小结	171	8.1.1	操作系统的接口	200
6.8	思考与练习	172	8.1.2	Linux 系统的接口	202
<b>第7章</b>	<b>文件管理</b>	<b>173</b>	8.2	Shell 命令接口	204
7.1	文件与文件系统	173	8.2.1	认识 Shell	204
7.1.1	文件、记录和数据项	173	8.2.2	Shell 的功能及版本	204
7.1.2	文件类型及文件系统模型	174	8.2.3	Shell 的工作流程与原理	205
7.1.3	文件操作	176	8.3	X 图形界面接口	208
7.1.4	文件的存取方式	176	8.3.1	X - Window 系统	208
7.2	文件的逻辑结构	177	8.3.2	X 系统的工作原理	209
7.2.1	文件逻辑结构类型	177	8.3.3	X 系统的启动与停止	210
7.2.2	顺序文件及索引文件	178	8.3.4	Linux 桌面系统	210
7.2.3	顺序索引文件	179	8.4	Linux 系统调用接口	211
7.2.4	直接文件和散列文件	180	8.4.1	系统调用接口概述	211
7.3	Linux 文件系统	180	8.4.2	系统调用接口的组成	213
7.3.1	Linux 文件系统的基本概念	180	8.4.3	系统调用过程	213
7.3.2	Linux 文件结构及特点	181	8.5	作业管理概述	215
7.3.3	Linux 文件类型和属性	181	8.5.1	作业及其类型	215
7.3.4	Linux 文件系统的组织方式	182	8.5.2	作业的状态及其转换	215
7.3.5	文件访问权限	183	8.5.3	作业控制级的接口	216
7.4	虚拟文件系统	184	8.6	作业调度	217
7.4.1	虚拟文件系统的引入	184	8.6.1	作业调度应考虑的因素	217
7.4.2	VFS 中的数据结构	185	8.6.2	作业调度算法	218
7.4.3	VFS 超级块数据结构	188	8.7	Linux 中的用户接口与 系统调用	220
7.4.4	VFS 的索引结点	189	8.8	本章小结	222
7.5	ext3 文件系统	189	8.9	思考与练习	222
7.5.1	ext3 文件的结构	191	<b>第9章</b>	<b>系统管理</b>	<b>223</b>
7.5.2	ext3 文件系统的格式	191	9.1	系统管理概述	223
7.5.3	ext3 文件存储分配策略	192	9.1.1	系统管理内容	223
7.6	文件系统的管理	193	9.1.2	系统管理工具	223
7.6.1	文件系统的注册和注销	193	9.1.3	root 的权威性与危险性	224
7.6.2	文件系统的安装	193	9.1.4	启动与关闭系统	224
7.6.3	文件系统的查看	195	9.2	用户管理	225
7.6.4	文件系统的卸载	197	9.2.1	用户(组)管理概述	225
7.7	文件的打开与读写	197			

9.2.2	用户和组管理	227	10.4.6	配置 DHCP 客户端	291
9.2.3	用户和组配置文件	234	10.4.7	DHCP 服务器配置案例	292
9.3	文件系统管理	239	10.5	DNS 服务器	294
9.3.1	文件系统的目录结构	239	10.5.1	DNS 概述	295
9.3.2	存储设备命名规则	240	10.5.2	DNS 查询模式	295
9.3.3	文件系统操作	242	10.5.3	DNS 服务器的类别	297
9.4	系统备份	246	10.5.4	安装 BIND 软件包	298
9.4.1	备份策略	247	10.5.5	BIND 服务的配置文件	300
9.4.2	备份内容	247	10.5.6	BIND 的启动和停止	306
9.4.3	备份命令	248	10.5.7	DNS 客户端的配置	308
9.5	系统监控	250	10.5.8	DNS 服务器配置案例	309
9.5.1	监视用户的登录	250	10.6	本章小结	312
9.5.2	监视进程的运行	251	10.7	思考与练习	312
9.5.3	监视内存的使用	254	第 11 章	Linux 操作系统安全管理	314
9.5.4	监视文件系统的使用	254	11.1	操作系统安全性概述	314
9.6	软件安装	255	11.1.1	计算机系统安全性的威胁 和特性	314
9.6.1	软件打包与安装	255	11.1.2	操作系统的安全性	315
9.6.2	RPM 软件包管理工具	256	11.1.3	计算机系统安全性评价 的标准	315
9.7	本章小结	261	11.2	操作系统的安全机制	317
9.8	思考与练习	261	11.2.1	内存保护机制	317
第 10 章	Linux 网络配置与管理	262	11.2.2	用户身份认证机制	318
10.1	网络配置基础	262	11.2.3	访问控制技术	319
10.1.1	网络相关概念	262	11.2.4	加密技术	320
10.1.2	TCP/IP 协议概述	263	11.2.5	病毒及其防治机制	321
10.2	Linux 网络配置	264	11.2.6	监控和审计日志	321
10.2.1	网络配置文件	264	11.3	Linux 系统的安全设置	322
10.2.2	网络配置实例	266	11.3.1	系统记录文件的安全性管理	323
10.3	Samba 服务器	269	11.3.2	启动和登录系统的安全性 设置	323
10.3.1	了解 Samba	269	11.3.3	限制网络访问的设置	324
10.3.2	Samba 服务工作原理	270	11.3.4	增强系统的安全性设置	325
10.3.3	安装 Samba 服务器	272	11.3.5	防止攻击的设置	327
10.3.4	Samba 服务的配置文件	274	11.4	Linux 系统的防火墙管理	328
10.3.5	启动和测试 Samba 服务	277	11.4.1	防火墙简介	328
10.3.6	Samba 服务器配置案例	279	11.4.2	防火墙的类型和设计策略	329
10.4	DHCP 服务器	281	11.4.3	Linux 常用的网络命令	330
10.4.1	DHCP 概述	281	11.4.4	配置 Linux 防火墙	333
10.4.2	DHCP 工作原理	282	11.5	本章小结	335
10.4.3	安装 DHCP 服务器	284	11.6	思考与练习	335
10.4.4	DHCP 服务的配置文件	285			
10.4.5	DHCP 服务的启动与停止	289			

# 第 1 章 Linux 操作系统概述

操作系统 (Operating System, OS) 是计算机重要的系统软件, 它的功能是实现计算机软硬件资源管理和隐藏计算机硬件操作细节, 并以接口的方式向应用程序提供通用底层服务, 为应用程序搭建一个友好的开发和运行平台。当今主流的操作系统主要有 Windows、UNIX 以及 iOS 等。本章在讨论操作系统相关概念基础上, 主要介绍的是长久以来非常流行, 一直占有较大市场份额且具有较好的市场应用前景的操作系统——Linux 操作系统。Linux 是一套免费使用和自由传播的类 UNIX 操作系统, 具有良好的应用前景。

## 1.1 认识操作系统

操作系统是管理和控制计算机硬件与软件资源的计算机程序, 是直接运行在“裸机”上的最基本的系统软件, 其他任何软件都必须在操作系统的支持下才能运行。

### 1.1.1 操作系统的诞生

操作系统是一个从无到有、从简单到复杂的发展过程。本节从操作系统的演变、硬件和软件的发展轨迹来介绍操作系统的发展。

#### 1. 操作系统的演变

在计算机诞生的初期, 硬件价格昂贵, 没有操作系统。每一个用户都要自行编写涉及硬件的源代码。程序通过卡片输入计算机, 一次只能完成一个功能 (计算、I/O、用户思考/反应), 工作效率非常低。最早出现的操作系统是简单的单道批处理系统, 它能串行执行预先组织好的一组任务。这种系统避免了此前系统一次只能运行一个任务, 每个任务必须先装入系统, 执行完之后才能装入下一个任务而浪费了装入时间的现象, 提高了系统效率。

但是, 程序运行到 I/O 操作期间, CPU 总是需要停下来等待数据传输完成, 而 I/O 操作时间比 CPU 处理数据时间要高出数十倍 (往往是 20 倍以上), 因此无形中浪费了大量的 CPU 时间, 也使得任务组中后续程序的执行被延迟, 那么, 如何避免数据传输等待所带来的时间浪费? 能否在传输期间解放 CPU, 使其可以去执行其他的任务? 为解决这个问题, 单道批处理系统发展成为多道批处理系统。所谓多道, 就是指处理器 (指单处理器系统) 可以交错运行多个程序, 在某个任务挂起时运行另一个程序。这样就解决了 CPU 等待数据传输所浪费的时间, 进一步提高了系统的运行效率。

当计算机所处理的任务不再仅仅局限于科学计算, 而是越来越多地涉及办公和日常活动时, 程序在执行过程中常常需要和用户不断交互, 任务执行结果随时都会因为用户的选择而改变, 而且往往需要多个用户同时使用系统。由于这种交互模式和共享模式需要任务响应时间尽可能短 (如果超过 20 秒, 人的思维就容易被打断或变得不耐烦), 为了让多数用户满意, 操作系统开始采用分时技术, 将处理器的运行时间分成数片, 平均或依照一定权重分发给系统中的各用户使用。这种使处理器虚拟地由多个用户共同使用的方式, 不但可以满足快速响应的要

求，也可以使所有用户产生计算机完全是在仅为自己服务的感觉。

上面给出了操作系统发展的几个主要阶段：单道批处理，多道批处理，分时系统。除此以外，现在还出现了分布式操作系统、嵌入式操作系统，不过总体技术思路仍然脱离不了多道、分时等概念。

## 2. 硬件的发展轨迹

操作系统理论是在计算机的应用中诞生并成长的，它的发展与计算机硬件的发展是密不可分的，表 1-1 是从硬件角度展现操作系统的发展轨迹。

表 1-1 从硬件角度看操作系统发展轨迹

年 代	硬件特点	操作系统特点	背 景
机械计算机时代（17 世纪 ~20 世纪初）	纯机械结构，低速只能进行简单的数学运算	纯手工操作	从计算尺到差分机再到分析机发展了数百年
第一代计算机（1946 年 ~20 世纪 50 年代末）电子管计算机	体积大，能耗高，故障多，价格高难以普及应用	无操作系统（程序以机器码编写，载体从插件板到卡片与纸带）	1906 年发明电子管；1946 年第一台电子管计算机 ENIAC 研制成功
第二代计算机（20 世纪 50 年代末 ~60 年代中期）晶体管计算机	采用印制电路；稳定性与可靠性大大提高批量生产成为可能；进入实际应用领域但数量有限	单道批处理系统以监督的软件形式出现，任务按顺序处理	1947 年发明晶体管
第三代计算机（20 世纪 60 年代中期 ~70 年代初）集成电路计算机	体积减小，性价比迅速提高；小型计算机发展迅速；进入商业应用领域，但尚不适合家庭应用的需求	涌现大批操作系统，包括多道批处理系统、分时系统和实时系统形成了现代操作系统的基本框架	1958 年发明集成电路；1971 年 Intel 公司发明微处理器
第四代计算机（20 世纪 70 年代中期至今）大规模集成电路计算机	性能大幅度提高，价格不断下降；个人计算机成为市场的主流；计算机迅速普及；计算机应用进入高速发展的轨道	操作系统的理论基本完善；系统与网络通信一体化（分布式操作系统和网络操作系统）；人机交互成为设计重点，操作系统性能日渐稳定	1981 年 IBM - PC 诞生；1993 年 Internet 开始商业化运作

从表 1-1 可以看出：

- 在硬件的性价比比较低时，操作系统设计追求硬件的使用率，从批处理系统发展到分时系统。
- 随着硬件性价比越来越高，操作系统的设计开始追求系统的可靠性和稳定性，出现了多处理器系统和分布式系统。
- 计算机普及后，操作系统的设计开始追求用户界面的友好。
- 第一代和第二代计算机系统应用范围很小，操作系统的发展非常缓慢，直到第三代计算机系统出现后，才得以高速发展。
- 从第三代到第四代计算机，操作系统的功能模块划分没有变化，说明计算机硬件结构已经稳定，操作系统的发展逐渐摆脱随硬件一起发展的状况，形成自己的理论体系。
- 进入第四代计算机系统后，分布式系统和多处理器系统虽然极大地扩充了操作系统理论，但其系统结构并没有变化，只是各功能模块得以进一步完善。

总的来讲，随着操作系统理论的不断发展和完善，操作系统设计中与硬件相关部分所占的比重越来越小，渐渐走出软件依附于硬件的局面，至今操作系统设计已经支撑起一个庞大的软件产业。

### 3. 软件的发展轨迹

操作系统首先是一个软件，它的设计脱离不了软件的范畴。从纯软件发展的角度对其进行考察，有助于了解操作系统的历史，表 1-2 从软件设计的角度给出了操作系统的发展轨迹。

表 1-2 软件设计角度下操作系统的发展轨迹

时期	主流操作系统	系统特点	计算机语言	软件特点	背景
无软件时期	无	手工操作	无编程语言，直接使用机器代码	手工操作	1936 年图灵提出图灵机模型
系统雏形期	单道批处理系统	作业运行的监督程序	编程语言雏形期	无交互机制	1957 年 FORTRAN 语言开发成功
操作系统理论的成形期	多道批处理系统、分时系统实时系统、多处理系统	操作系统结构确立，分为处理机管理、内存管理、设备管理、文件管理等模块	编程语言大量涌现，结构化程序设计 C 语言逐渐成为主导	字符人机交互界面，操作命令繁多	20 世纪 60 年代的软件危机引发了软件工程的发展；1969 年 UNIX 诞生；1972 年推出 C 语言
现代操作系统时期	类 UNIX 系列、Windows 系列	人机交互成为主题；可视化界面；多媒体技术	面向对象语言成为主流	过渡至图形界面，注重操作可视化	20 世纪 80 年代中期，面向对象技术开始逐步发展
网络时代	网络操作系统分布式操作系统	微内核技术兴起	Java 语言和脚本语言兴起	追求设计个性化；注重感官效果	1995 年推出 Java
开源软件时代	嵌入式系统	单内核与微内核竞争激烈	编程工具向跨平台方向发展	可移植性成为主题	1991 年发布了免费的操作系统 Linux

分析表 1-2，可以了解：

- 程序设计理论约束着操作系统设计，操作系统发展滞后于计算机语言的发展。从结构化设计到对象化设计，操作系统总是最后应用新编程理论的软件之一。
- 人机交互技术主要是为用户考虑，这是操作系统设计方面的变革。
- 以 Linux 为代表的开源软件的出现，打破了带有神秘色彩的、传统的封闭式开发模式。

### 4. 单内核操作系统与微内核操作系统

#### (1) 单内核操作系统

单内核操作系统也叫集中式操作系统。整个系统是一个大模块，可以被分为若干逻辑模块，即处理器管理模块、存储器管理模块、设备管理模块和文件管理模块，其模块间的交互是通过直接调用其他模块中的函数实现的。

单内核模型以提高系统执行效率为设计理念，因为整个系统是一个统一的内核，所以其内部调用效率很高。单内核的缺点也正是由于其源代码是一个整体而造成的，通常各模块之间的界限并不特别清晰，模块间的调用比较随意，所以进行系统修改或升级时，往往“牵一发而动全身”，导致工作量加大，使其难于维护。

#### (2) 微内核操作系统

微内核操作系统是指把操作系统结构中的内存管理、设备管理、文件系统等高级服务功能尽可能地从中分离出来，变成几个独立的非内核模块，而在内核中只保留少量最基本的功能，使内核变得简洁可靠，因此叫微内核。

微内核实现的基础是操作系统理论层面的逻辑功能的划分。几大功能模块在理论上是相互独立的，形成比较明显的界限，其优点如下。

- 充分的模块化，可独立更换任一模块而不会影响其他模块，从而方便第三方开发、设计模块。

- 未被使用的模块功能不必运行，因而能大幅度减少对系统的内存需求。
- 具有很高的可移植性，理论上只需要单独对各微内核部分进行移植修改即可。由于微内核的体积通常很小，而且互不影响，因此工作量很小。

但是，因为各个模块与微内核之间是通过通信机制进行交互的，微内核系统运行效率较低。微内核是面向对象理论在操作系统设计中的产物，在实际应用中，微内核尚处于发展阶段。

### 1.1.2 操作系统的目的及作用

可以从不同的角度来认识操作系统的目的和作用。从使用者的角度看，操作系统使得计算机易于使用；从程序员的角度看，操作系统把软件开发人员从与硬件打交道的烦琐事务中解放出来；从设计者的角度看，有了操作系统，就可以方便地对计算机系统中的各种软、硬件资源进行有效的管理。

#### 1. 从使用者角度

人们对操作系统的认识一般是从使用开始的，打开计算机，呈现在眼前的首先是操作系统。如果用户打开的是操作系统字符界面，就可以通过命令行完成需要的操作。例如，要在 Linux 下复制一个文件，则输入：

```
cp /floppy/TEST mydir/test
```

上述命令可以把 /floppy 目录下的 TEST 文件复制到 mydir 目录下，并更名为 test。

为什么可以这么方便地复制文件？操作系统为此做了什么工作？首先，文件这个概念是从操作系统中衍生出来的。如果没有文件这个实体，就必须指明数据存放的具体物理位置，即位于哪个柱面、哪个磁道、哪个扇区。其次，数据转移过程是复杂的 I/O 操作，一般用户无法关注这些具体的细节。最后，这个命令的执行还涉及其他复杂的操作，但是，因为有了操作系统，用户只需要知道文件名，其他烦琐的事务完全由操作系统来处理。

如果用户在图形界面下操作，上述处理就更加容易。实际上，图形界面的本质也是执行各种命令，例如，如果复制一个文件，那么就要调用 cp 命令，而具体的复制操作最终还是由操作系统去完成。因此，不管是敲击键盘或者单击鼠标，这些简单的操作在指挥着计算机完成复杂的处理过程。正是操作系统把烦琐留给自己，把简单留给用户。

#### 2. 从程序开发者角度

从程序开发者的角度看，不必关心如何在内存存放变量、数据，如何从外存存取数据，如何把数据在输出设备上显示出来等。例如，cp 命令的 C 语言实现片段如下。

```
inf = open( "/floppy/TEST" ,O_RDONLY,0);
out = open( "/mydir/test" ,O_WRONLY,0600);
do{
    l = read( inf,buf,4096);
    write( outf,buf,l);
} while(l);
close( outf);
close( inf);
```

在这段程序中，用到 4 个函数 open(), close(), write() 和 read(), 它们都是 C 语言函数



库中的函数。进一步研究可知，这些函数都要涉及 I/O 操作，因此，它们的实现必须调用操作系统所提供的接口，也就是说，打开文件、关闭文件、读写文件的真正操作是由操作系统完成的。这些操作非常烦琐，对于不同的操作系统其具体实现过程也可能不同，程序开发者不必关心这些具体操作。

### 3. 从操作系统在整个计算机系统中所处位置

计算机系统层次结构示意图如图 1-1 所示。

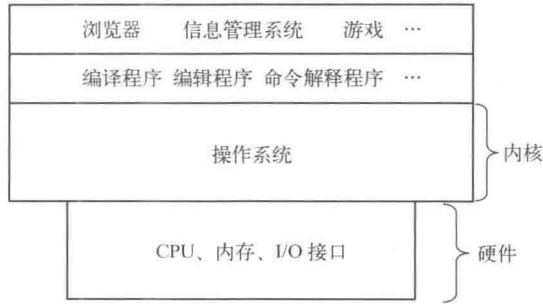


图 1-1 计算机系统层次结构示意图

因为操作系统这个术语越来越大众化，因此许多用户把它们在显示器上看到的東西理所当然地认为是操作系统，例如 Windows 中的图形界面、IE 浏览器、系统工具集等，这些都是操作系统的一部分。但是，本书讨论的操作系统是指内核（Kernel）。用户界面是操作系统的外在表象，而内核是操作系统的内在核心，由它真正完成用户程序所要求的操作。从图 1-1 可以看出，一方面，操作系统是上层软件与硬件相联系的窗口和桥梁；另一方面，操作系统是其他所有用户程序运行的基础。

下面以一个名为 test.c 的 C 程序的执行过程为例，了解一下操作系统的具体作用。

```
#include <stdio.h>
main()
{
    printf("Hello world\n");
}
```

用户对上述程序编译、连接后，生成一个可执行的二进制文件，其机器执行过程如下。

- 1) 用户告诉操作系统执行 test 程序。
- 2) 操作系统通过文件名找到该程序。
- 3) 检查其类型，检查程序首部，找出代码和数据存放的位置。
- 4) 文件系统找到第一个磁盘块。
- 5) 操作系统建立程序的执行环境。
- 6) 操作系统把程序从磁盘装入内存，并跳到程序开始处开始执行。
- 7) 操作系统检查字符串的位置是否正确。
- 8) 操作系统找到字符串被送往的设备。
- 9) 操作系统将字符串送往该设备，窗口系统确定这是一个合法的操作，然后将字符串转换成像素。
- 10) 窗口系统将像素写入存储映像区。

11) 视频硬件将像素表示转换成一组模拟信号, 用于控制显示器 (重画屏幕)。

12) 显示器发射电子束, 在显示器上显示 “Hello world”。

从这个简单的例子可以看出, 任何一个程序的运行只有借助于操作系统才能得以顺利完成。因此, 从本质上说, 操作系统是应用程序运行的基础设施。

#### 4. 从操作系统设计者的角度

操作系统是一个庞大、复杂的系统软件, 其设计目标有两个: 一是尽可能地方便用户使用计算机; 二是让各种软件资源和硬件资源高效、协调地运转。

笼统地说, 计算机的硬件资源包括 CPU、存储器和各种外部设备。其中, 外部设备种类繁多, 如磁盘、鼠标、网络接口、打印机等。操作系统对外部设备的操作是通过 I/O 接口进行的, 软件资源主要指存放在存储介质上的文件。

假设在一台计算机上有 3 道程序同时运行, 并试图在一台打印机上输出运算结果, 这意味着必须考虑以下问题: ①3 道程序在内存中如何存放? ②什么时候让某个程序占用 CPU? ③怎样有序地输出各个程序的运算结果? 这些问题的解决都必须求助于操作系统, 也就是说, 操作系统必须对内存、CPU 进行管理, 当然也包括对外部设备的管理。因此, 从操作系统设计者的角度考虑, 一个操作系统必须包含以下几部分: 操作系统接口、CPU 管理、内存管理、设备管理、文件管理。

综上所述, 操作系统是计算机系统中的一个系统软件, 是一些程序模块的集合——它们能以尽量有效、合理的方式组织和管理计算机的软、硬件资源, 合理的组织计算机的工作流程, 控制程序的执行, 并向用户提供各种服务功能, 使得用户能够灵活、方便、有效地使用计算机, 使整个计算机系统能高效、顺畅地运行。

### 1.1.3 操作系统的主要功能

操作系统的主要功能是资源管理、程序控制和人机交互等。计算机系统的资源可分为设备资源和信息资源两大类。设备资源指的是组成计算机的硬件设备, 如中央处理器、主存储器、磁盘存储器、打印机、磁带存储器、显示器、键盘和鼠标等。信息资源指的是存放于计算机内的各种数据, 如文件、程序库、知识库、系统软件和应用软件等。

操作系统位于底层硬件与用户之间, 是两者沟通的桥梁。用户可以通过操作系统的用户界面, 输入命令。操作系统则对命令进行解释, 驱动硬件设备, 实现用户要求。以现代观点而言, 一个标准个人计算机的 OS 应该提供如下的功能。

- 进程管理。
- 内存管理。
- 文件系统。
- 网络通信。
- 安全机制。
- 用户界面。
- 驱动程序。

#### 1. 资源管理

系统的设备资源和信息资源都是操作系统根据用户需求按一定的策略来进行分配和调度的。操作系统的存储管理就负责把内存单元分配给需要内存的程序以便其执行, 在程序执行结束后将它占用的内存单元收回以便再使用。对于提供虚拟存储的计算机系统, 操作系统还要与

硬件配合以做好页面调度工作，根据执行程序的要求分配页面，执行页面调入和调出内存以及回收页面等操作。

处理器管理，也称处理器调度，是操作系统资源管理功能的另一个重要内容。在一个允许多道程序同时执行的系统中，操作系统会根据一定的策略将处理器交替地分配给系统内等待运行的程序。一道等待运行的程序只有在获得了处理器资源后才能运行。一道程序在运行中若遇到某个事件，例如启动外部设备或一个外部事件的发生等，操作系统就要来处理相应的事件，然后再将处理器资源重新分配。

操作系统的设备管理功能主要是分配和回收外部设备以及控制外部设备按用户程序的要求进行操作等。对于非存储型外部设备，如打印机、显示器等，它们可以直接作为一个设备分配给一个用户程序，在使用完毕后回收以便给另一个有需求的用户使用。对于存储型的外部设备，如磁盘、磁带等，则是提供存储空间给用户，用来存放文件和数据。存储性外部设备的管理与信息管理是密切结合的。

信息管理是操作系统的—个重要的功能，主要是向用户提供一个文件系统。一般来说，一个文件系统向用户提供创建文件、撤销文件、读写文件、打开和关闭文件等功能。有了文件系统后，用户可按文件名存取数据而无须知道这些数据存放在哪里。这种做法不仅便于用户使用而且还有利于用户共享公共数据。此外，由于文件建立时允许创建者规定使用权限，这样就可以保证数据的安全性。

## 2. 程序控制

一个用户程序的执行自始至终是在操作系统的控制下进行的。一个用户将他要解决的问题用某一种程序设计语言编写了一个程序，然后就将该程序连同对它执行的要求输入到计算机内，操作系统就根据要求控制这个用户程序的执行直到结束。操作系统控制用户的执行主要有以下一些内容：调入相应的编译程序，将用某种程序设计语言编写的源程序编译成计算机可执行的目标程序，分配内存等资源将程序调入内存并启动，按用户指定的要求处理执行中出现的各种事件以及与操作员联系请示有关意外事件的处理等。

## 3. 人机交互

操作系统的人机交互功能是决定计算机系统“友善性”的一个重要因素。人机交互功能主要靠可输入/输出的外部设备和相应的软件来完成。可供人机交互使用的设备主要有键盘、显示器、鼠标、各种模式识别设备等。与这些设备相应的软件就是操作系统提供人机交互功能的部分。人机交互功能的主要作用是控制有关设备的运行和理解并执行通过人机交互设备传来的有关的各种命令和要求。

## 4. 进程管理

不管是常驻程序或者应用程序，它们都以进程为标准执行单位。当年运用冯·诺依曼架构设计建造第一台电子计算机时，每个 CPU 最多只能同时执行一个进程。早期的 OS（例如 DOS）不允许任何程序打破这个限制，且 DOS 同时也只有一个进程被执行。现代操作系统，即使只拥有一个 CPU，也可以利用多进程（Multitask）功能同时执行多个进程。进程管理指的是操作系统调整多个进程执行顺序的功能。

由于大部分的计算机只包含一个 CPU，在单内核的情况下多进程只是简单迅速地切换各进程，让每个进程都能够被执行；在多内核或多处理器的情况下，所有进程通过多种协同技术在各处理器或内核上转换。越多进程同时被执行，每个进程能分配到的时间比率就越小。很多操作系统在遇到此问题时会出现诸如音效断续或鼠标跳格的情况（称作崩溃，一种操作系统