

国家信息产业部全国网络与信息技术中心
广东省Linux公共服务技术支持中心
NTC-Linux系列认证惟一指定教材



Linux系统 管理与开发系列丛书

广东省Linux公共服务技术支持中心
组编

Linux

系统开发员

16.81
08



机械工业出版社
CHINA MACHINE PRESS



Linux 系统管理与开发系列丛书

Linux 系统开发员

广东省 Linux 公共服务技术支持中心 组编

田 钧 何世晓 主编

余 琴 周立发 徐永俊 编著



机械工业出版社

本书是 NTC-Linux 系列软件开发部分的初级教程, 针对的“NTC-Linux”认证级别为 Linux 系统开发员。本书主要介绍了 Linux 系统环境下编程的基础知识, 主要内容包括 Linux 操作系统的基础理论及基本操作、Linux 下的常见编程开发, 详细介绍了 Linux 下的 C 编程环境的搭建, 包括编译器 gcc、调试器 gdb、库函数 glibc、工程管理器 make 等工具的使用, 并讲解了 Linux 下文件 I/O 编程、MySQL 数据库接口编程、bash 编程、GUI 开发以及基于 LAMP 的编程开发。学完本书的读者将具有 Linux 系统环境下编程所必需的基本技能, 并为进一步学习 Linux 系统环境下软件开发的中、高级技能打下良好的基础。

本书为国家信息产业部全国网络与信息技术中心 (NTC) 和广东省 Linux 公共服务技术支持中心 (GDLC) NTC-Linux 认证的指定培训教材, 适用于参加 Linux 系统开发员认证的考生, 以及各大高校计算机及相关专业的学生, 也可供希望转入 Linux 开发领域的工程技术人员参考使用, 还可作为广大 Linux 爱好者的参考书。

图书在版编目 (CIP) 数据

Linux 系统开发员 / 广东省 Linux 公共服务技术支持中心组编. —北京: 机械工业出版社, 2007.5

(Linux 系统管理与开发系列丛书)

ISBN 978-7-111-21342-0

I. L… II. 广… III. Linux 操作系统—程序设计 IV. TP316.89

中国版本图书馆 CIP 数据核字 (2007) 第 055793 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: 丁 诚

责任编辑: 韩 菲

责任印制: 洪汉军

北京汇林印务有限公司印刷

2007 年 5 月第 1 版·第 1 次印刷

184mm×260mm·15 印张·370 千字

0001—5000 册

标准书号: ISBN 978-7-111-21342-0

定价: 23.00 元

凡购本书, 如有缺页, 倒页, 脱页, 由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379739

封面无防伪标均为盗版

Linux 系统管理与开发系列丛书编委会

丛书主编:

袁 泉: 广东省 Linux 公共服务技术支持中心 主任

广东省开源软件产业发展促进委员会 会长

胡 铮: 国家信息产业部全国网络与信息技术中心 主任

丛书编委: (按姓氏拼音排序)

赵世奎: 国家信息产业部全国网络与信息技术中心机构审核 主任

余世清: 广东省 Linux 公共服务技术支持中心技术部 部长

秦 恺: 广东省 Linux 公共服务技术支持中心-全国软件测评中心 主任

田 钧: 广东省开源软件产业发展促进会 副秘书长

广东省高校 Linux 推进联盟 秘书长

广东省 Linux 公共服务技术支持中心培训部 部长

何世晓: 广东省 Linux 公共服务技术支持中心培训部 技术总监

丛 书 序

从 1946 年世界上第一台电子计算机诞生开始, 计算机的发展已走过了 61 个春秋, 科学技术的日新月异, 为我们开启了一个无比辉煌的电子时代。1991 年, 芬兰赫尔辛基的大学生 Linus Torvalds 开发了 Linux 操作系统, 经过短短的 16 年, Linux 已发展成为当今最流行的操作系统之一。

Linux 具有源代码开放、性能卓越、安全可靠等与生俱来的特点, 在信息化建设中, 无论从稳定性、安全性以及价格等方面来看, Linux 都具有明显的优势。长期以来, 我国一直处在商业操作系统的垄断下, 严重阻碍了我国自主软件产业的发展。Linux 的出现, 为我们提供了一个自由分享人类科学成果的机会, 为我国软件产业的发展提供了一个良好的契机。因而, Linux 在我国信息化建设中将担当起特殊的历史使命, 必将使民族软件产业在自主创新道路上走得更远。

目前, Linux 已经在我国电子政务、电信、金融、教育等信息化建设领域崭露头角, 但其普及的深度和广度还远远不够。因此, 如何利用 Linux 进一步推动我国信息化建设已成为一个备受瞩目的课题。如今, 社会对 Linux 专业人才的需求不断扩大, 在今后的几年内, 高水平的 Linux 专业人才无疑将成为 IT 领域乃至整个就业市场中的热点, Linux 人才的培养迫在眉睫。而 Linux 在我国发展至今, 还没有一个权威的国家级 Linux 技能认证, 显然不利于 Linux 人才的培养与选拔。

正是在这样的背景下, 国家信息产业部全国网络与信息技术中心 (NTC) 与广东省 Linux 公共服务技术支持中心 (GDLC) 联合开发了“NTC-Linux 系列认证”体系, 并推出具有实用性、可读性、权威性的“Linux 系统管理与开发系列丛书”教材。这是中国在培养 Linux 专业人才方面的一件喜事, 也是 GDLC 务实推进我国开源软件发展战略布局中的一个重要举措。

这套丛书共 6 本, 它们分别是《Linux 系统管理员》、《Linux 系统管理师》、《Linux 系统高级管理师》、《Linux 系统开发员》、《Linux 系统开发师》、《Linux 系统高级开发师》。该丛书内容全面, 自成体系, 几乎涵盖了 Linux 系统管理与开发的各个方面。

我们有理由相信, 这套丛书将进一步推动开源软件在中国的普及, 为促进中国软件产业的发展做出积极的贡献。

广东开源软件产业发展促进委员会会长 袁泉
广东省 Linux 公共服务技术支持中心主任

前 言

从 1991 年 Linux 发布第一个版本算起, Linux 已经走过了十几年的路程, 并从一个天才大学生的心血来潮之作演化为名副其实的主流操作系统。Linux 以其安全、可靠、稳定、开源等特点, 不仅打破了微软公司对 PC 操作系统的垄断, 也震撼了 UNIX 前辈们盘踞高端服务器市场的地位, 在桌面、服务器、嵌入式等各个领域都取得了巨大的成功。

但是, 一个成功的操作系统仅仅有优秀的内核还是不够的, 还必须大量的应用程序在它上面运行, 为广大普通用户提供使用操作系统的便利。现在, Linux 已提供了完备的应用程序开发环境, 包括优秀的编辑器和编译器、成熟的调试工具、丰富的库函数。同时, Linux 的绝大部分接口都遵守 POSIX 规范, 继承了 UNIX 的许多编程约定, 使得大多数在传统 UNIX 下运行的应用程序可以很容易地移植到 Linux 平台。

为什么要写本书

随着 Linux 在网络、企业、政府和消费电子市场中逐步占据重要的地位, 关注和投身 Linux 的开发人员越来越多, 但目前市面上介绍 Linux 开发的资料却非常少, 很多非 Linux 平台下的开发人员感到入行无门。同时, Linux 行业发展至今, 还没有出现一个权威的国家级 Linux 技能认证。因此, 为了更好地指导 Linux 爱好者和 IT 业内开发人士学习 Linux 技术, 同时获得权威的 Linux 开发类技能认证, 国家信息产业部全国网络与信息技术中心(简称 NTC)与广东省 Linux 公共服务技术支持中心(简称 GDLC)联合开发了“NTC-Linux 系列认证”体系, 将陆续推出一系列实用、可操作、易读、权威的 Linux 软件开发类教材。

本书的主要内容

本书是 NTC-Linux 系列软件开发部分的初级教程, 也是认证级别为 Linux 系统开发员的指定教程。

全书由 3 部分构成:

第一部分由第 1~3 章组成。这 3 章的内容涵盖了 Linux 操作系统的基础理论及基本操作。从操作系统基础讲起, 介绍了软件技术基础, 最后详细介绍了 Linux 操作系统的安装和常见命令。

第二部分由第 4~6 章组成。这 3 章的内容包括了 Linux 下的开发环境和工具。其中介绍了 Linux 下的 C 编程环境的搭建, 包括编译器 gcc、调试器 gdb、库函数 glibc、工程管理器 make 等工具的使用, 并详细介绍了 C 语言基础和程序调试方法的基本知识。

第三部分由第 7~11 章组成, 是本书的重点内容——Linux 下的编程开发。本部分详细讲解了 Linux 下文件 I/O 编程、MySQL 数据库接口编程、bash 编程、GUI 开发以及基于 LAMP 的编程开发。

本书的特点

本书有如下特点:

1) 每章均有学习目标。在每一章的开始处都详细列出了在本章中读者需要掌握的内容, 方便读者学习总结。

2) 实例丰富, 并与 Linux 开发实践紧密结合, 给出了大量的实例, 供读者进行实际操作。

3) 语言通俗易懂, 对重要内容讲解细致。

4) 图例清晰, 并具有针对性。

本书适合的读者

本书为 NTC 和 GDLC 联合认证的指定认证教材, 适合于参加“NTC-Linux 开发助理工程师”认证的考生, 以及各大高校计算机及相关专业的学生, 也可供希望转入 Linux 开发领域的工程技术人员参考使用, 还可作为广大 Linux 爱好者的参考书。

本书的创作团队

本书创作团队中的全体人员均就职于广东省 Linux 公共服务技术支持中心, 并专职从事开源软件和 Linux 操作系统软件的开发与推广工作。

本书第 2、4、9、10 章由周立发执笔, 第 5、6 章由田钧执笔, 第 7、8、11 章由徐永俊执笔, 第 1、3 章由余琴执笔, 全书由何世晓、田钧统稿, 余琴、周立发从本书的技术角度对本书进行了审阅。

由于时间仓促, 加之水平有限, 本书中一定会有很多不足之处, 敬请读者批评指正。

致谢

在本书的写作过程中, 得到了本单位 (GDLC) 中心主任袁泉博士的大力支持, 也得到了技术部部长、培训业务经理及广大技术部同仁的大力支持。感谢他们为整个创作团队提供了良好的写作环境。另外, 还要感谢培训部的全体同仁, 对创作团队成员的关心、鼓励与支持。

编 者

目 录

丛书序

前言

第1章 操作系统基础..... 1

1.1 计算机基础知识 1

1.1.1 电子计算机的诞生 1

1.1.2 计算机的发展 1

1.1.3 计算机分类 2

1.1.4 计算机的特点 2

1.1.5 计算机的性能指标 3

1.1.6 计算机的应用领域 3

1.2 操作系统概述 4

1.2.1 操作系统的定义 4

1.2.2 操作系统在计算机中的层次结构 5

1.2.3 操作系统的发展和演变 5

1.2.4 宏内核与微内核操作系统 6

1.3 资源管理 7

1.3.1 处理器管理 7

1.3.2 存储管理 7

1.3.3 设备管理 7

1.3.4 文件管理 8

1.3.5 作业管理 8

1.4 内核和用户界面 8

1.4.1 内核 8

1.4.2 用户界面 9

1.5 网络技术基础 11

1.5.1 分类 11

1.5.2 拓扑结构 11

1.5.3 传输媒体 11

1.5.4 通信协议 12

1.5.5 因特网和互联网 12

1.5.6 互联网相关协议 13

第2章 软件技术基础..... 14

2.1 软件定义 14

2.2 软件开发基础 15

2.2.1 软件开发 15

2.2.2 程序设计 15

2.2.3 程序设计语言 15

2.2.4 软件工程 16

2.3 软件测试基础 16

2.3.1 软件测试 16

2.3.2 黑盒测试与白盒测试 17

2.3.3 Alpha、Beta 和 Gamma 测试 17

2.4 软件开发文件基础 17

2.5 软件维护基础 18

2.5.1 软件维护的定义 18

2.5.2 软件维护的特点 18

2.5.3 软件维护的过程 19

2.5.4 软件的可维护性 20

第3章 Linux 系统基础 21

3.1 Linux 简介 21

3.1.1 什么是 Linux 21

3.1.2 Linux 的发展史和特点 21

3.1.3 认识 Linux 的版本 23

3.1.4 Linux 的发展前景 23

3.2 Linux 的安装 24

3.2.1 安装 Linux 前的准备工作 24

3.2.2 安装方式 24

3.2.3 Linux 的安装步骤 25

3.2.4 Linux 的卸载步骤 31

3.3 Linux 系统常用命令 32

3.3.1 显示目录和文件的命令 32

3.3.2 修改目录与文件权限、属主和
属组的命令 34

3.3.3 创建和删除目录的命令 39

3.3.4 创建和删除文件的命令 40

3.3.5 移动命令 41

3.3.6 复制命令 42

3.3.7 显示文件内容命令 44

3.3.8 查找命令 48

3.3.9 过滤命令 51

3.3.10 关机和重启计算机命令 52

3.3.11 压缩和打包命令 55

3.3.12 用户操作命令	60	5.6.1 函数的分类	95
3.3.13 改变目录和查看当前目录命令	63	5.6.2 函数定义的一般形式	96
3.3.14 文件链接命令	64	5.6.3 函数调用	97
3.3.15 帮助命令	65	5.6.4 函数的参数和函数的值	98
3.3.16 其他命令	65	5.6.5 函数的嵌套调用	99
第4章 编程环境	72	5.6.6 函数的递归调用	100
4.1 程序设计概述	72	5.7 常用数据结构	101
4.2 程序设计语言概述	73	5.7.1 数据结构概述	101
4.2.1 机器语言	73	5.7.2 线性表	102
4.2.2 汇编语言	74	5.7.3 栈和队列	105
4.2.3 高级语言	74	5.7.4 串	106
4.3 脚本语言	76	5.7.5 多维数组	108
4.4 搭建C语言编程环境	77	5.7.6 广义表	109
4.4.1 函数库 glibc	78	5.7.7 树	111
4.4.2 编译器 gcc	78	5.8 常用排序算法	114
4.4.3 系统头文件 glibc_header	79	5.8.1 排序概述	114
4.5 GNU C/C++编译器 gcc	79	5.8.2 排序算法分析	114
4.6 make 命令和 makefile	80	5.8.3 插入排序	115
4.6.1 make 命令的语法	80	5.8.4 交换排序	117
4.6.2 make 命令的 flags 选项	80	5.8.5 选择排序	119
4.6.3 makefile 文件的基本结构	80	5.9 常用查找算法	120
4.6.4 makefile 文件的变量	81	5.9.1 查找的基本概念	120
4.6.5 makefile 文件的隐含规则	82	5.9.2 顺序查找	121
4.6.6 makefile 文件的函数	82	5.9.3 二分查找	122
4.6.7 makefile 文件实例	82	5.9.4 分块查找	124
第5章 C语言基础	85	第6章 程序调试	126
5.1 C语言概述	85	6.1 程序调试概述	126
5.1.1 C程序的结构特点	85	6.1.1 程序调试的重要性	126
5.1.2 C语言的字符集	86	6.1.2 常见调试方法	126
5.1.3 C语言词汇	86	6.1.3 调试的目的	126
5.2 C语言程序的输入和输出	87	6.2 gdb 的使用	127
5.2.1 输出语句	87	6.2.1 gdb 简介	127
5.2.2 输入语句	88	6.2.2 gdb 的常用命令	127
5.3 C语言中的数据类型	89	6.2.3 gdb 的基本使用方法	128
5.4 C语言的流程控制	90	6.2.4 gdb 调试实例	131
5.5 C语言中的数组和指针	90	6.3 内存调试	132
5.5.1 数组	90	6.4 strace 的使用	133
5.5.2 指针	92	第7章 C编程操作文件与目录	135
5.5.3 数组指针	94	7.1 简述	135
5.6 C语言中的函数调用	94	7.2 文件与文件描述符	135

7.3 文件 I/O 操作	136	9.8 算术运算	196
7.3.1 open 和 close 函数	136	9.9 读写文件	198
7.3.2 read、write 和 lseek 函数	138	9.10 正则表达式	199
7.4 标准 I/O 开发	140	9.11 程序的调试	200
7.4.1 打开和关闭文件	141	第 10 章 GUI 编程	202
7.4.2 文件读写	143	10.1 什么是 GUI	202
7.4.3 输入输出	145	10.2 Gtk 与 Gnome	203
7.5 文件属性和目录	148	10.3 Gnome 的开发结构	203
7.5.1 文件属性	148	10.3.1 非 Gnome 库	204
7.5.2 目录操作	154	10.3.2 Gnome 库	205
第 8 章 C 编程操作 MySQL 数据库 ..	158	10.3.3 其他库	205
8.1 数据库概述	158	10.3.4 编程语言和编程工具	206
8.2 数据模型和描述	159	10.4 GUI 编程环境的搭建	207
8.3 SQL 语言	161	10.4.1 基础环境搭建	207
8.3.1 SQL 的主要语句	161	10.4.2 Gtk/Gnome 编程环境搭建	208
8.3.2 数据库查询	164	10.4.3 QT/KDE 编程环境搭建	208
8.4 常见数据库	170	10.5 Gtk 编程和 Glade 界面设计	208
8.4.1 Oracle 数据库管理系统	171	10.5.1 Glade 简介	208
8.4.2 Sybase 数据库管理系统	172	10.5.2 Gtk+/Gnome 程序代码框架	208
8.4.3 Ingres 智能关系数据库管理系统 ..	172	10.5.3 Glade 介绍	209
8.4.4 Informix 数据库管理系统	174	10.5.4 Gtk 源代码编辑和 Anjuta	209
8.4.5 DB2 数据库管理系统	174	第 11 章 LAMP	211
8.5 MySQL 数据库	175	11.1 构建 LAMP 环境	211
8.5.1 MySQL 数据库的安装	175	11.1.1 安装一个 Linux 系统	212
8.5.2 MySQL 基础知识	176	11.1.2 安装和设置 Apache2	212
8.5.3 MySQL 数据库日常管理	178	11.1.3 安装 MySQL 并配置	213
8.6 MySQL 数据库 C API	180	11.1.4 安装 PHP 支持	213
8.6.1 数据类型	181	11.1.5 安装 PHP 访问 MySQL 的 支持模块	214
8.6.2 MySQL 数据库 C API 函数	182	11.1.6 其他软件包的安装	214
8.6.3 用 C 语言编程访问 MySQL 数据 库的步骤	183	11.2 HTML	214
第 9 章 bash 编程	186	11.2.1 HTML 的概念	214
9.1 bash 简介	186	11.2.2 HTML 的版本	214
9.2 bash 编程的一般步骤	187	11.2.3 HTML 的未来	215
9.3 bash 程序的执行	188	11.2.4 HTML 文件的组成	215
9.4 变量	189	11.3 PHP 语言编程	219
9.5 流程控制	191	11.3.1 PHP 入门速成	220
9.6 条件检测	194	11.3.2 PHP 编程实例——聊天室设计 ..	225
9.7 引号的运用	196	参考文献	229

第1章 操作系统基础

本章学习目标

- 了解计算机的基础知识。
- 了解操作系统的基本原理和概念。
- 了解资源管理的概念。
- 了解内核和用户界面的区别。
- 了解网络的基础知识。

1.1 计算机基础知识

计算机(Computer)是一种能接收和存储信息,并按照存储在其内部的程序(这些程序是人们意志的体现)对输入的信息进行加工、处理,然后输出处理结果的高度自动化的电子设备。

1.1.1 电子计算机的诞生

(1) 1946年2月,世界上第一台计算机ENIAC在美国诞生,它不具备现代计算机的主要特征——存储程序和程序控制。

(2) 世界上第一台按存储程序功能设计的计算机EDVAC是美国于1946年开始设计,并于1950年研制完成的。

(3) 世界上第一台实现存储程序功能的计算机EDSAC是英国于1947年开始设计,并于1949年5月投入运行的。

1.1.2 计算机的发展

半个多世纪以来,计算机技术取得了突飞猛进的发展。在人类科技史上还没有一门学科可以与电子计算机的发展相提并论。人们根据计算机的性能和当时的硬件技术状况,将计算机的发展分成几个阶段,每一阶段在技术上都是一次新的突破,在性能上都是一次质的飞跃。

电子计算机的发展按电子逻辑器件可划分为4个阶段:

1) 第一代计算机(从ENIAC问世~20世纪50年代初期,为电子管时代)用光屏管或汞延时电路做存储器,输入输出采用穿孔纸带或卡片。软件处于初始阶段,没有系统软件,程序设计语言只有机器语言或汇编语言。应用以科学计算为主。

2) 第二代计算机(20 世纪 50 年代中期~20 世纪 60 年代中期, 为晶体管时代), 用磁芯和磁鼓做存储器, 产生高级程序设计语言和批量处理系统。应用领域扩大至数据处理和事务处理, 并逐渐用于工业控制。

3) 第三代计算机(20 世纪 60 年代中期~20 世纪 70 年代初期, 为中小规模集成电路时代)的主存储器开始采用半导体存储器, 外存储器有磁盘和磁带, 有了操作系统、标准化的程序设计语言和人机会话式的 BASIC 语言。不仅应用于科学计算, 还应用于企业管理、自动控制、辅助设计和辅助制造等领域。

4) 第 4 代计算机(20 世纪 70 年代中期至今, 为大规模超大规模集成电路时代)的应用涉及各个领域, 如办公自动化、数据库管理、图像识别、语音识别、专家系统等, 并且进入了家庭。

从 20 世纪 80 年代开始, 日本、美国和欧洲的一些发达国家都宣布开始新一代计算机的研究。各国普遍认为, 新一代计算机应该是智能型的, 它应能模拟人类的智能行为, 理解人类自然语言, 并继续向着微型化、网络化方向发展。

1.1.3 计算机分类

计算机的种类可按用途、规模或处理对象等多种标准进行划分。

1. 按用途划分

(1) 通用计算机: 该类计算机用于解决一般问题, 使用领域广泛、通用性较强, 在科学计算、数据处理和过程控制等多个领域中都能适用。

(2) 专用计算机: 用于解决某个特定方面的问题, 配有专门的软件和硬件, 如在生产过程自动化控制、工业智能仪表等领域使用的计算机。

2. 按规模划分

(1) 巨型计算机: 应用于国防尖端技术和现代科学计算。巨型计算机的运算速度可达百万亿次每秒, 研制巨型机是衡量一个国家经济实力和科学水平的重要标志。

(2) 大中型计算机: 具有较高的运算速度, 每秒可以执行几千万条指令, 而且有很大的存储空间, 往往用于科学计算、数据处理或作为网络服务器使用。

(3) 小型计算机: 规模较小、结构简单、运行环境要求较低, 一般应用于工业自动控制、测量仪器、医疗设备中的数据采集等方面。小型计算机在作为巨型计算机系统的辅助计算机方面也起到了重要作用。

(4) 微型计算机: 中央处理器(CPU)采用微处理器芯片, 体积小巧轻便, 广泛用于商业、服务业、工厂的自动控制、办公自动化以及大众化的信息处理等方面。

3. 按处理对象划分

(1) 数字计算机: 计算机处理时输入和输出的数值都是数字量。

(2) 模拟计算机: 处理的数据对象直接为连续的电压、温度、速度等模拟量。

(3) 数字模拟混合计算机: 输入输出既可是数字量也可是模拟量。

1.1.4 计算机的特点

计算机是一种高度自动化的信息处理设备。主要特点有处理速度快、计算精度高、记忆能力强、可靠的逻辑判断能力、可靠性高且通用性强。

- (1) 处理速度快: 计算机的运算速度用 MIPS (百万条指令每秒) 来衡量。
- (2) 计算精度高: 数的精度主要由表示这个数的二进制码的位数决定。
- (3) 记忆能力强: 存储器能存储大量的数据和计算机的程序。
- (4) 可靠的逻辑判断能力: 具有可靠的逻辑判断能力是计算机的一个重要特征, 是计算机能实现信息处理自动化的重要基础。
- (5) 可靠性高、通用性强。

1.1.5 计算机的性能指标

计算机的主要技术性能指标有主频、字长、内存容量、存取周期、运算速度及其他指标。

(1) 主频 (时钟频率): 是指计算机 CPU 在单位时间内输出的脉冲数。它在很大程度上决定了计算机的运行速度, 单位为 MHz。

(2) 字长: 是指计算机的运算部件能同时处理的二进制数据的位数。字长决定了计算机的运算精度。

(3) 内存容量: 是指内存储器中能存储的信息的总字节数。通常以 8 个二进制位 (bit) 作为一字节 (Byte)。

(4) 存取周期: 存储器连续两次独立的“读”或“写”操作所需的最短时间, 单位为 ns (纳秒, $1\text{ns}=10^{-9}\text{s}$)。存储器完成一次“读”或“写”操作所需的时间称为存储器的访问时间 (或读写时间)。

(5) 运算速度: 这是一个综合性的指标, 单位为 MIPS。影响运算速度的因素, 主要是主频和存取周期, 字长和存储容量也有影响。

(6) 其他指标: 兼容性 (包括数据和文件的兼容性、程序兼容性、系统兼容性和设备兼容性)、可靠性 (用平均无故障工作时间, 即 MTBF 来衡量)、可维护性 (用平均修复时间, 即 MTTR 来衡量)、允许配置的外部设备的最大数目、计算机系统的文字处理能力、数据库管理系统及网络功能等。性能/价格比是一项综合性评价计算机经济性的指标。

1.1.6 计算机的应用领域

什么样的实际问题可以用计算机来解决呢? 从广义上讲, 如“为我烹制一道可口的菜”这样的问题是无法用计算机来解决的, 计算机本身的优势在于数值计算。因此, 通常用“可计算性”来表明一个问题是否可以用计算机解决。事实上, 很多非数值问题, 如文字识别、图象处理等, 都可以通过转化成数值问题来处理。一个可以用计算机解决的问题应该被定义为“可以在有限步骤内被解决的问题”, 所以像哥德巴赫猜想这样的问题是不属于“可计算问题”之列的。因为计算机没有办法给出数学意义上的证明, 因此也没有任何理由期待计算机能解决世界上所有的问题。

计算机的应用领域, 按其应用特点可分为科学计算、信息处理、过程控制、计算机辅助系统、多媒体技术、计算机通信、人工智能等。

(1) 科学计算: 指计算机应用于解决科学研究和工程技术中提出的数学问题 (数值计算)。一般要求计算机速度快、精度高、存储容量大。科学计算是计算机最早的应用领域。

(2) 信息处理：信息处理主要是指非计算形式的数据处理，包括对数据资料的收集、存储、加工、分类、排序、检索和发布等一系列工作。信息处理包括办公自动化（OA）、企业管理、情报检索、报刊编排处理等。特点是要处理的原始数据量大，而算术运算较简单，有大量的逻辑运算与判断，结果要求以表格或文件形式存储、输出。要求计算机的存储容量大，速度方面则不是最重要的。信息处理目前应用最广，占有所有应用的 80% 左右。

(3) 过程控制：计算机可以用于科学技术、军事领域、工业、农业等各个领域的过程控制。在计算机控制系统中，需要有专门的数字-模拟（D/A）转换设备和模拟-数字（A/D）转换设备。由于过程控制一般都是实时控制，有时对计算机速度的要求不高，但要求可靠性高、响应及时。

(4) 计算机辅助系统：可分为计算机辅助教学（CAI）、计算机辅助设计（CAD）、计算机辅助制造（CAM）、计算机辅助测试（CAT）、计算机集成制造（CIMS）等系统。

(5) 多媒体技术：就是把数字、文字、声音、图形、图像和动画等多种媒体有机地组合起来，利用计算机、通信和广播电视技术，使它们建立起逻辑联系，并能进行加工处理（包括对这些媒体的输入、压缩和解压缩、存储、显示和传输等）的技术。目前多媒体计算机技术的应用领域正在不断拓宽，除了知识学习、电子图书、商业及家庭应用外，在远程医疗、视频会议等领域中都得到了极大的推广。

(6) 计算机通信：是计算机技术与通信技术结合的产物。计算机网络技术的发展将处在不同地域的计算机用通信线路连接起来，配以相应的软件，达到资源共享的目的。

(7) 人工智能：是研究解释和模拟人类智能、智能行为及其规律的一门学科。其主要任务是建立智能信息处理理论，进而设计可以展现某些近似于人类智能行为的计算系统。人工智能学科包括知识工程、机器学习、模式识别、自然语言处理、智能机器人和神经计算等多方面的研究。

1.2 操作系统概述

1.2.1 操作系统的定义

操作系统（Operating System, OS）是管理计算机软硬件资源，控制程序运行，改善人机界面和为应用软件提供运行环境的底层软件。它是安装在硬件平台之上的基本软件。到目前为止，用于通用计算机上的操作系统主要有两大系列：UNIX 系列和微软 Windows 系列。UNIX 系列包括多个组织的操作系统，其中几个主要的是 System V、BSD 和 Linux。Linux 是遵循 POSIX 规范开发的操作系统，保持了与 BSD UNIX 和 System V 的兼容。

在计算机应用的过程中，人们接触最频繁的是操作系统，如磁盘操作系统 DOS、图形界面操作系统 Windows、开放源代码的操作系统 Linux 等。但是，操作系统往往是比较复杂的系统软件，相对于使用而言，要掌握它的运行机制就不是那么容易了。

可以从不同的角度来认识操作系统。从使用者的角度看，操作系统使计算机易于使用。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繁琐事务中解放出来。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各种软硬件资源

进行有效的管理。

1.2.2 操作系统在计算机中的层次结构

如果把操作系统放在整个计算机系统中考虑,其所处的层次结构如图 1-1 所示。

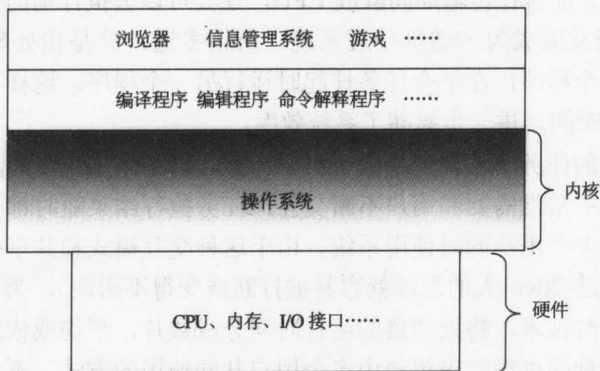


图 1-1 计算机系统层次结构示意图

因为操作系统这个术语越来越大众化,因此许多用户把他们在屏幕上看到的界面理所当然地认为是操作系统的全部,如图形界面、网络浏览器、系统工具等,但这些都是操作系统的一部分。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。

操作系统理论研究者有时把操作系统分成 4 大部分:

(1) 驱动程序:是最底层、直接控制和监视各类硬件的部分,它们的职责是隐藏硬件的具体细节,并向其他部分提供抽象的、通用的接口。

(2) 内核:是操作系统最核心的部分,通常运行在最高特权级,负责提供基础性、结构性的功能。

(3) 支撑库:是一系列特殊的程序库,它们的职责就是把系统提供的基本服务包装成应用程序所能使用的编程接口(API),它是最靠近应用程序的部分。例如,GNU C 运行期库就属于此类,它把各种操作系统的内部编程接口包装成 ANSI C 和 POSIX 编程接口的形式。

(4) 用户程序:指操作系统中所有面向操作员的非上述 3 类的程序,通常是用于提供特定高级服务的部件。例如,UNIX/Linux 中各种守护进程都通常被划归此列。

1.2.3 操作系统的发展和演变

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 处理数据的时间要高出很多（往往是 20 倍以上），因此无形中浪费了大量宝贵的 CPU 的运算时间，也使得任务组中后续程序的执行被延迟。那么，如何避免等待数据传输所带来的时间浪费？能否在传输期间解放 CPU，使其可以去执行别的任务？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 20s，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用户使用。这种使处理器虚拟地由多个用户共同使用的方法，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没有脱离多道、分时等概念。

1.2.4 宏内核与微内核操作系统

1. 宏内核

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

宏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。

宏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，难于维护。

2. 微内核

微内核是指把操作系统中的内存管理、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：

（1）充分的模块化，可独立更换任一模块而不会影响其他模块，从而方便第三方设计、开发模块。

（2）未被使用的模块功能不必运行，因而能大幅度减少系统的内存需求。

（3）具有很高的可移植性。理论上讲，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，因此工作量相对较小。

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，所以微内核的明显缺点是系统运行效率较低。

微内核是面向对象理论在操作系统设计中应用的产物。在实际应用中，微内核尚处于发展阶段。

1.3 资源管理

计算机资源可分为两大类：硬件资源和软件资源。硬件资源指组成计算机的硬件设备，如中央处理器、主存储器、磁带存储器、打印机、显示器、键盘输入设备等。软件资源主要指存储于计算机中的各种数据和程序。系统的硬件资源和软件资源都由操作系统根据用户需求按一定的策略分配和调度。

从资源管理的观点看，操作系统具有 5 大功能：处理器管理、存储管理、文件管理、设备管理和作业管理等。

1.3.1 处理器管理

处理器管理又称进程管理，实质上是对处理器执行“时间”的管理，即如何将 CPU 的运行资源真正合理地分配给每个任务，主要是对中央处理器（CPU）进行动态管理。由于 CPU 的速度要比其他硬件快得多，而且任何程序只有占有了 CPU 才能运行。因此，CPU 是计算机系统中最重要、最宝贵、竞争最激烈的硬件资源。

为了提高 CPU 的利用率，可以采用多道程序设计技术（Multiprogramming）。当多道程序并发（Erupt Simultaneously）运行时，引进进程的概念（将一个程序分为多个处理模块，进程是程序运行的动态过程）。通过进程管理，协调（Coordinate）多道程序之间的 CPU 分配调度、冲突处理及资源回收等关系。

1.3.2 存储管理

存储管理的实质是对存储空间的管理，主要指对内存资源的管理。只有被装入主存储器的程序才有可能去竞争中央处理器。因此，有效地利用主存储器可保证多道程序设计技术的实现，也就保证了中央处理器的使用效率。

存储管理就是要根据用户程序的要求为用户分配主存储区域。当多个程序共享有限的内存资源时，操作系统就按某种分配原则，为每个程序分配内存空间，使各用户的程序和数据彼此隔离（Segregate），互不干扰（Interfere）及破坏；当某个用户程序工作结束时，要及时收回它所占的主存储区域，以便再装入其他程序。另外，操作系统利用虚拟内存技术，把内、外存结合起来，共同管理。

1.3.3 设备管理

设备管理的实质是对硬件设备的管理，其中包括对输入输出设备的分配、启动、完成和回收。设备管理负责管理计算机系统中除了中央处理器和主存储器以外的其他硬件资源，是系统中最具有多样性和变化性的部分，也是系统的重要资源。