

682

7/10-2/1
1/1

UNIX 教程

(修订本)

孟庆昌 等编著



A0937308

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书全面、系统、由浅入深地介绍了 UNIX 系统的使用、原理、开发和管理等内容。全书共分 11 章。第 1 章概述 UNIX 结构和特点;第 2 章介绍 UNIX 命令格式和常用命令;第 3 章介绍 vi 编辑器的各种用法;第 4 章介绍 SCO UNIX 系统的图形环境;第 5 章介绍 UNIX 操作系统内核的组成及实现;第 6 章至第 8 章介绍 shell 程序设计、Korn shell 和 C shell;第 9 章介绍设备管理;第 10 章介绍 UNIX 系统管理;第 11 章介绍 UNIX 的网络通信。每章后面还给出了思考题。

本书是众多 UNIX 用户、系统管理人员和项目开发人员学习和应用 UNIX 系统的理想参考书,还可作为 SCO OpenServer 的培训教材。

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,翻版必究。

图书在版编目(CIP)数据

UNIX 教程(修订本)/孟庆昌等编著.-北京:电子工业出版社,2000.2

ISBN 7-5053-4905-8

I.U… II.孟… III.操作系统,UNIX-教材 IV.TP316.81

中国版本图书馆 CIP 数据核字(1999)第 77100 号

书 名: UNIX 教程(修订本)

编 著 者: 孟庆昌 等

责任编辑: 张燕虹

特约编辑: 刘志林

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京朝阳隆华印刷厂

装 订 者: 三河市新伟装订厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 21.5 字数: 550 千字

版 次: 2000 年 2 月第 1 版 2000 年 4 月第 2 次印刷

书 号: ISBN 7-5053-4905-8
TP·2398

印 数: 5000 册 定价: 28.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;
若书店售缺,请与本社发行部联系调换。电话: 68279077

再版前言

《UNIX 教程》(第一版)发行以后,受到广泛好评,很多院校和培训中心、学习班都把它当作教材或主要参考书。

在一年多的使用过程中,我们广泛征求读者、院校、出版社等各方人员的意见,大家在充分肯定本书价值的同时,也热情地提出建议。这就需要丰富本书的内容,加强实用性。为此,在出版社同志的大力支持下对这本书进行了修订。

修订本书的指导思想是以实用、开发、设计为主,不搞“大全”。为此,在内容上增加了 3 章和部分小节。在第 2 章增加了对一些常用命令(如进程管理命令等)的介绍。很多读者反映,虽然可以利用 man 命令列出各个命令的说明文档,但不方便。增加的 3 章介绍了 Korn shell、C shell 和设备管理。

第一版中只介绍了 Bourne shell,这是基础知识。随着大家对 UNIX 的深入学习和掌握,感到在实际工作中需要具备 Korn shell 和 C shell 的知识。本修订本将 3 个 shell 语言一起介绍,前后呼应,由浅入深,便于自学。Bourne shell 仍是基础,随后重点介绍 Korn shell 和 C shell 的特别之处,从而也节省了篇幅。

在系统中往往配置很多设备,有些设备常用,用户也常进行安装或修改配置,如打印机、终端。有些设备虽常用,但一般不做改动,如硬盘驱动器、光驱等。有的设备相对来说用得较少,如磁带机。权衡关系,在“设备管理”一章中只介绍了对打印机和终端的管理。

对于网络通信没有增加内容。这部分知识很重要,但展开讲会涉及很多内容。思考再三,还是“忍痛割爱”,维持原状。是单写一本书,还是补充内容,以后视情况而定。

UNIX 技术发展很快,普及面越来越广,广大读者的需求水平越来越高。由于作者能力有限,时间又紧迫,书中仍难免有不妥甚至错误之处,恳请广大读者批评指正。

参加此书修订、编写、录入整理等工作的人员有孟庆昌、刘振英、孟平、胡菱、马超、王金柱等。

作 者

1999 年 11 月于北京

前 言

当今全球科技事业处于蓬勃发展的时期,而计算机软件又是其中发展最快的行业之一。个人电脑走进千家万户,网络通信缩短了时空。大发展伴随大竞争。近几年来,Windows 3.1、Windows 95、Windows NT、OS/2 以及 UNIX 之间的市场份额争夺此起彼伏。纵观风云变幻,UNIX 虽受到冲击,但仍在稳定发展。尤其是随着 Internet 应用出现全球“热”之后,UNIX 发展又逢新春。很显然,这主要取决于 UNIX 的一系列特点。

十多年来,中国 UNIX 用户协会一直从事 UNIX 的普及、培训、研究和开发等工作。我和其他同事曾多次在院校、培训班等场合介绍 UNIX 系统。虽在讲授过程中加入新技术介绍,但总感到教材不配套。学员反映更是强烈,迫切希望编写出一本能适合广大读者要求的书籍。本书正是在社会多方人士的推动下促成的。

全书共分 8 章。第 1 章概述 UNIX 的历史、结构和特点;第 2 章介绍 UNIX 常用命令及命令格式;第 3 章介绍文本编辑器——vi 的各种用法;第 4 章介绍 UNIX 图形环境,以 SCO OpenServer 5.0 为例介绍图形环境的概念和桌面系统的各种功能;第 5 章介绍 UNIX 操作系统(内核)的组成和功能模块的实现;第 6 章介绍 shell 程序设计,包括变量、特殊符号、控制结构等;第 7 章介绍 UNIX 系统管理的任务及实现,包括系统安装、用户建立与管理、文件系统管理与维护、文件系统的后备、系统安全性和审计子系统等;第 8 章介绍网络通信,包括协议概述、TCP/IP 命令、对 TCP/IP 的配置和管理等。

在编写本书过程中力图注意以下几点:

1. 突出实用性,兼顾理论性。本书既不同于大学“操作系统”教材,又不是公司产品说明书,而是力求全面、系统地讲述 UNIX 的概念、技术及其应用,使广大读者知道 UNIX 是什么、UNIX 做什么,以及如何让 UNIX 去做。

2. 在选材上既考虑到 UNIX 技术的系统性、先进性,又遵循学习新知识的规律。UNIX 从诞生至今已二十多年,可谓“根深叶茂”,而且新技术不断涌现。课本知识毕竟落后于科技发展。不可能在一、两本书中包涵 UNIX 系统的丰富内容,应尽量让读者提高能力。

3. 在讲法上由浅入深,适于不同知识背景的读者使用。

本书 1~4 章是一般性介绍:整体概述、常用命令和图形界面等;5、6 章是理论和开发部分,介绍 UNIX 内核实现和 shell 编程;最后 7、8 章是系统管理和网络应用部分,这是更具体、更深入一层的内容。在每章后面留有思考题,供读者复习、自查用,以加深对本章内容的理解。

本书的实例都以 SCO OpenServer 5.0 为背景。一方面由于它是 UNIX 家族的佼佼者,另一方面是它在国内市场的份额正在迅速上升。

参加本书编写工作的还有刘振英、孟平等。本书在编写过程中得到过众多同事同仁的大力帮助,尤其是 SCO 公司为本书提供丰富资料。在此向所有关心、帮助本书出版的朋友一并表示衷心感谢。由于技术更新速度非常快,加上个人能力有限,书中会有疏漏、欠缺以至错误之处,敬请广大读者指正。

作 者

1998 年 1 月于北京

第 1 章 UNIX 系统概述

1.1 UNIX 的历史

1.1.1 什么是操作系统

随着信息时代的前进,计算机技术的应用已遍及到社会生活的各个角落,计算机已走入寻常百姓家庭。DOS、Windows 95、OS/2、NT 和 UNIX 等也成为大家熟悉的词语。

构成一个计算机系统要有计算机硬件和软件。CPU、内存、设备(如磁盘、终端、打印机)等信息处理系统中的物理部件或装置统称为硬件;而与之相对的,在数据处理系统中的程序、过程、规则和相关文档统称为软件,它与存放它的介质无关。

上述的 DOS、Windows 95、UNIX 等都是软件系统的名称。按照完成的功能、所起的作用以及在整个系统中所处的地位来分,软件可分为系统软件和应用软件。应用软件是由终端用户编写的或者为用户设计的、完成某种服务的软件或者与用户工作有关的软件,例如电子表格、字处理系统、财务软件以及各种游戏等等。而系统软件是与具体应用无关的软件,它为应用软件的运行提供支持,如操作系统、编译程序、数据库系统等都是系统软件。

操作系统是系统软件的核心,它控制程序的执行和提供资源分配、调度、输入/输出控制 and 数据管理等服务。如 DOS、UNIX、OS/2 和 Windows NT 都是得到广泛使用的操作系统。虽然各种操作系统就其规模、实现、功能、特性及硬件环境等方面而言存在不少差异,但它们提供的服务有很多是共同的,通常包括以下几个方面:

(1) 程序执行——操作系统把要执行的程序装入内存,然后调度它运行。当程序正常完成或发生意外事故而无法运行下去时,必须把它终止。

(2) 资源分配——多个用户或者多个作业同时运行时,每一个都必须分得相应的资源。系统中各类资源都由操作系统统一管理,如 CPU 调度、内存分配、文件存储等,按照一定的策略决定何时把资源分给哪个作业,并确定一次分配多少所需资源。

(3) I/O 操作——正在运行的程序往往需要进行输入/输出,包括文件读写和 I/O 驱动等,操作系统必须处理这些众多繁杂的 I/O 事务。

(4) 文件系统管理——用户的程序和数据等往往都以文件形式保存在系统中,以后用户通过文件名对它们进行存取,需要时就创建或打开文件,不用时就关闭或者删除它。而内部的具体实现由操作系统完成。

(5) 出错检测——程序在机器上运行时可能出现各种错误,有的是硬件故障,有的是人为操作不当(如非法命令),或者是程序本身有错(如算术溢出、地址异常等),操作系统对每类错误都要检测到,并采取相应措施。

(6) 中断处理——系统中往往包含多种中断,如 I/O 中断、时钟中断、程序中断等,操作系统必须对所有中断进行处理。

(7) 统计——往往我们希望了解各个用户对系统资源的使用情况,如使用什么类型的资

源,使用多少等,操作系统要对各种资源的情况进行统计,为改进系统性能提供服务。

(8) 保护——在多道程序运行环境中,作业对各种资源的需求经常发生冲突,如对同一文件、同一内存区或者同一数据的读/写,为此,操作系统必须进行调节和合理的调度。

按结构层次观点来看,计算机系统可大致如图 1-1 所示。



图 1-1 计算机系统组成

从图 1-1 中可以看出,操作系统与硬件的关系相当密切,不仅对硬件资源直接实施控制和管理,而且与硬件相配合实现某些功能,如中断系统。另外,操作系统也为其他软件的运行建造工作环境,所以往往把操作系统称为软件工作平台。

操作系统对外提供了多种服务,从而用户可以方便、安全地使用计算机硬件和运行自己的程序。现代操作系统通常向用户提供三种类型的接口:命令接口——用户从键盘上键入命令,系统提供相应服务;程序接口——也称系统调用接口,用户在自己的程序中使用系统调用,从而获取系统更基层的服务;图形接口——用户利用鼠标、窗口、菜单、图标等图形用户界面工具直观、方便、有效地使用系统服务和各种应用程序及实用工具。

1.1.2 UNIX 的历史

UNIX 是当代最著名的多用户、多进程、多任务的分时操作系统。它的前身是 MULTICS 操作系统。MULTICS 是在 1968 ~ 1969 年间由 MIT、AT&T 和 GE 等众多单位联合开发的大型、多用户分时系统,Bell 实验室的 Ken Thompson 与 Dennis Ritchie 也参加了该项目的开发工作。由于项目过于庞杂,很难达到预期目标,AT&T 于 1969 年退出了该研究项目。

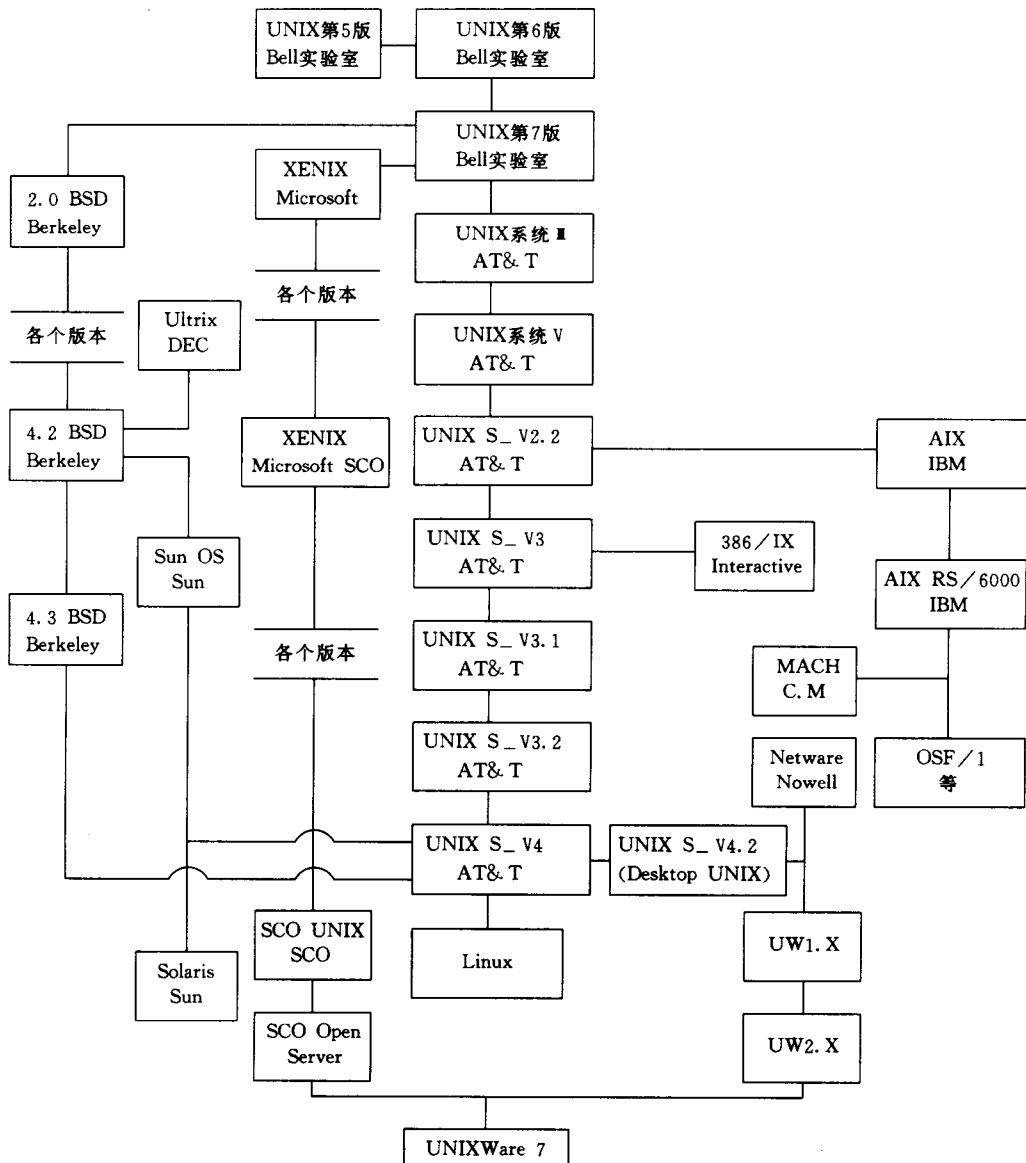
在 1970 年,Ken Thompson 在 GE 的机器上进行一项名为太空旅游的实验计划。由于实验费用太高,而且该机器的 I/O 功能欠缺,为此他用汇编语言在 PDP-7 计算机上设计了一个小型的操作系统,取名为 UNIX,以示对 MULTICS 的区别。1971 年,Dennis Ritchie 开发了 C 语言,并在 1973 年用 C 语言重写了 UNIX,这就成为今日 UNIX 的最初蓝本。从而 UNIX 与 C 语言就紧密地结合在一起。

在 1975 年应学术界的要求 Bell 实验室推出 UNIX Version 6。在 1979 年为满足商业需求推出 UNIX Version 7。在 1983 年 Bell 实验室又推出 UNIX System V,并提出 SVID(System V Interface Definition)。值得一提的是,Berkeley 大学在发展 UNIX 的历史上扮演了重要角色,发布 BSD 版本。

在 UNIX 的发展历程中,还产生了很多其他的商业版本,如 IBM 公司的 AIX、DEC 公司的 Ultrix、Sun Microsystems 公司的 SunOS/Solaris。为了抢占 UNIX 市场,在 1988 年由 IBM、DEC、HP 等公司成立了 OSF(Open Software Foundation)组织;次年(1989 年)由 AT&T、SUN、NEC 和 UNISYS 等公司成立了 UI(UNIX Internationals)联盟。

为了克服不同厂家生产的 UNIX 版本的差异对可移植性的损害,满足开放性的要求,AT&T 和许多国际组织开始对 UNIX 进行标准化工作,如/usr/group 组织了 IEEE1003.1,提出了针对

UNIX 核心的 POSIX(Portable Operating System Interface for Computer Environments)标准,欧洲的 X/OPEN 组织提出了 XPG(X/OPEN Portability Guide)标准。在 1990 年 ISO 制定了 ISO/IEC 9945-1-1990 标准,它以美国 IEEE Std 1003.1-1988 为基础,是整个可移植操作系统界面(POSIX)的第一部分:系统应用程序界面即 POSIX.1。这就是大家常说的 UNIX 操作系统的界面标准。



者。值得一提的是自由软件 Linux 近 2~3 年得到迅速发展。Linux 是遵循 POSIX 标准的操作系统,它不仅可以用在 Intel、AMD 及 Cyrix 系列个人机上,还可用在许多工作站上。它得到许多专业人员以及大型数据库软件公司等广泛支持。它继承了 UNIX 的主要特征,在 Internet 和 Intranet 的应用中占有明显优势。Linux 也为我国软件行业的发展带来新的机遇。

1.2 UNIX 的特点

UNIX 从诞生至今已有 20 多年的历史,其中经历了激烈的市场竞争。特别是近十年来受到 Windows 3.1、Windows 95、Windows NT 以及 OS/2 等产品的强烈冲击。但是 UNIX 系统仍稳定地占有一席之地,是工作站平台上的主导操作系统。尤其是随着 Internet 的高速发展和广泛应用,UNIX 的应用又得到进一步扩大。UNIX 系统所以能取得这样大的成功,是和它具有一系列特点分不开的。UNIX 系统的主要特点可归纳为以下几点:

(1) 可移植性好。UNIX 操作系统和核外实用程序是用 C 语言书写的,因而容易阅读、理解和修改,可移植性良好。虽然在效率上 C 语言比汇编语言稍差,但具有很多汇编语言所无法比拟的优点,它隐藏了具体机器的结构。

(2) 良好的用户界面。UNIX 向用户提供两种界面:用户界面和系统调用。UNIX 的传统用户界面是基于文本的命令行界面,即 shell,它既可以联机使用,又可存在文件上脱机使用。shell 有很强的程序设计能力,用户可方便地用它编制程序,因而为用户扩充系统功能提供了更高级的手段。

UNIX 还为用户提供了图形用户界面。它利用鼠标、菜单、窗口、滚动条等设施的优点,给用户呈现一个直观、易操作、交互性强的友好的图形化界面。

系统调用是用户在编写程序时可以使用的界面。用户可以在编写 C 语言程序时直接使用,系统通过这个界面为用户程序提供低级、高效率的服务。

(3) 树形分级结构的文件系统。UNIX 具有一个树形分级结构的文件系统,它由基本文件系统和若干可装卸的子文件系统组成。这种结构既有利于动态扩大文件存储空间,又有利于安全和保密。

(4) 字符流式文件。在 UNIX 中,文件是无结构的字符流序列,用户可以按需要任意组织其文件格式,对文件既可顺序存取,也可随机存取。另外,在 UNIX 中,把普通数据文件、目录文件和外部设备都统一作为文件处理。它们在用户面前有相同的语法语义,使用相同的保护机构。这就简化了系统设计,又便于用户使用。

(5) 丰富的核外系统程序。UNIX 系统的核外部分包含有非常丰富的语言处理程序、系统实用程序和软件开发用的工具。UNIX 可以提供十几种常用的程序设计语言的编译和解释程序。如 C、FORTRAN、BASIC、PASCAL、Ada、COBOL、LISP 和 PROLOG 等。它提供的实用程序和工具有汇编程序、编译程序、连接装配程序、查错程序、格式排版程序、语言开发工具 YACC 和 LEX,等等。它们均作为文件存于文件系统中。用户通过 shell 命令使用它们。正是这些系统软件为用户提供了相当完备的程序设计环境。

(6) 设计思想先进,核心精干。它是对现有操作系统技术的精炼和发展。在总体设计思想上,它突破以往设计中贪大求全的惯例,而着眼于向用户提供一个良好的程序设计环境。也就是说,UNIX 核心的设计简洁而功能很强。它本身程序不大,但为用户提供了一个很实用的软件运行和软件开发的环境。

UNIX 系统大致分为三层:最里面是 UNIX 内核,即 UNIX 操作系统常驻内存的部分,它直接附着在硬件上;中间层是 shell,即命令解释程序,这是用户与系统核心的接口;最外层是应用层,它包括众多的应用软件、实用程序和除 UNIX 操作系统之外的其他系统软件。图 1-3 示出了 UNIX 系统的概貌和结构。

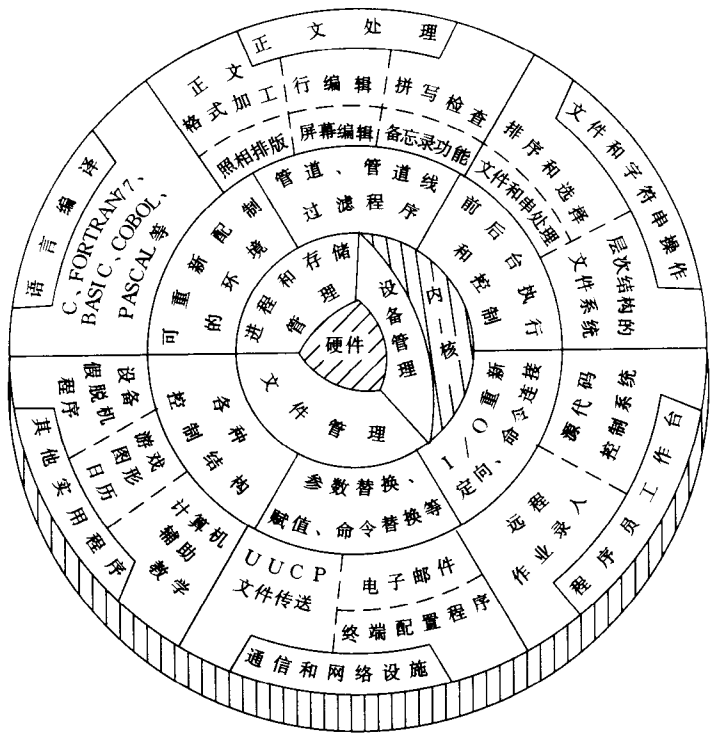


图 1-3 UNIX 系统的结构

(7) 管道文件连通。一个程序的输出可作为另一程序的输入,利用这种管道线机制可把若干程序结合在一起,共同完成复杂的工作。这对于程序开发者和用户都提供了很大方便。

(8) 提供电子邮件和对网络通信的有力支持。UNIX 系统本身提供了 uucp 通信工具,可以实现 UNIX 机器与 UNIX 机器之间经过串行口的通信,可帮助用户完成远程文件传送、远程登录以及远程执行命令。UNIX 除提供标准的 TCP/IP 应用程序外,还支持一组网络服务工具程序,所以在 Internet 网上站点主机大多运行 UNIX 系统。

(9) 系统安全。UNIX 采取了许多安全技术和措施以满足 C2 级安全标准。它包括对读、写进行权限控制、带保护的子系统、审计跟踪、核心授权等,这为网络多用户环境中的用户提供了必要的安全保障。

还可以从其他方面入手总结出若干 UNIX 的特点。当然,UNIX 系统也存在不足,如实时功能较差、易使用性和易安装性不佳、对硬件环境要求较高等。有些问题正逐步得到解决。

思考题

1. 什么是操作系统? 什么是软件? 软件主要分哪两大类? 操作系统属于哪一类?

2. 操作系统的服务主要包括哪些方面？
3. UNIX 系统的主要特点有哪些？
4. 试概括出 UNIX 系统的层次结构。

第 2 章 UNIX 常用命令

在提示符后键入命令,由系统解释执行,这是 UNIX 系统与用户的交互界面。UNIX 系统提供的命令十分丰富。本章介绍 UNIX 的常用命令,如文件操作命令、目录操作命令、口令修改、软盘使用命令及求助命令等。

2.1 进入和退出系统

2.1.1 进入系统

用户利用 UNIX 系统进行工作之前,要执行注册。在字符终端界面环境下,终端屏幕上出现如下注册提示行:

```
login:
```

在它的后面键入自己的注册名(如作者的注册名是 mengqc)。如果你是新用户,那么要与系统管理员联系,取得一个合法的注册名。

键入注册名、并按 < Enter > (回车)键后,在屏幕上出现:

```
password:
```

要求你在其后键入你的口令。在输入口令字符串时,它并不在屏幕上显示出来,以利于保密。输入完口令并按 < Enter > 键后,系统就对你的注册名和口令进行验证。如果确认无误,则在屏幕上显示若干行信息,最后一行上出现: \$ _。

\$ 是一般用户的主提示符,其后是闪烁的光标条。在这种情况下,表示你被系统接受,进入了系统。

如果你尚未建立口令,那么在 password: 之后直接按 < Enter > 键,系统确认后,也在屏幕上显示 \$ 提示符。

如果你键入的注册名或者口令不正确,那么在输入完口令、并按 < Enter > 键后,屏幕上会出现:

```
Login incorrect
```

```
Wait for login retry:
```

```
login:
```

此时,系统要求你重新进行注册。如果经若干次(例如 5 次)注册不成功,则系统给出 login exit 信息。

只有注册成功后,才能进入系统正常工作。

在图形界面环境下注册过程与上面相似,即:在注册窗口中相应的输入框内分别键入用户注册名和口令,然后按 < Enter > 键或用鼠标单击 < login > 按钮。

2.1.2 退出系统

当完成工作后,需要从系统中退出来。退出系统的过程很简单:在 \$ 提示符出现后,同时按下 < Ctrl > 和 < D > 键(通常表示为 < Ctrl > D),屏幕上会再次出现:

```
login;
```

表示你已经从系统中退出。如果你想再次进入系统,必须重新注册。

在 SCO UNIX 中还提供另一种退出系统的方式,即在 \$ 提示符后键入 exit 命令:

```
$ exit < Enter >
```

```
SCO OpenServer ( TM ) Release 5 ( scosysv ) ( tty01 )
```

```
login;
```

如在图形界面环境下想退出系统,可在主窗口下单击 < File > 菜单,然后在所列出的菜单中选择并单击 < Exit >; 在随后出现的“Are you sure you want to log out?”窗口中单击 < OK > 按钮。这样,屏幕上又出现注册窗口。

2.2 简单命令

下面是一些常用的简单命令。

2.2.1 pwd 命令

pwd 命令的功能是显示当前工作目录的全路径名。例如在我的系统中注册以后马上键入命令行:

```
$ pwd < Enter >
```

屏幕上显示:

```
/usr/mengqc
```

```
$
```

表明我当前的工作目录是 /usr /mengqc。由于不同时候、不同用户在不同环境中所用的工作目录会有很大差异,所以由 pwd 命令显示的结果就因具体情况而异。

这里提请读者注意:在所有命令行字符串后都要键入 < Enter > 键,系统才对该命令加以接收、分析、执行。因此在以下的命令行示例中都省去 < Enter >, 作为默认方式,你在实际上机操作时必须在输入命令之后按下 < Enter > 键。

2.2.2 date 命令

date 命令的功能是显示系统当前的日期和时间。例如:

```
$ date
```

```
Wed Aug 13 15:30:58 CST 1997
```

```
$
```

日期是 1997 年 8 月 13 日,星期三,时间是 15 时 30 分 58 秒。

2.2.3 who 命令

who 命令的功能是显示当前已注册到系统的所有用户名、所用终端名和注册到系统的时间。例如：

```
$ who
root    tty02 Aug 13 16:12
mengqc  tty01 Aug 13 15:50
```

表明目前有两个用户在系统中,root(超级用户)使用 tty02 终端,注册时间为 8 月 13 日 16 时 12 分; mengqc 使用 tty01,注册时间为 8 月 13 日 15 时 50 分。

2.2.4 echo 命令

echo 命令的功能是将命令行中的参数重新显示出来。例如：

```
$ echo How are you!
How are you!
```

2.2.5 ls 命令

ls 命令的功能是列出目录中的内容。如果只是简单地键入 ls 命令,后面不带任何参数,那么系统将列出当前工作目录的内容。例如：

```
$ ls
a.out
C+ +1.C
C+ +2.C
m1.c
m2.c
script1
```

这正是当前工作目录/usr/mengqc 中所包含的下属文件名。

2.2.6 cal 命令

cal 命令的功能是显示日历。它可以显示公元 1~9999 年中任意一年或任意一个月的日历。可以不带任何参数直接使用该命令：

```
$ cal
Mon Oct 25 11:29:38 1999

      Sep                      Oct                      Nov
Su Mo Tu We Th Fr Sa      Su Mo Tu We Th Fr Sa      Su Mo Tu We Th Fr Sa
                1  2  3  4                1  2                1  2  3  4  5  6
 5  6  7  8  9 10 11      3  4  5  6  7  8  9      7  8  9 10 11 12 13
12 13 14 15 16 17 18      10 11 12 13 14 15 16      14 15 16 17 18 19 20
19 20 21 22 23 24 25      17 18 19 20 21 22 23      21 22 23 24 25 26 27
26 27 28 29 30           24 25 26 27 28 29 30      28 29 30
                          31
```

从中看出,如果 cal 不带任何参数,则显示上月、本月和下月的日历,以及当前的日期和时间。

在 cal 命令之后可以有一个表示年份的数字,指定显示某一年全年的日历。例如,cal 1998。注意,年份 1998 不能简写为“98”,因为“cal 98”将显示公元 98 年的日历,而不是 1998 年的日历。

可以指定显示某一年某一月的日历,如 cal 8 1996,将显示 1996 年 8 月份的日历。在月份(数字 8)与年份(1996)之间留有空格。月份可以使用英文缩写形式,如 cal Aug 1996。

2.2.7 uname 命令

uname 命令显示所用操作系统的名称。

```
$ uname
SCO_SV
```

uname 命令可带有多个选项。常用选项有:

- m 显示机器硬件名称。
- r 显示操作系统发行版本。
- s 显示操作系统名称(默认行为)。
- v 显示操作系统版本。
- X 显示以下信号:系统名称、节点名称、操作系统发行号、核心 ID 号、处理器类型、总线类型、序列号、用户许可证数(两个用户,八个用户或无限制)、OEM 号、原始号以及 CPU 个数。

2.2.8 logname 命令

logname 命令显示用户的注册名。例如:

```
$ logname
mengqc
```

如果没有找到注册名,则显示出错消息,并返回非 0 终止状态。

2.2.9 env 命令

env 命令显示或者设置当前环境。例如:

```
$ env
EDITOR = /usr/ucb/vi
HOME = /usr/meng
HZ = 100
LOGNAME = mengqc
PATH = /bin:/etc:/usr/bin:/tcb/bin
SHELL = /bin/sh
TERM = ansi
TZ = CST - 8
.....
```

以“变量名 = 值”的形式显示各环境变量的值。以上依次表示:所用编辑器、用户主目录、每秒时钟中断次数、用户注册名、命令的搜索路径、查找所用 shell 的路径、所用终端类型、所在时区等。

2.3 UNIX 命令的一般格式

UNIX 命令行的一般格式是:

命令名称 [选择项] [参数]

其中,命令名是命令的名称,如上述的 `date`, `ls` 等,它总是出现在命令行的开头位置。

选择项是一种标志,常用来扩展命令的特性或功能。[选择项]中的方框号表示语法上选择项可有可无,如前面的示例中都没有出现选择项。选择项往往是一个个英文字母,在字母前面有一个连字符“-”,例如: `ls -l`。

有时也可以把几种表示不同含义的选项字母组合在一起对命令发生作用,例如: `ls -la`。

参数表示命令的自变量,如文件名、参数值等。参数也是可有可无,可多可少,依据具体命令的要求而定,例如: `ls -l /usr/mengqc`。

在命令行中,命令名、选择项和参数彼此之间都需要用空格(通常是这样)或制表符隔开;否则,若连在一起,则往往会出错。

2.4 文件操作命令

2.4.1 文件及其分类

文件是 UNIX 用来存储信息的基本结构。一个文件就是被命名的一组信息。文件名是文件的标识,它通常由字母数字符、下划线符和句点组成。构成文件名的字符串的长度在较老的系统中限定为 14,而在当今多数 UNIX 系统中取消这一限制,允许文件名可以更长。为了易于识别和管理,文件名往往带有后缀(或扩展名),如 `ml.c`, `myfile.o` 等。通常后缀 `.c` 表示 C 源文件, `.C` 表示 C++ 源文件, `.h` 表示前导文件, `.s` 表示汇编程序文件, `.o` 表示目标文件。

UNIX 的文件通常分为三大类:

(1) 普通文件:包括源程序、数据表格、可执行的目标程序、书信文稿、文档等文件,系统处理的绝大部分文件都是普通文件。普通文件的类型标志是“-”。

(2) 目录文件:由所有相关文件的目录组合在一起构成的文件。它类似于学生的花名册。目录文件也往往简称目录。目录的类型标志是“d”。

(3) 特别文件:为了统一管理,UNIX 系统把所有 I/O 设备都按文件格式供用户使用。所以,特别文件就是设备文件,如终端、打印机、磁盘等。特别文件分为块特别文件(如磁盘、磁带、CD-ROM 等)和字符特别文件(如终端、打印机等)。块与字符特别文件的类型标志分别是“b”和“c”。

为了对文件进行保护,UNIX 系统中提供了文件存取控制方式。把所有用户划分为三种身份,依次是:文件主(user)、同组用户(group)和其他用户(other)。每种用户对一个文件可拥有读(r)、写(w)和执行(x)的权利。这样就用 9 个二进制位表示文件的存取方式。例如:

111101100 (对应的八进制数表示为 754)

表示文件主(前三位)对该文件可读、可写、可执行;同组用户(中间三位)可读、可执行;而其他用户(后面三位)只可读。(其中 1 表示置上相应权限。)

用字符形式表示上述文件方式如下所示:

```
rwXr-xr--
```

这 9 位的顺序是固定的。(其中 “ - ” 表示对应用户不具备相应权限。)文件方式可根据需要进行修改。

2.4.2 文件显示命令

1. file 命令

file 命令用来确定文件类型。file 命令对每个参数进行一系列测试以便对它分类。如果参数看来像是 ASCII 文件,则 file 查看它的前 512 个字节,并猜测它的语言。如果参数是可执行的,则 file 显示它的版本号。

file 的句法格式为:

```
file [-ch][ -f ffile][ -m mfile] argument...
```

其中各选项含义为:

- c 为格式错误的文件核查幻数文件(magic)。由于效率的原因通常不执行这个证实。
- f ffile 把 ffile 看作是包含被测文件的名称的一个文件。
- h 不遵循符号连接。按照默认,文件遵循符号连接。
- m mfile 使用 mfile 作为幻数文件。

例如:

```
$ file ex1
ex1: commands text
$ file su_1
su_1: ascii text
$ file ttt
ttt: directory
```

2. cat 命令

cat 命令用来连接并显示文件。它顺序读取每一个文件,并将它们在标准输出上输出。如果没有指定输入文件或者只是给定一个连字符(-),则从标准输入上读取。

cat 的句法格式为:

```
cat [-u][ -s][ -v][ -t][ -e] 文件名...
```

其中各选项含义为:

- s 文件不存在时不产生警告
- u 输出不带缓冲
- v 把非显示字符(制表符、换行和换页除外)显示出来。控制字符被显示成 ^X (< Ctrl > x) 形式,这里 X 是随 < Ctrl > 键一起按下的键,例如 < Ctrl > m 被显示为 ^M。< Del > 字符被显示为 ^?;非 ASCII 字符(高位置 1)被显示为 M - x,这里 x 是低 7 位所确定的字符。
- t 制表符显示为^I,换页符为^L。如果没有使用 -v 选项,则这个选项被忽略。

- e 在每行的结尾(换行符之前)显示一个“\$”字符。如果没有使用 -v 选项,则这个选项被忽略。

例如:

```
$ cat m1.c
```

则在屏幕上显示出文件 m1.c 的内容。

```
$ cat file1 file2 > file3
```

则把文件 file1 和文件 file2 连接起来,放入文件 file3。(此时在屏幕上并不能直接看到该命令执行后的结果。可以键入命令行 cat file3,就可看到连接后的内容。)

3. more 命令

more 命令每次浏览一屏文件。more 允许逐屏测试连续正文。你可以指定被测文件清单,或者利用管道线由另一个命令的输出作为 more 的输入,从而控制其显示。

正常情况下每个满屏之后 more 中止,在该屏底部显示:

```
- - More - - (n%)
```

提示已显示内容占全部内容的百分比。如果按下回车,就显示出下一行;如按下空格键,则显示另一屏。

more 的句法格式为:

```
more [ -cdeilrsuv ] [ -n number ] [ -p command ] [ -t tag ] [ - /pattern ] [ file... ]
```

```
more [ -cdeilrsuv ] [ - number ] [ + command ] [ - t tag ] [ - /pattern ] [file...]
```

其中各选项含义为:

- /pattern 从 pattern(模式)给出的正文模式的第一次出现位置开始显示。
- n number 和 - number 指定显示窗口的行数,more 用它代替默认值。该选项的第二种形式被逐渐抛弃。
- p command 和 + command 指定对每个被测试的文件首先执行 more 命令。该选项的第二种形式被逐渐抛弃。
- t tag 显示一个满屏的包含 tag 字符串的文件。它等价于使用 more 的命令:t。
- c 在文件显示之前先清屏。
- d 在每屏的底部增加提示信息“Hit space to continue, Reboot to abort”。
- e 在写最后文件的最后一行时立即退出。
- i 执行模式匹配,不区分大小写。
- l 不处理 < Ctrl > l(换页)。如果没有给出这个选项,则 more 命令将在包含 < Ctrl > l 字符的行后中止,好像到达满屏的末尾。亦即,如果文件由换页符开头,则文件被显示之前先清屏。
- r 回车符被显示为“^M”。
- s 把输出的多个空白行压缩成一个空白行。
- u 禁止产生下线符的转义序列。
- v 正常情况下 more 命令忽略控制字符。使用该选项后,将使控制字符显示为^C形式,这里 C 代表相应的可显示的 ASCII 字符。不可显示的非 ASCII 字符(高位置 1)显示为 M-C 形式,这里 C 是相应的高位未置 1 的字符。如果输出不是到达终端,more 就