

邢国庆 等编著

UNIX

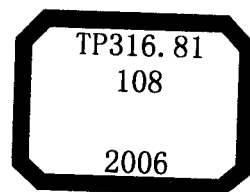
从入门到精通 (普及版)



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



UNIX 从入门到精通 (普及版)

邢国庆 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书从 UNIX 系统的基本命令入手,由浅入深,逐步阐述了 UNIX 操作系统的基本概念和原理,同时给出了大量的应用实例。在此基础上,对 UNIX 的进程管理、文件系统的内部组织结构、文件系统的管理与维护、系统的启动与关机过程、TCP / IP 网络管理、Shell 编程、软件包的制作与安装等方面,进行了深入的讨论。本书内容丰富,语言流畅,实为学习、使用、管理和维护 UNIX 系统的一本必备工具书。

本书可作为学习各种版本的 UNIX 操作系统的参考书。

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,侵权必究。

图书在版编目(CIP)数据

UNIX 从入门到精通(普及版) / 邢国庆等编著. —北京:电子工业出版社, 2006.11

ISBN 7-121-03172-8

I.U... II.邢... III.UNIX 操作系统 IV.TP316.81

中国版本图书馆 CIP 数据核字(2006)第 108221 号

责任编辑:吴源

印 刷:北京天竺颖华印刷厂

装 订:三河市金马印装有限公司

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

北京市海淀区翠微东里甲 2 号 邮编:100036

开 本:787×1092 1/16 印张:36.25 字数:900 千字

印 次:2006 年 11 月第 1 次印刷

定 价:50.00 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 zltts@phei.com.cn。

服务热线:(010) 88258888。

前 言

目前, 存在许多不同版本的 UNIX, 如 IBM 公司的 AIX, Sun 公司的 Solaris, 以及 HP 公司的 HP-UX 等。那么, 学习 UNIX 操作系统究竟应该从何处下手呢? 这是一个令许多 UNIX 初学者感到困惑和迷茫的问题。

20 世纪 80 年代末期, UNIX 失去了一次统一的机会, UNIX 发展到今天, 虽然中间经过诸多努力, 曾尝试研发一个统一版本的 UNIX, 而且这也是业界多年形成的共识, 但是, 由于各个 UNIX 厂家过多顾及自身的立场、利益和偏爱, 使得这一统一过程举步维艰、一波三折, 最终仍然形成 AIX、Solaris、HP-UX 以及 UNIX 的派生体 Linux 各领风骚、多足鼎立的局面。但是, 由于这些产品必须遵循各种公认的标准, 如 X/Open、POSIX 和 SVID 等, 故不同版本的 UNIX 之间其实还是大同小异的, 仍具有很多的共性。

因此, 对于立志学习 UNIX 的人来讲, 只需选用任何一种 UNIX 系统, 从最基本的命令行开始(切记不要走图形界面的捷径), 逐步深入其中, 等到得心应手时, 再学习其他版本的 UNIX, 就会达到触类旁通之效。即使有时遇到不一致的地方, 查一下相关的 UNIX 产品手册, 问题就会迎刃而解。

那么, 怎样才能快速掌握一门 UNIX 技术呢? 依据笔者的经验, 熟悉 UNIX 操作系统的最好方法就是: 创建一个 UNIX 学习环境, 选购一本有效的参考书。一边看书学习, 一边上机操作。这也是作者写作本书的目的。同时建议读者从 Sun 公司的网站上下载一个最新的 Intel x86 版的 Solaris 10, 作为学习 UNIX 操作系统的运行环境。

另外, 在学习过程中, 要敢于动手, 不要怕出错, 反正只是一个学习环境, 而非正常工作的服务器, 即使系统瘫痪了也不要紧, 大不了再重新安装一次, 还能借此再重温一次安装过程。总之, 多看、勤动手是最好的学习方法。一旦熟悉了某一个 UNIX 系统, 其他 UNIX 系统也就不在话下, 无需多虑。

本书力图从 UNIX 系统的基本命令入手, 由浅入深, 逐步阐述 UNIX 操作系统的基本概念和原理, 同时给出了大量的应用实例。在此基础上, 对 UNIX 的进程管理、文件系统的内部组织结构、文件系统的管理与维护、系统的启动与关机过程、TCP / IP 网络管理、Shell 编程、软件包的制作与安装等方面, 进行深入的讨论, 使读者在学习、使用、管理和维护 UNIX 系统等方面真正能够有所收益。

除了附录介绍的图形用户界面, 书中的大部分内容主要介绍 UNIX 的共同特性, 并不涉及具体的 UNIX 系统。由于现今的 UNIX 并没有一个统一的版本, 同样的命令, 在不同的系统中, 命令的选项与输出的结果不尽相同, 这为编写一本具有普适性的 UNIX 书籍, 带来很大的问题和难度。为了尽可能地保持本书所述内容的普遍性, 在讨论过程中, 我们尽量介绍大多数系统都提供的命令和功能。在介绍具体的命令时, 也尽量不讲特定系统独有的内容, 即使很有特点的选项也是如此。我们给出的命令语法, 也只是常用的选项和几个厂家均有的内容, 而并非命令的全部。这一点需要特别予以说明。

在本书给出的例子中, 需要用户输入的命令均以黑体形式给出。其中, 命令提示符为井号“井”者, 表示只有具有超级用户身份的用户才能使用的命令, 命令提示符为美元符号“\$”者, 表示可由普通用户使用的命令。

最后还要请读者注意, UNIX 操作系统是一个庞大的系统, 功能之丰富, 信息量之大, 涵盖面之广, 非一般系统所能比拟。因此要随时注意查看随机资料, 扩充和丰富自己的知识。

本书是作者学习 UNIX 系统近 20 年的经验和体会, 如果能够使读者在学习 UNIX 操作系统时有所裨益, 将是作者莫大的荣幸。但由于时间仓促, 且限于作者的水平和能力, 如有不当甚至谬误之处, 恳请广大读者给予批评指正。

在本书的写作过程中, 从写作的宗旨与原则, 到章节的选定与编排, 都得到了电子工业出版社美迪亚公司的策划编辑责任编辑的热情鼓励和全力帮助。任正梅、赵东江、黄辰、曾伟玲、王群、曹雷、伊晓强、仇鹏涛、张秋慧、邸静和邢梦可等也给予了大力协助, 在此一并表示感谢。另外, 还要感谢本书所列参考书目的作者。

目 录

第 1 章 UNIX 概述 1	2.11 查阅联机文档 45
1.1 UNIX 早期发展过程概述..... 1	2.12 命令行的解释执行过程 46
1.1.1 UNIX 的缘起..... 2	2.12.1 读取命令行..... 46
1.1.2 UNIX 的交替发展..... 4	2.12.2 回显输入的命令..... 46
1.1.3 UNIX 的战国时代..... 5	2.12.3 变量替换..... 47
1.1.4 策略与标准之争..... 6	2.12.4 命令替换..... 47
1.1.5 UNIX 的黑暗时期..... 7	2.12.5 I/O 重定向..... 48
1.1.6 AT&T UNIX System V Release 4.0..... 7	2.12.6 IFS 处理..... 49
1.1.7 后 UNIX 时代..... 8	2.12.7 元字符扩展与文件名 生成..... 49
1.2 UNIX 的层次组织结构..... 9	2.12.8 引号引用处理..... 50
1.3 UNIX 的逻辑组织结构..... 11	2.12.9 跟踪执行过程..... 50
1.3.1 进程管理子系统..... 12	2.12.10 环境处理..... 50
1.3.2 内存管理子系统..... 13	2.12.11 执行命令..... 50
1.3.3 文件管理子系统..... 13	2.12.12 实例验证..... 51
1.3.4 I/O 管理子系统..... 14	2.13 定制 Shell 工作环境 52
1.3.5 硬件系统..... 14	2.13.1 设置环境变量..... 52
第 2 章 UNIX 系统入门 15	2.13.2 设置命令检索路径..... 53
2.1 命令行结构..... 15	2.13.3 定制命令提示符..... 53
2.2 后台进程..... 18	2.13.4 定义命令别名..... 55
2.3 标准输入、标准输出与标准 错误输出..... 19	2.13.5 设定中文运行环境..... 55
2.4 输入输出重定向..... 20	2.13.6 定制.profile 初始化文件..... 55
2.5 管道..... 25	第 3 章 文件系统基础知识 56
2.6 元字符与文件名生成..... 27	3.1 文件系统的层次结构..... 56
2.7 转义与引用..... 30	3.1.1 树形结构..... 56
2.8 命令历史..... 33	3.1.2 路径名..... 57
2.8.1 fc 命令..... 34	3.2 文件系统的组织结构..... 58
2.8.2 history 命令..... 37	3.3 文件的类型..... 59
2.8.3 重复执行先前的命令..... 38	3.3.1 普通文件..... 60
2.8.4 编辑并执行校正后的命令..... 38	3.3.2 目录文件..... 61
2.9 命令别名..... 39	3.3.3 特殊文件..... 64
2.10 作业控制..... 42	3.3.4 链接文件..... 67
	3.3.5 符号链接文件..... 68

3.3.6 管道文件.....	70	4.16.4 利用管道实现其他处理 功能	96
第 4 章 文件和目录操作	71	4.17 文件目录的安全机制.....	97
4.1 创建文件.....	71	4.17.1 显示文件的访问权限.....	97
4.2 显示文件列表	72	4.17.2 修改文件目录的访问 权限	98
4.2.1 使用 ls 命令显示文件列表	72	4.17.3 设置文件目录的访问 权限	100
4.2.2 利用通配符显示文件.....	74	4.17.4 其他访问权限设置.....	103
4.2.3 列出隐藏文件.....	76	4.18 文件内容检索	104
4.2.4 递归地列出文件.....	77	4.18.1 利用 grep 检索文件 内容	104
4.3 显示文件的内容.....	78	4.18.2 使用 grep 进行过滤.....	105
4.3.1 使用 cat 命令显示文件	78	4.18.3 使用 grep 检索多个 文件	105
4.3.2 使用 more 命令分页显示 文件	78	4.18.4 检索不包含特定字符串 的文本行.....	106
4.3.3 使用 pg 命令分页显示文件.....	80	4.18.5 在 grep 中使用正则表 达式	106
4.3.4 使用 head 命令显示文件前 几行内容.....	80	4.18.6 检索元字符本身.....	108
4.3.5 使用 tail 命令显示文件最后 几行内容.....	81	4.18.7 在命令行中使用引号.....	108
4.4 复制文件.....	81	第 5 章 编辑文件	109
4.5 移动文件.....	82	5.1 启动 vi 编辑器.....	109
4.6 删除文件.....	84	5.1.1 创建文件.....	109
4.7 确定文件内容的类型.....	85	5.1.2 状态行	110
4.8 建立链接和符号链接文件	85	5.2 vi 编辑器的两种工作模式.....	110
4.9 显示当前工作目录	87	5.2.1 输入模式.....	111
4.10 改换目录	87	5.2.2 命令模式.....	111
4.11 创建目录.....	89	5.3 保存编辑的文件并退出 vi	111
4.12 移动目录	89	5.4 vi 编辑器的基本命令	112
4.13 复制目录	90	5.4.1 移动光标位置.....	112
4.14 删除目录	91	5.4.2 输入文本.....	113
4.15 比较文件之间的差别	91	5.4.3 修改与替换文本.....	114
4.15.1 使用 diff 命令比较两个 文件.....	91	5.4.4 撤销先前的修改.....	114
4.15.2 使用 diff3 命令比较三个 文件.....	92	5.4.5 删除文本.....	114
4.15.3 使用 bdiff 命令比较大型文件	94	5.4.6 复制、删除与粘贴文本	115
4.16 检索文件	94	5.4.7 按指定的数量重复执行 命令	116
4.16.1 简单检索.....	95	5.5 使用 ex 命令	116
4.16.2 使用逻辑运算符.....	95		
4.16.3 利用 find 命令本身实现 其他处理功能.....	96		

5.5.1 显示行号.....	116	优先级	138
5.5.2 多行复制.....	117	6.5.3 修改分时进程的优先级.....	139
5.5.3 移动文本行.....	117	6.5.4 修改进程的调度类别.....	140
5.5.4 删除文本行.....	117	6.6 调整分时进程的优先级	141
5.6 检索与替换.....	117	6.7 定时运行系统任务和用户 程序.....	143
5.6.1 检索字符串.....	118	6.7.1 cron 监控进程的调度 过程	143
5.6.2 模式检索	118	6.7.2 调度定时重复执行的 任务	145
5.6.3 替换字符串.....	119	6.7.3 提交一次性定时执行的 任务: at.....	145
5.7 编辑多个文件	120	6.8 调度定时重复执行的任务.....	146
5.7.1 编辑多个文件.....	120	6.8.1 crontab 文件的工作原理.....	146
5.7.2 合并文件与合并文本行	120	6.8.2 crontab 文件的语法格式.....	147
5.8 定制 vi 编辑器的运行环境	120	6.8.3 创建和编辑 crontab 文件.....	147
5.8.1 临时设定 vi 的运行环境	120	6.8.4 显示 crontab 文件.....	149
5.8.2 永久性地定制 vi 的运 行环境.....	122	6.8.5 删除 crontab 文件.....	150
5.9 其他特殊说明	123	6.8.6 crontab 命令的访问控制.....	151
5.9.1 删除或替换特殊字符	123	6.8.7 应用实例——数据库 定时备份.....	153
5.9.2 在编辑期间运行 UNIX 命令.....	123	6.9 调度执行单个作业或任务	155
5.10 vi 编辑器命令总结.....	124	6.9.1 创建 at 作业.....	155
第 6 章 进程管理	127	6.9.2 显示 at 作业及作业队列.....	156
6.1 ps 命令概述.....	128	6.9.3 删除 at 作业.....	157
6.2 列出进程及其状态信息	130	6.9.4 at 命令的访问控制.....	157
6.2.1 列出经常请求运行 的进程.....	130	6.9.5 应用实例——系统定 时关机	159
6.2.2 列出系统中的所有进程.....	130	第 7 章 用户与用户组	161
6.2.3 列出进程的重要状 态信息.....	131	7.1 增加与删除用户	161
6.2.4 列出进程的详细状 态信息.....	132	7.1.1 /etc/passwd 文件.....	162
6.2.5 利用 ps 命令监控异 常进程.....	132	7.1.2 /etc/shadow 文件.....	163
6.3 监控进程及系统资源	132	7.1.3 用户管理实例.....	164
6.4 强行终止进程的运行	135	7.2 定制用户的工作环境	168
6.5 调整进程的调度类别及优 先级	137	7.2.1 选择命令解释程序.....	168
6.5.1 显示进程调度的类别及 优先级.....	137	7.2.2 设置用户初始化文件.....	171
6.5.2 调整进程的调度类别与		7.3 增加与删除用户组.....	176
		7.4 监控用户	179
		7.4.1 使用 who 命令查询系统中	

的用户.....	179	9.4 while 循环.....	240
7.4.2 使用 finger 命令查询系统 中的用户.....	180	9.5 until 循环.....	245
7.4.3 使用 whodo 命令查询系统 中的用户活动.....	181	9.6 select 循环.....	246
7.4.4 向注册用户发送消息.....	181	9.7 嵌套的循环.....	248
第 8 章 Shell 基础知识	183	9.8 循环控制.....	250
8.1 引言.....	183	9.8.1 break 和 continue 命令.....	250
8.1.1 为什么需要 Shell 编程.....	184	9.8.2 true 命令.....	252
8.1.2 什么是 Shell 脚本.....	185	9.8.3 sleep 命令.....	252
8.1.3 运行 Shell 脚本.....	185	9.8.4 shift 命令.....	253
8.1.4 退出与出口状态.....	186	9.8.5 getopt 命令.....	254
8.1.5 调用适当的 Shell 解 释程序.....	188	9.8.6 getopt 命令.....	256
8.1.6 位置参数.....	189	9.9 循环结构代码块的 I/O 重定向..	258
8.2 变量与变量替换.....	192	9.9.1 while 循环的 I/O 重定向.....	259
8.2.1 变量分类.....	193	9.9.2 until 循环的 I/O 重定向.....	260
8.2.2 变量的赋值.....	193	9.9.3 for 循环的 I/O 重定向.....	261
8.2.3 内部变量.....	194	9.10 Here 文档.....	262
8.2.4 变量的引用与替换.....	196	9.11 函数.....	268
8.2.5 变量的间接引用.....	198	9.12 并列结构.....	276
8.2.6 特殊的变量替换.....	199	9.12.1 逻辑与并列结构.....	276
8.2.7 变量声明与类型定义.....	203	9.12.2 逻辑或并列结构.....	278
8.3 命令与命令替换.....	204	9.13 数组.....	278
8.3.1 Shell 内部命令.....	204	9.14 信号捕捉处理.....	285
8.3.2 部分命令介绍.....	207	9.15 其他 Shell 课题.....	290
8.3.3 命令替换.....	216	9.15.1 子 Shell.....	290
8.4 test 语句.....	218	9.15.2 进程替换.....	292
8.4.1 文件测试运算符.....	219	9.15.3 调试.....	293
8.4.2 字符串测试运算符.....	220	9.15.4 系统性能考虑.....	298
8.4.3 整数值测试运算符.....	222	第 10 章 磁盘空间管理	300
8.4.4 逻辑运算符.....	223	10.1 磁盘空间管理.....	300
第 9 章 Shell 高级编程	226	10.1.1 常用的磁盘空间管理 工具.....	300
9.1 if 语句.....	226	10.1.2 使用 df 命令检查磁盘 空间的使用情况.....	301
9.1.1 if 语句的表现形式.....	226	10.1.3 使用 du 命令检查目录占 用的存储空间.....	304
9.1.2 嵌套的 if/then 条件测试.....	229	10.1.4 使用 quot 命令查询每个 用户占用的存储空间.....	306
9.1.3 if/then 结构参考.....	231	10.1.5 使用 find 命令找出超过一 定容量限制的文件.....	307
9.2 case 控制结构.....	233		
9.3 for 循环.....	236		

10.1.6 使用 find 命令找出并删除 长期闲置不用的文件.....308	件包358
10.1.7 使用 find 命令找出并删除 core 文件.....310	11.3.5 pkgtrans 命令364
10.1.8 使用 ls 命令检测文件的 大小.....311	11.4 安装软件包365
10.1.9 清除临时目录或文件.....311	11.5 查询软件包369
10.2 利用 UNIX 提供的标准 工具进行备份.....312	11.6 检测软件包370
10.2.1 使用 cpio 执行备份和 恢复.....312	11.7 卸载软件包373
10.2.2 使用 tar 执行备份和 恢复.....320	第 12 章 TCP/IP 网络通信.....377
10.2.3 使用 dd 命令实现文件系 统的原样复制.....325	12.1 TCP/IP 简介377
10.3 限额控制328	12.1.1 TCP/IP 协议的层 次结构.....377
10.3.1 限额概述.....328	12.1.2 TCP/IP 协议如何处理 数据通信.....379
10.3.2 设置限额.....330	12.2 设置网络接口382
10.3.3 限额的维护.....332	12.3 主机名字解析385
第 11 章 软件包的制作与管理338	12.4 设置网络路由385
11.1 软件包组成简介338	12.4.1 静态路由.....386
11.1.1 基本组成部分.....339	12.4.2 动态路由.....388
11.1.2 选用的信息文件.....339	12.5 配置网络服务.....390
11.1.3 选用的 Shell 脚本文件.....339	12.6 TCP/IP 网络管理.....392
11.2 软件包的相关文件和命令339	12.6.1 使用 ifconfig 命令管理接 口配置.....392
11.2.1 pkginfo 文件340	12.6.2 使用 netstat 命令监控 网络状态.....393
11.2.2 prototype 文件341	12.6.3 使用 ping 命令测试远程主 机的连通性.....398
11.2.3 pkgmap 文件.....344	12.6.4 启动路由监控进程的日 志功能.....399
11.2.4 copyright 文件345	12.6.5 使用 traceroute 命令跟踪 路由信息.....400
11.2.5 depend 文件346	12.7 监控网络性能401
11.2.6 space 文件.....346	12.7.1 使用 ping 命令检测网络 主机的响应能力.....401
11.2.7 compver 文件.....347	12.7.2 使用 ftp 命令检测网络主 机的传输性能.....402
11.2.8 软件包的相关工具.....347	12.7.3 使用 spray 命令检测网络 主机的可靠性和传输性能.....403
11.3 制作软件包347	12.7.4 通过 netstat 命令检测网 络主机的可靠性和传输 能力404
11.3.1 制作软件包的步骤.....348	
11.3.2 创建 pkginfo 文件354	
11.3.3 利用 pkgproto 命令创 建 prototype 文件355	
11.3.4 利用 pkgmk 命令制作软	

12.8 检测网络问题	406	14.2.5 /etc/inittab 文件举例	448
12.8.1 一般网络故障修复技巧	406	14.2.6 修改 inittab 文件	452
12.8.2 基本的网络软件检查	406	14.3 用户注册进程	452
12.9 网络应用——Telnet	407	14.3.1 用户注册进程的启动 过程	452
12.10 网络应用——FTP	410	14.3.2 sac 进程与 _sactab 文件	454
12.10.1 连接 FTP 服务器	413	14.3.3 ttymon 进程与 ttydefs 文件	455
12.10.2 ftp 应用	414	14.3.4 login 进程与 passwd 文件	455
12.10.3 ftp 访问控制	416	14.3.5 Shell 进程与 profile 文件	455
12.10.4 ftp 自动注册	416	14.3.6 utmpx 和 wtmpx 文件	455
第 13 章 网络文件系统	418	14.4 系统的关机过程	456
13.1 NFS 的体系结构	418	14.4.1 使用 shutdown 命令关闭 系统	456
13.2 设置 NFS 服务器	419	14.4.2 使用 init 命令关闭系统	457
13.3 共享本地文件系统	420	14.5 应用实例	458
13.4 设置 NFS 客户机	423	第 15 章 文件系统内部组织	461
13.5 安装远程文件系统	424	15.1 文件系统的组织结构	461
13.6 NFS 性能监测	426	15.2 超级块	464
13.6.1 服务器性能监测	426	15.3 信息节点	465
13.6.2 客户机性能监测	429	15.3.1 特权标志位	466
13.7 NFS 故障修复策略	431	15.3.2 数据块地址数组	467
13.8 NFS 故障修复步骤	432	15.4 数据区与空闲数据存储块 的组织	468
13.8.1 检查 NFS 客户机的连接 状态	432	15.5 信息节点的分配与释放	469
13.8.2 检测远程 NFS 服务器	432	15.6 数据块的分配与释放	470
13.8.3 检验服务器上的 NFS 服务	434	15.7 信息节点与目录和文件的 关系	473
13.8.4 重新启动 NFS 服务	435	15.8 UFS 文件系统	473
13.8.5 重新启动 rpcbind	436	15.8.1 UFS 文件系统的组织 结构	473
13.8.6 查询提供 NFS 文件共享 服务的主机	436	15.8.2 引导块	474
第 14 章 系统的启动与关机过程	437	15.8.3 超级块	474
14.1 UNIX 系统的初始引导过程	437	15.8.4 柱面组信息块	475
14.1.1 引言	437	15.8.5 信息节点区与信息节点	477
14.1.2 系统的初始引导过程	440	15.8.6 数据块区	479
14.1.3 系统初始化	441	15.8.7 UFS 数据块的分配与释	
14.2 系统生成与 init 进程	443		
14.2.1 运行级	444		
14.2.2 改变运行级	446		
14.2.3 /etc/inittab 文件	446		
14.2.4 处理方式	447		

放过程.....	479	A.2.1 硬件要求.....	528
15.8.8 UFS 信息节点的分配与 释放过程.....	481	A.2.2 安装前的准备工作.....	528
15.8.9 定制 UFS 文件系统参数.....	481	A.2.3 Solaris 操作系统的安 装步骤.....	532
第 16 章 文件系统管理	484	附录 B 通用桌面环境简介	543
16.1 创建文件系统.....	485	B.1 注册到 CDE 桌面系统.....	543
16.1.1 创建 s5 文件系统.....	485	B.1.1 注册.....	544
16.1.2 使用 mkfs 命令创建 UFS 文件系统.....	485	B.1.2 在会话期间选用不同的 语言.....	545
16.1.3 使用 newfs 命令创建文件 系统.....	490	B.1.3 使用命令行会话.....	545
16.2 使用 labelit 命令命名文件 系统.....	492	B.1.4 退出系统.....	546
16.3 安装、卸载文件系统.....	492	B.1.5 恢复主会话.....	546
16.3.1 安装文件系统概述.....	492	B.2 CDE 桌面系统	546
16.3.2 /etc/vfstab 文件.....	494	B.2.1 CDE 桌面.....	546
16.3.3 安装文件系统.....	495	B.2.2 CDE 工作区.....	547
16.3.4 卸载文件系统.....	500	B.2.3 前面板.....	547
16.4 确定文件系统的类型.....	502	B.2.4 工作区菜单.....	549
16.4.1 使用 vfstab 文件确定文 件系统的类型.....	502	B.3 文件管理器	550
16.4.2 使用 fstyp 命令确定文件 系统的类型.....	503	B.4 运行应用程序	551
16.5 检测与修复文件系统.....	505	B.4.1 前面板中运行应用.....	551
16.5.1 何时需要检测文件系统.....	505	B.4.2 从 CDE 工作区菜单中运行 应用.....	551
16.5.2 文件系统检测的内容.....	507	B.4.3 从应用管理器中运行 应用.....	551
16.5.3 交互地检测与修复 UFS 文件系统.....	510	B.4.4 从文件管理器中运行 应用.....	552
16.5.4 自动地检测与修复 UFS 文件系统.....	513	B.4.5 从终端窗口运行应用.....	552
16.5.5 解决 fsck 命令无法修复 的 UFS 文件系统问题.....	514	B.5 应用管理器	552
16.5.6 恢复严重受损的超级块.....	514	B.6 式样管理器	553
16.5.7 fsck 的阶段处理方式.....	516	B.7 文本编辑器	553
16.5.8 利用其他工具修复文件 系统.....	525	B.8 终端窗口	554
附录 A 安装 UNIX 操作系统	528	B.9 移动介质管理器	555
A.1 下载 Solaris 操作系统.....	528	B.10 管理注册服务器	555
A.2 安装 Solaris 操作系统.....	528	B.10.1 启动注册服务器.....	556
		B.10.2 停止注册服务器.....	556
		B.11 定制注册界面	557
		B.11.1 定制注册屏幕的外观表 现形式.....	558
		B.11.2 指定 CDE 桌面会话过程	

中使用的默认语言	560
B.11.3 限定注册界面语言选择	
菜单中的语言列表	560
B.12 实现本地化的 CDE 桌面	
会话	561
B.12.1 设置 LANG 环境变量	561
B.12.2 设置其他环境变量	563
B.12.3 app-defaults 资源文件的	
本地化	563
B.12.4 背景的本地化	563
B.12.5 调色板的本地化	563
B.12.6 帮助文件的本地化	564
B.12.7 NLSPATH 变量与消息	
分类文件的本地化	564
B.12.8 使用指定的语言创建或	
编辑文件	564
B.12.9 使用指定的语言执行终	
端仿真程序	565
B.12.10 使用指定的字体	565

第1章 UNIX 概述

UNIX 之所以能够取得如此巨大的成功，这与其早期的发展历史有着极大的关系。作为开始，本章将首先简单地介绍 UNIX 早期的发展过程，以及 UNIX 版本的变化；从总体上直观地介绍一下 UNIX 的层次组织结构与逻辑组织结构；简述 UNIX 核心的四个主要组成部分，使读者对 UNIX 有一个全面的印象。本章讨论的主要内容包括：

- UNIX 发展过程概述
- UNIX 的层次组织结构
- UNIX 的逻辑组织结构

1.1 UNIX 早期发展过程概述

UNIX 之所以能够取得如此巨大的成功，与 UNIX 早期的发展历史及其功能特点有着极大的关系，概括起来主要有以下三个原因：

- 简洁性。UNIX 最初的成功其主要的原因并不在于它在技术上有多先进，而是由于 UNIX 对操作系统概念和技术的极大简化。以当时的硬件技术水平和软件实践而言，这确实是一项明智之举。
- 开放性。由于受与美国联邦政府司法部签订的法令限制，AT&T 公司当时不能介入计算机及其相关技术产品的商业活动，使得 UNIX 成为一个公开的、并非与硬件一起捆绑销售的软件产品。贝尔实验室向大学和研究机构公开其源代码，供教学和研究用。贝尔实验室的“不宣传，不支持，不负责纠错”的策略也促使 UNIX 的研究人员互帮互助，互相交流经验，最终导致 UNIX 用户组的成立。
- 可移植性。UNIX 系统的简洁性和开放性，吸引了许多人的关注，进而产生了极大的兴趣，使许多人相信可以把 UNIX 移植到自己的计算机上，籍此建立一个自己的操作系统和开发环境，提供一个自助的研究、开发的编程环境。而 C 语言的可移植性也使 UNIX 系统的移植成为可能。

此外，下述四个事件（或因素）也是 UNIX 能够取得巨大的成功的重要原因：

- C 语言。C 语言的出现与成功反过来也促进了 UNIX 系统的成功。UNIX 与 C 语言相辅相成，互为依托，使得 UNIX 系统成为有别于其他操作系统的一种独特发展现象。使用 C 语言编写操作系统既提高了操作系统的开发效率，又易于阅读和理解，更有利于把 UNIX 移植到不同的硬件平台上。
- TCP/IP。TCP/IP 在 UNIX 系统中的出现使得 ARPANET 以及后来的 Internet 成为现实。同时，Internet 和 TCP/IP 的日益流行也促使 UNIX 系统得到更广泛的应用，从而促进了 UNIX 系统的蓬勃发展。
- C 语言标准 I/O 库。C 语言标准 I/O 库的出现进一步奠定了 UNIX 系统可移植性的基础。

使得系统开发人员完全可以抛开底层硬件的实现细节，只需专注于高层的功能实现，缩短了操作系统研发的周期。

- 向大学和研究机构公开源代码。向大学和研究机构公开源代码，其效果有二：前期激发了对 UNIX 系统研究和移植的兴趣，导致 UNIX 成为操作系统的新宠；及至后期，UNIX 成为大学操作系统课程的代名词。许多大学均以 UNIX 作为操作系统课程的研究对象，从而出现了“UNIX 操作系统设计”等著名的 UNIX 教材。同时也培养了许多潜在的 UNIX 系统准用户。

1.1.1 UNIX 的缘起

1965 年，麻省理工学院（MIT）和通用电器公司（GE）开始联合开发一个称之为 Multics 的新型操作系统。其设计目标是开发一个多用户、并发访问、交互式的分时操作系统，具有足够强的计算和数据存储能力，使用户能够方便地共享数据。按当时主要以批处理为主的操作系统技术水平而言，这确实是一项雄心勃勃的计划。不久之后，AT&T 贝尔实验室的计算科学研究中心也加入了这个联合开发项目。

经过多年的努力，到 1969 年初，Multics 系统的初始版本开始在 GE-645 计算机上投入运行，可以同时支持三个用户。但与原来的设想和计划相距甚远，既没有提供预期的通用计算服务，也不清楚什么时候才能达到其原先的设计目标。鉴于 Multics 项目前途渺茫，贝尔实验室决定退出这一工程项目。

一切都始自 1969 年，一个闷热而潮湿的夏天，而且始自一个游戏程序。

在开发 Multics 系统期间，Ken Thompson 在一个运行 GECOS 操作系统的 Honeywell 635 计算机上，用 FORTRAN 语言编写了一个“空间旅行”游戏程序，仿真太阳系中各行星的相对位置和运行轨迹，并设计了一艘宇宙飞船，使之穿梭飞行于各行星之间。

由于结束了 Multics 的研发工作，贝尔实验室计算科学研究中心的技术人员失去了“方便的交互计算服务”。为此，Ken Thompson、Dennis Ritchie 和其他曾参与过 Multics 项目的贝尔实验室同事开始尝试改善自己的编程环境。在此期间，Ken Thompson 萌发了编写一个多用户文件系统的想法，以便每个人都能够按一定的目录层次组织和存储自己的文件。经与 D.Ritchie 和 Rudd Canaday 讨论，最终在纸上草拟了一个文件系统的设计方案。不久以后，Thompson 编写了一个内核和一个模拟其文件系统设计方案的仿真程序，并在 GE-645 计算机上实现了这个文件系统。

这个新的文件系统使 Thompson 能够以类似于 Multics 中的层次目录结构存储他开发的游戏程序源文件。但 Thompson 的“空间旅行”游戏程序在 GE-645 机器中的运行结果却并不令人满意。GE-645 是一个分时系统，忽停忽动的响应速度，使得 Thompson 很难控制飞船的飞行。由于运行结果不理想，且运行费用太贵（运行一次游戏需要花费 75 美元，这在当时是一笔不小的开支），促使 Thompson 想尽快寻找一个替代系统。

很快，Thompson 在一个角落里找到了一台几乎无人使用的 PDP-7 计算机，这是一台 16 位字长的机器，具有 16KB 内存，并配有一台图形显示终端。对于 Thompson 来讲，这显然是一个好得不能再好的替代品了。但由于需要借助于 GECOS 的开发环境，在 GE-645 与 PDP-7 之间进行交叉编译，并需要使用纸带在两台计算机之间交换数据，因而极感不便。

不久之后，Thompson 完成了其游戏程序的开发。为了建立一个更好的开发和运行环境，Thompson 和 Ritchie 开始实现其原先设计并经过仿真验证的文件系统。

由此，催生了初始的 UNIX 系统。尽管当时这个新的小型操作系统尚没有名字，但其中却包含一个小型的系统内核、早期版本的 UNIX 文件系统、进程管理子系统和内存管理子系统，提供一个小型的命令解释程序（当时尚未出现 Bourne Shell）、一组有限的实用程序和文件系统操作命令，以及可以同时支持两个用户的分时系统环境，并且第一次引入了信息节点的概念。

在此期间，GE-645 一直用作开发环境，在两台机器之间进行交叉编译，但很快便在 PDP-7 上开发了汇编程序。经过不断完善，这一新的系统最终不再需要 GECOS 作为开发环境而能够自我支持。

UNIX 的名字来自贝尔实验室另外一名参与 UNIX 开发的研究人员 Brian Kernighan。作为 Multics (Multiplexed Information and Computing System) 系统的双关语，他把这一支持两个用户的小型操作系统取名为 UNICS (Uniplexed Information and Computing System)。不久之后，UNICS 又进一步演变成现在熟悉的 UNIX。

虽然新生的 UNIX 系统受到很多人赏识，但直至得到实际应用时，其潜力才被人们所认知。UNIX 系统的第一个真正用户是贝尔实验室的专利部门，作为文档管理系统为之提供文字处理功能。但此时却存在一个问题，由于 PDP-7 是一台借来的机器，机器性能也不强，很难满足计算机研究小组不断增长的需求。到了 1971 年，第一个汇编版本的 UNIX 系统开始移植到 PDP-11/20 机器（56KB 内存，两个 2.4MB 的硬盘）上。此时，UNIX 可以支持较多的用户了，同时也增加了一个称作 roff 的文本格式处理系统。

但直至 1971 年 11 月，才真正公布了 UNIX 系统第一版，并在文档上开始正式命名为 UNIX。同时也提供了第一版的“UNIX 程序员手册”（作者 K.Thompson 和 D.M.Ritchie）。其中包括 60 多个命令，如 boot（重新引导系统）、cat（显示文件内容）、chdir（改换工作目录）、chmod（修改文件访问权限）、chown（修改文件属主）、cp（复制文件）、ls（列举目录内容）、mv（移动或重新命名文件）、wc（计数文件的字符、字和行数）和 who（查询系统中的用户）。

在获得了早期的成功之后，Thompson 准备为 UNIX 开发一个 FORTRAN 编译器，但却由此催生了一个称之为 B 的解释语言（简化版的 BCPL 语言——Basic Combined Programming Language）。

1972 年 6 月，采用 B 语言重写的 UNIX 内核增加了管道功能，用于连接一个程序的输出与另一个程序的输入，以便组合各种命令，构建新的功能。从而出现了 UNIX 第二版。但由于受 B 语言的限制（数据没有类型之分，仅支持单一机器字的数据类型，其他数据类型只能通过函数或特殊的操作符实现），且由于 B 是一种解释性的语言，使系统的性能受到了影响。

于是，擅长编程语言的 Ritchie 则在 B 语言的基础上开发了一种新的 C 语言。1973 年 2 月，Thompson 和 Ritchie 利用 C 语言重写了 UNIX 操作系统。尽管这次重写并没有功能性的重大变革，但由于使用高级语言编写操作系统，改变了依赖汇编语言开发操作系统的历史，使 UNIX 系统具有了可移植性的条件，因而对 UNIX 的发展有着极其重大的意义和影响。

此时，UNIX 在贝尔实验室内部的安装数量已经增加到 25 个。因而成立了一个 UNIX 系统小组，为贝尔实验室提供内部支持。在此期间，又于 1973 年 11 月和 1974 年 6 月，相继推出了 UNIX 的第四版和第五版。

1974 年 7 月，Thompson 和 Ritchie 在一份计算机科学年鉴——《ACM 通讯》上公开发表了 UNIX 操作系统，第一次把 UNIX 公诸于世。这篇论文的发表标志着 UNIX 时代的开始，使得 UNIX 系统日益流行，世界各地的研究机构也开始关注这一新的操作系统，从而促使贝尔实

验室开发出 UNIX 第五版。

由于 1956 年与美国联邦政府签署的法令的限制，AT&T 不能销售计算机及其软件产品。由于这个法令，AT&T 既不能登广告为 UNIX 做宣传，又不能在市场上销售，也无法支持这一系统。尽管如此，UNIX 系统还是越来越流行。

于是，贝尔实验室开始向大学免费提供 UNIX，以便用于教学。其中包括加州大学的伯克利分校和澳大利亚悉尼的新南威尔士大学。同时，UNIX 系统在电话营运公司也十分流行，安装的数量不断增长。因为 UNIX 为程序开发、网络通信提供了良好的运行和开发环境。

1975 年 5 月，UNIX 第六版开始发行并通过西电（WE）公司第一次向贝尔实验室之外的用户提供 UNIX。此时，UNIX 的代码绝大部分都是用 C 语言编写的，而在当时，为了提高计算机硬件的性能，大多数操作系统都是采用汇编语言编写的，几乎没有听说操作系统能够用高级语言开发。

因此，C 语言很快成为众所周知的可以代替汇编语言的系统开发工具。由于 C 语言编译器很小，很容易移植到各种计算机硬件系统中。另外，由于 C 语言的丰富功能，易于阅读和学习，高效的运行速度，使得系统程序员很快就放弃了使用汇编语言开发系统程序。

1.1.2 UNIX 的交替发展

1. BSD 版的 UNIX

1975 年，Thompson 作为一个访问学者回到其母校——加州大学伯克利分校，在学校的 PDP-11/70 计算机上安装了 UNIX 第六版，开始进行 UNIX 的教学和研究工作。几乎与此同时，Bill Joy 和 Chuck Haley 作为研究生也来到伯克利。从此拉开了 BSD 版 UNIX 的序幕。

此后，UNIX 的发展开始变得比较复杂。有时，贝尔实验室增加了某个新的特性，然后以源码的形式对外分发。伯克利分校改善或增加了另一个新的特性，这个新的特性最终又合并到贝尔实验室的下一个版本中，使得 UNIX 系统的功能不断地得到提升和广泛应用。

随着 Thompson 及其他贝尔实验室成员在伯克利分校教授 UNIX，使许多学生和教授对 UNIX 产生了极大的兴趣。Bill Joy 把在此期间开发的 Pascal 编译器、ex（一个新的行编辑器）等软件，连同移植的 UNIX V6 捆绑到一起，称之为 BSD（Berkeley Software Distribution），从 1977 年的年底开始向外分发。而 1978 年推出的 2BSD 则是重写 UNIX V6 的重大尝试，其中包括了 csh、mail，以及 Bill Joy 开发的著名 vi 编译器和 termcap 等。

后来，伯克利分校计算机科学系又开始把 UNIX V7 移植到 32 位的 VAX 机器上，为 UNIX 的功能扩展铺平了道路。同时在 UNIX 内核中引入了页面调度和虚拟内存管理的概念，从而导致 BSD 3.0 的出现。

1981 年 6 月，伯克利又推出了 4BSD，其中包括快速文件系统、作业控制、可靠的信号处理，以及自动重新引导系统等功能。UNIX 的这些功能特征，促使美国国防部高级研究项目局（DARPA）对 VAX 中的 UNIX 系统的内存管理，以及 VAX 自己的操作系统 VMS 进行评估，并投资在 UNIX 系统中作进一步的研究；尤其在网络通信方面，更是作为研究的重点，从而导致在 1981 年 6 月推出了 4.1BSD，增加了自动配置功能。并在第二年的 4 月首次开发出 ARPANET 网站的测试版，因而在 UNIX 中实现了著名的 TCP/IP 和套接字。从此时开始，UNIX 的发展进入了一个关键的起点。