



UNIX应用教程

杨华中 编著

人民邮电出版社



UNIX 应用教程

杨 华 中 编著

人 民 邮 电 出 版 社

UNIX 应用教程

杨华中 编著

责任编辑 滑玉

*

人民邮电出版社出版发行
北京崇文区夕照寺街 14 号
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

*

开本: 787×1092 1/16 1996 年 9 月 第 1 版
印张: 13.75 1996 年 9 月 北京第 1 次印刷
字数: 328 千字 印数: 1—5 000 册

ISBN 7-115-06214-5/TP·336

定价: 19.00 元

内容提要

本书重点介绍 UNIX 的基本概念、使用方法和程序设计技巧。主要内容分两部分。第一部分由浅入深地介绍 UNIX 的基本功能、常用命令的使用方法和系统维护；第二部分主要介绍 UNIX 为程序设计开发者提供的各种实用工具，同时还提供关于进程通信、共享内存和网络编程的若干实例。

本书适合于 UNIX 普通用户、软件开发人员和系统管理人员阅读，并可供大学师生和各种专业技术人员参考。

UNIX 经历了 20 余年的发展, 目前已成为应用最广、影响最大的操作系统, 在科学计算、工程应用、网络通信、事务处理和教学科研等各领域均取得了辉煌的成就。在国内, UNIX 曾经被视为学院派, 但对于需要通过 InterNet 来获取和传播信息的普通用户而言, 会使用 UNIX 操作系统是十分必要的, 因为, InterNet 上提供各种服务的计算机大多运行 UNIX 操作系统。由此可见, UNIX 必将随着计算机的家庭化和 InterNet 的普及而进入寻常百姓之家。

本书是为具有一定计算机知识的读者而编写的, 不论是否接触过 UNIX 系统, 只要需要使用 UNIX 系统, 总可以从本书中找到所需的内容。UNIX 的版本很多, 而且也各有其特点, 目前最流行的版本有系统 V 和 BSD 4. x 系列。本书介绍的内容虽然基于系统 V 和 BSD 4. x 系列, 但也适合于别的 UNIX 版本, 这使得本书特别适合于普通读者。

本书力求内容浅显、通俗易懂, 尽量避免使用 UNIX 中过于抽象的概念和词汇; 本书还尽量和 DOS 进行对比, 以便于熟悉 DOS 的广大读者能快捷地学会使用 UNIX 系统。

随着微电子技术、EDA 工具和计算机网络的发展, 工程技术人员越来越需要掌握 UNIX 的应用方法和 C 程序设计技术, 为此, 我们从 1995 年开始, 为清华大学电子工程系的研究生和高年级本科生开设了《C/UNIX 程序设计》课程, 本书便是根据该课程的讲义改编而成的。如果读者能通过本书掌握 UNIX 的基本技能并用于解决实际问题, 这将是作者最欣慰的事。

在《C/UNIX 程序设计》的课程设置、讲义形成和本书的编写等诸多方面曾得到清华大学电子工程系系主任董在望教授, 副系主任汪蕙、刘序明教授以及朱雪龙教授的热心关怀, 特别是汪蕙副系主任和朱雪龙教授, 他们还对讲义的具体内容提了许多宝贵意见, 谨借此机会向他们表示由衷的谢意; 最后还要感谢王志华博士和王永同志对本书的支持。

作者

1996 年 1 月

于清华园

引 言

UNIX 是一个分时、多用户、多任务、具有网络通信功能和可移植性的操作系统。UNIX 自 1969 年诞生以来,经历了 20 余年的发展历程;今天,UNIX 已广泛移植在微型计算机、小型计算机、工作站、大型计算机和巨型计算机上,成为应用最广、影响最大的操作系统,在科学计算、工程应用、网络通信、事务处理和教学科研等领域均取得了辉煌的成就。

UNIX 提供了丰富的软件开发工具和 200 多个实用程序;有一个功能极强的 Shell 命令解释程序,它既能为用户提供了友好的命令界面,又可以用于程序设计;UNIX 上的 X 窗口系统有两大流行的风格,OPEN LOOK 和 Motif,它们都能提供十分友好的图形用户界面(GUI)。而且,没有经验的用户也可以在 UNIX 上轻松自如地完成 GUI 应用程序的开发;UNIX 提供的 lex、yacc、awk、make、lint 和 sdb 等程序开发工具更是让程序员如虎添翼;UNIX 所提供的网络通信软件包可使你加入 InterNet 的行列:发送和接收电子邮件,免费拷贝有价值的计算机软件,阅读世界各地的电子书刊、新闻杂志,参与全球范围的学术讨论,从事娱乐、购物等多种活动;还可以在 UNIX 系统上创建自己的首页(Home Page),通过 InterNet 将信息提供给无数的使用者。UNIX 及其应用程序大多是用 C 语言书写的,便于移植;具有极大的通用性、灵活性和可扩充性。

UNIX 的主要流派包括 UNIX 系统 V、BSD 4.x 和 XENIX 等,虽然其版本众多,但它们的设计思想、常用命令和实用程序并没有什么区别。本书介绍的内容适合于当今的各种 UNIX 版本,都是实际工作中常用的命令和使用技巧。至于那些很深入但又很少使用的命令,读者可在学习本书的基础上借助 UNIX 的 man 命令进行自学。

本书力求内容浅显、通俗易懂,尽量避免使用 UNIX 中过于抽象的概念和词汇。考虑到国内许多需要学习和应用 UNIX 的读者大部分都会使用 DOS 操作系统,为便于读者理解、掌握 UNIX 系统,本书在介绍 UNIX 的内容时尽量和 DOS 进行对比。熟悉 DOS 的读者应特别注意:UNIX 系统严格区分字母的大小写,在阅读本书或在 UNIX 上工作时都应特别注意这一点。

本书共十二章,可分为两大部分。前九章为第一部分,介绍 UNIX 的基本概念、常用命令和各种实用程序的使用方法和技巧。这些内容主要包括:第一章,介绍 UNIX 的基本特点;第二章,介绍如何建立与 UNIX 系统的联系,如何应用 UNIX 的联机手册进行自学;第三章,介绍 UNIX 的文件系统及相关命令;第四章,介绍 UNIX 的全屏幕正文编辑程序 vi;

第五章,介绍 UNIX 的命令解释程序 Shell 和 C Shell 环境;第六章,介绍 UNIX 的网络应用程序,如电子邮件 mail 和网络文件传输 ftp;第七章,介绍 UNIX 系统上的 X 窗口系统;第八章,介绍 UNIX 的系统维护知识,包括网络环境配置、文件系统备份和系统管理员的日常职责;第九章,介绍 UNIX 的模式扫描与匹配语言——awk 语言。本书的后三章为第二部分,介绍 UNIX 的程序设计技术。其中:第十章介绍 UNIX 程序设计环境;第十一章介绍 UNIX 的各种系统调用;第十二章介绍 UNIX 的进程间通信及编程技巧,这是在 UNIX 系统上进行程序设计的核心,本书在介绍相关的程序设计思想时还给出了许多实用程序供读者模仿。

为了使叙述简洁明了,也为了便于读者阅读和使用,本书采用了下述约定规范:

1. UNIX 的术语

凡第一次出现的 UNIX 术语,本书一律用醒目的粗体表示,并在后面的括号中给出对应的英文,以便于读者阅读 UNIX 的联机手册,例如,**进程(process)**、**管道(pipe)**等。

2. 键和字符

从键盘上输入的键一律用粗体字符,例如,**R**eturn、**l**s、**C**ontrol-**D**等;而相应的字符则用非粗体表示,如 ls 等;用“-”连接的两个键表示一个复合键,即:必须在按下第一个键的同时按第二个键,例如:**C**ontrol-**D**表示按 **C**ontrol 键的同时,再按 **D** 键。这里,**D** 虽然是大写,但并不需要按 **S**hift 键,也就是说,复合键中的字母键是不区分大小写的。

3. 提示符、UNIX 的输出和回车

UNIX 系统给出的提示符和输出均用非粗体表示,如,login;等,而本书中,省略了 UNIX 的命令提示符 \$、或 %、或用户喜欢的字符串,所有的回车符一律省略。

4. 命令

本书中的命令都另起一行,并用灰衬底表示,例如:

```
ls -al
```

表示一个命令,如果需要在计算机上执行该命令,则必须依次输入各字符(大小写必须完全一致),然后再输入一个回车符(键盘上的 **E**nter 或 **R**eturn 键)。

目 录

引 言	1
第一章 UNIX 操作系统简介	1
1.1 UNIX 操作系统的发展历程	1
1.2 UNIX 的系统结构及主要特点	2
1.3 从用户角度看 UNIX 操作系统	4
1.3.1 文件系统	5
1.3.2 命令解释器 Shell 和 C Shell	6
1.3.3 命令	6
1.3.4 组合出效率	8
1.4 使用 UNIX 的注意事项	8
第二章 初学者入门	9
2.1 准备进入 UNIX 系统	9
2.2 建立与 UNIX 系统的联系	10
2.2.1 注册	10
2.2.2 改变口令字、显示当前的时间和日期	11
2.2.3 改正输入命令行中的错误	11
2.2.4 终止正在运行的程序	12
2.2.5 注销	12
2.3 如何配置用户环境	12
2.3.1 用命令 stty 设置终端特性	12
2.3.2 设置环境变量	13
2.3.3 设置别名	13
2.4 如何与别的用户通信	15
2.5 了解系统的工作状况	16
2.5.1 查看哪些用户正在上机	16
2.5.2 了解进程的运行状态	17

2.5.3 关于后台进程	17
2.6 如何使用 UNIX 的联机手册	18
第三章 文件系统使用	21
3.1 重要的标准目录和文件	21
3.2 主目录、工作目录与 pwd 命令	22
3.3 目录的组织与管理	22
3.3.1 创建目录:mkdir 命令	23
3.3.2 改变工作目录	23
3.3.3 列出目录内容:ls 命令	24
3.3.4 删除目录:rmdir 命令	26
3.5 基本的文件处理命令	26
3.5.1 显示、打印文件内容:cat, more, pr 和 lp 命令	26
3.5.2 复制文件或文件树:cp, rcp 和 ftp 命令	28
3.5.3 符号链接的建立与拆除:ln 和 unlink 命令	29
3.5.4 更改文件名:mv 命令	30
3.5.5 删除文件或文件树:rm 命令	30
3.5.6 查找所需的文件:find 命令	31
3.5.7 文件的压缩与恢复:compress、uncompress、gzip 和 gunzip	33
3.6 简单的信息处理	34
3.6.1 统计文件的行数、词数和字符数:命令 wc	35
3.6.2 辨识文件间的差异:diff 命令	35
3.6.3 查找简单的信息:grep 命令	37
3.6.4 检查文件中的拼写错误:spell 命令	37
第四章 全屏幕编辑程序 vi	39
4.1 启动全屏幕编辑软件 vi	39
4.2 掌握 vi 的诀窍	40
4.3 基本命令	40
4.3.1 在屏幕上定位光标的方法	40
4.3.2 插入或附加正文	41
4.3.3 删除正文	41
4.3.4 查找与修改正文	41
4.3.5 剪贴正文	42
4.3.6 行编辑命令	43
4.3.7 恢复命令 Undo	45
4.4 特殊命令	45
4.4.1 合并两行	45
4.4.2 重复前一条命令	46
4.4.3 改变英文字母的大小写	46
4.4.4 清除和重新显示	46

4.5 保存文本与退出 vi	47
4.6 命令 vi 的特殊语法	47
4.6.1 编辑多个文件	47
4.6.2 浏览文件	48
4.6.3 恢复因中断而丢失的文件内容	48
第五章 标准 SHELL 和 C SHELL	49
5.1 sh 和 csh 功能简介	49
5.2 Shell 命令语言	50
5.2.1 元字符: ? * [] --	50
5.2.2 命令的分隔与组合	51
5.2.3 输入输出重定向	51
5.2.4 进程的启动、执行、睡眠、唤醒和终止	53
5.3 Shell 程序设计	54
5.3.1 Shell 程序的基本结构及其运行	54
5.3.2 变量	55
5.3.3 流程控制语句	57
5.3.4 Here 文件	60
5.3.5 函数	60
5.3.6 中断信号	61
5.4 Shell 程序的调试	61
5.5 C Shell 的特点	61
5.5.1 用户注册初始化文件	62
5.5.2 命令历史	62
5.5.3 可设置命令输入提示符	62
5.5.4 命令的替换与引用	62
5.5.5 别名 alias	63
5.5.6 能理解标准 Shell 的用法	64
第六章 网络应用程序	65
6.1 计算机地址、网络域与电子邮件地址	65
6.2 电子邮件: mail 命令	66
6.2.1 发送电子邮件	67
6.2.2 mail 的交互环境及阅读、保存和删除邮件	68
6.3 了解计算机网的状态: ruptime、rwho 和 finger 命令	70
6.4 获取网络中计算机的绝对地址: nslookup 命令	71
6.5 在别的计算机上运行一个程序: rsh 命令	71
6.6 远程注册: rlogin 和 telnet 命令	72
6.7 网络文件传输: rcp 和 ftp 命令	73
6.7.1 网络文件名	73
6.7.2 直接拷贝命令 rcp	74

6.7.3 能力非凡的 ftp 命令	74
6.8 网络浏览器:gopher、mosaic 与 netscape	76

第七章 X 窗口系统..... 77

7.1 X 窗口系统的基本特征	77
7.2 X 的窗口管理器	78
7.3 X 窗口系统的启动及相应的环境配置	78
7.4 X 窗口的基本使用技巧	81
7.4.1 鼠标的的基本使用方法	81
7.4.2 创建新窗口	82
7.4.3 选择窗口菜单项的方法	82
7.4.4 改变窗口的层次	83
7.4.5 隐藏闲置的窗口	83
7.4.6 移动窗口	83
7.4.7 改变窗口的大小	83
7.4.8 刷新窗口	83
7.4.9 退出窗口	84
7.4.10 退出 X 窗口系统	84
7.5 X 窗口系统在网络方面的应用	84

第八章 系统维护..... 87

8.1 系统管理员与超级用户	87
8.2 系统管理员的日常职能	88
8.2.1 启动和关闭系统	88
8.2.2 增添和消除用户	88
8.2.3 新增或拆除终端	89
8.2.4 检查邮件	89
8.2.5 与用户通信	90
8.2.6 打印机管理:命令 lpstat、lpq 和 lprm	90
8.3 文件系统备份	91
8.3.1 档案文件的建立与抽取:命令 tar	91
8.3.2 功能强大的档案文件处理工具:命令 bar	92
8.3.3 文件系统后援:dump 命令	92
8.4 文件系统的监控与维护	93
8.4.1 了解可使用的硬盘空间:df 命令	93
8.4.2 清理文件系统:fsck 命令	93
8.4.3 文件系统的加载与拆卸:mount 与 umount 命令	94
8.4.4 网络文件系统与共享:exportfs 命令	96
8.5 对换区(swap)	97
8.6 配置网络环境	98
8.7 重新生成系统软件	100

8.8 微机 UNIX 系统的安装	100
第九章 模式扫描与处理——awk 语言	103
9.1 awk 语言的基础知识	103
9.1.1 记录与域	103
9.1.2 awk 程序的基本结构与运行过程	104
9.1.3 运行 awk 程序的简单方法	104
9.2 模式	105
9.2.1 BEGIN 和 END	105
9.2.2 关系表达式	106
9.2.3 正则表达式	107
9.2.4 逻辑运算与模式组合	108
9.2.5 逗号与模式范围	109
9.3 动作	109
9.3.1 变量与数组	109
9.3.2 算术运算	111
9.3.3 串操作与串函数	111
9.3.4 串与数	112
9.3.5 流程控制语句	112
9.3.6 自定义函数	114
9.4 输入、输出	115
9.5 应用实例	115
第十章 软件设计开发的支撑工具	119
10.1 支撑工具简介	119
10.2 C 语言编译器:cc	120
10.3 C 程序的检查工具:lint	122
10.3.1 lint 的使用方法	122
10.3.2 lint 应用实例	123
10.3.3 在 C 程序中控制 lint 的输出	127
10.4 档案库的生成与维护工具:ar 与 ranlib	129
10.4.1 命令 ar	130
10.4.2 命令 ranlib	131
10.5 中、大型程序的自动维护工具:make	131
10.5.1 make 的基本工作原理	131
10.5.2 依赖关系的描述	131
10.5.3 make 的内部转换规则	132
10.5.4 建立简单的 make 文件	132
10.5.5 注释与宏	133
10.5.6 make 命令的使用方法	134
10.5.7 关于 make 的一些高级技巧	135

10.6 源代码调试程序:sdb 或 dbx	137
10.6.1 源代码调试工具的基本功能	137
10.6.2 使用源代码调试工具 sdb 或 dbx 的场合	137
10.6.3 源代码调试工具对编译器的要求	138
10.6.4 源代码调试工具的使用方法	138
10.7 词法分析器:lex	140
10.7.1 lex 源程序的结构	141
10.7.2 lex 中的定义	141
10.7.3 lex 的正则表达式	142
10.7.4 lex 的动作	142
10.7.5 lex 处理二义性的准则	144
10.7.6 lex 的使用方法	144
10.8 编译器的自动生成工具:yacc	145
10.8.1 yacc 源程序的基本结构	146
10.8.2 说明段	146
10.8.3 文法描述的基本规范	149
10.8.4 关于文法描述的二义性	150
10.8.5 消除文法二义性的方法	151
10.8.6 动作	152
10.8.7 命令 yacc 的使用方法	155
10.9 源码控制系统:SCCS	156
10.9.1 SCCS 的版本结构	156
10.9.2 SCCS 的使用方法	157
第十一章 UNIX 的系统调用	163
11.1 显示系统错误信息:perror()	164
11.2 文件系统调用	164
11.2.1 基本函数:open()、lseek()、read()、write()和 close()	164
11.2.2 创建特殊文件:mknod()	166
11.2.3 改变目录:chdir()与 chroot()	166
11.2.4 改变文件的所有者和存取权限:chown()与 chmod()	167
11.2.5 查询文件状态:stat()与 fstat()	168
11.2.6 创建管道:pipe()	168
11.2.7 复制文件描述符:dup()	169
11.2.8 文件的链接与删除:link()与 unlink()	169
11.2.9 文件系统的加载与拆卸:mount()与 umount()	169
11.3 进程控制的系统调用	170
11.3.1 创建子进程	170
11.3.2 设置进程的有效 ID 号:setuid()、setgid()和 setpgpr()	171
11.3.3 软中断信号的发送与捕获:kill()与 signal()	171
11.3.4 等待进程的退出:wait()	172
11.3.5 终止一个进程	172

11.3.6 执行 Shell 程序:execve()	173
11.3.7 进程调度:nice()、pause() 和 sleep()	173
11.4 有关时间的系统调用	174
11.4.1 设置系统的时间和日期:stime()	174
11.4.2 查询当前的系统时间:time()	174
11.4.3 获得进程所消耗的时间:times()	174
11.4.4 置闹钟信号:alarm()	175
第十二章 进程间通信	177
12.1 管道与简单的同步通信	178
12.2 有名管道	179
12.3 信号与全局跳转	182
12.4 消息、信号灯和共享内存	185
12.4.1 消息	185
12.4.2 信号灯	189
12.4.3 共享内存	191
12.5 套接字与网络通信	196
12.6 小结	202
参考文献	204

UNIX 操作系统简介

1.1 UNIX 操作系统的发展历程

UNIX 操作系统是由美国贝尔实验室的 Ken L. Thompson 和 Dennis M. Ritchie 在 1969~1971 年设计和实现的,其最初的目的是为创建一个较好的程序开发环境。UNIX 直接吸取了 Multics 和 CTSS 的特征,UNIX 一词就是针对 Multics 的双关语,是由 Brian W. Kernighan 想出来的。UNIX 操作系统成功的关键在于它对少数关键思想作了恰如其分的选择并加以精美的实现。

UNIX 最早是在 PDP7 小型机上实现的,尽管早期的版本大有前途,但在未被用于实际项目之前 UNIX 没能发挥其潜力。UNIX 系统于 1971 年被成功地移植到了 PDP11 上。在研制成功 C 语言之后,Ritchie 于 1973 年又用 C 语言重写了 UNIX 操作系统,使它对其外部用户具有巨大的影响,这也是 UNIX 最早的正式版本,称为 UNIX 第五版。

UNIX 以其简洁、高效和可移植性好等特性在美国的大学和科研机构中赢得了一片赞誉,Thompson 和 Ritchie 在 1974 年第 7 期《ACM 通讯》上发表的“UNIX 分时系统”一文为 UNIX 的广泛流行起了推波助澜的作用,而 Thompson 和 Ritchie 于 1983 年获得的图灵(Turing)奖则进一步承认了 UNIX 对计算机工业的卓越贡献。

1977 年是 UNIX 发展的关键一年,首先是交互系统公司(Interactive System Corp.)成为 UNIX 的第一个增值转卖商,增强了 UNIX

的功能,使之在办公室自动化中得到应用;这一 UNIX 取得的另一个成就是被成功地移植到了非 PDP 机——Interdata 8/32 计算机,这表明 UNIX 稍加改变或完全不变就可运行于别的机种。

1978 年公布的 UNIX 第七版对后来 UNIX 的进一步发展有深远的影响,它主要运行于 PDP11/70 和 Interdata 8/32 计算机上。到 1979 年,UNIX 又被移植到了 DEC 公司的 VAX 系列机上,称为 V32。随着微处理器的日益普及和迅速发展,其它公司也把 UNIX 系统移植到新的机器上,而 UNIX 简单清晰的特点却吸引着很多开发者以它们自己的方式增强 UNIX 系统,因此出现了许多 UNIX 的变体,其中,最具代表性的变体包括 BSD (Berkeley Software Distribution)、Ultrix、XENIX 和 LINUX 等。贝尔实验室则在 1977 年到 1982 年这段时间内将若干 AT&T 变体综合起来,形成了商用 UNIX 系统 III;后来,贝尔实验室又将若干特征融入系统 III 上,并于 1983 年形成新产品 UNIX 系统 V,这也是当今最流行的 UNIX 版本。UNIX 从此被工业界所承认,并成为许多厂商和行业共同采用和推广应用的对象,这使得到 1986 年为止就有约 2 000 000 计算机安装了 UNIX 操作系统。

特别值得一提的是 U. C. Berkeley 的 Bill Joy 和 Ozalp Babaoglu 于 1979 年所开发的 3BSD,于 1983 年发展成 UNIX 4.2BSD,并于 1987 年形成 UNIX 4.3BSD 版本。贝尔实验室的 UNIX 虽然提供了共享文件指针、信号(signal)、进程跟踪(trace)、文件、无名管道(pipe)、有名管道(named FIFO)、信号灯装置(semaphore)、消息(message)和共享内存(shared memory)等 9 种进程间通信(inter-process communication)机制,但这些方式均不能令人完全满意;而 BSD 提供的套接字(socket)则为进程间通信和分布式处理提供了一种比较有效的方法,为计算机网络的发展作出了重要的贡献,也使 UNIX 在计算机网络领域独领风骚。

为了让读者更清晰地了解 UNIX 阵营中系统 V 和 BSD 这两大流派的各种版本间的关系,图 1.1 给出了 UNIX 操作系统的发展史。

1.2 UNIX 的系统结构及主要特点

图 1.2 是 UNIX 系统的体系结构示意图。图的中心是计算机硬件,它为 UNIX 软件提供诸如中断处理、存储管理等支持。靠近硬件的内层称为 UNIX 的**内核(kernel)**,也就是我们所说的操作系统,它直接同计算机硬件打交道,并为外层应用程序提供公共服务。内核的主要作用是将应用程序和计算机硬件隔离起来,这使得应用程序不依赖具体的计算机硬件,因而为应用程序提供了很好的可移植性。内核程序可分为文件子系统和进程控制子系统两大部分,而进程控制子系统又可分为内存管理、进程调度和进程间通信等三个模块。内核是 UNIX 系统中唯一不能由用户任意改变的部分。内核的外层是实用程序,内核提供对实用程序的支持,而系统调用则是沟通 UNIX 内核和实用程序的桥梁。

外层的程序包括命令解释器 shell、正文编辑程序(vi, ed, emacs 等)、C 编译程序、源码控制系统 SCCS 和文档准备包等,应用程序通过系统调用与内核进行交往。这些系统调用通知内核为应用程序作哪些工作,并在核心与应用程序之间交换数据。图 1.2 中的许多程序属于 UNIX 的标准配置,也就是我们平常所说的命令。名为 a.out 的程序是用户的私有程序,

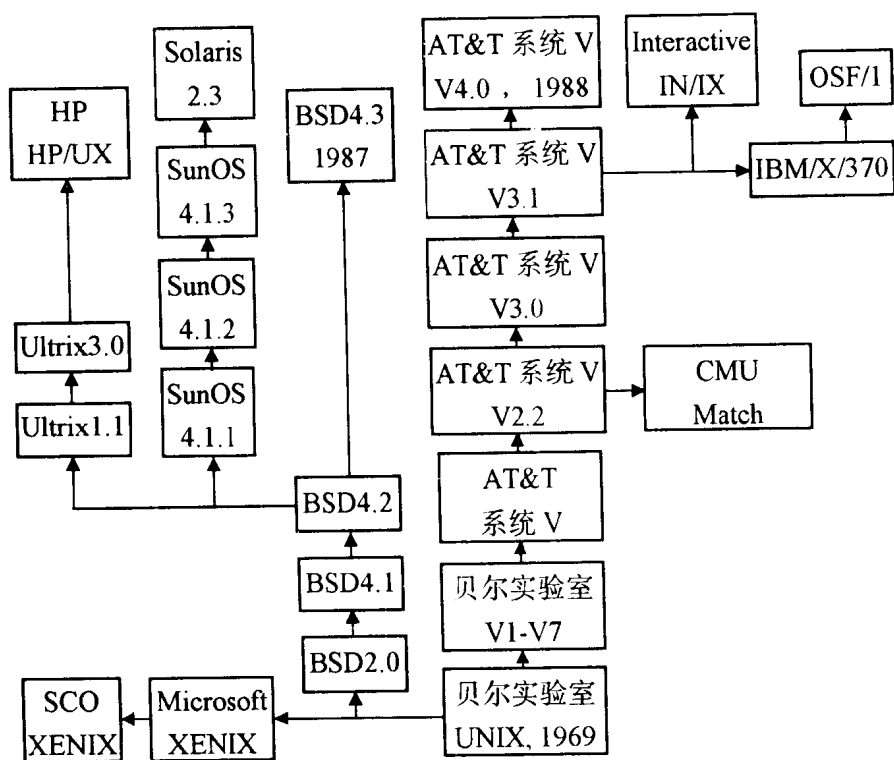


图 1.1 UNIX 操作系统发展史

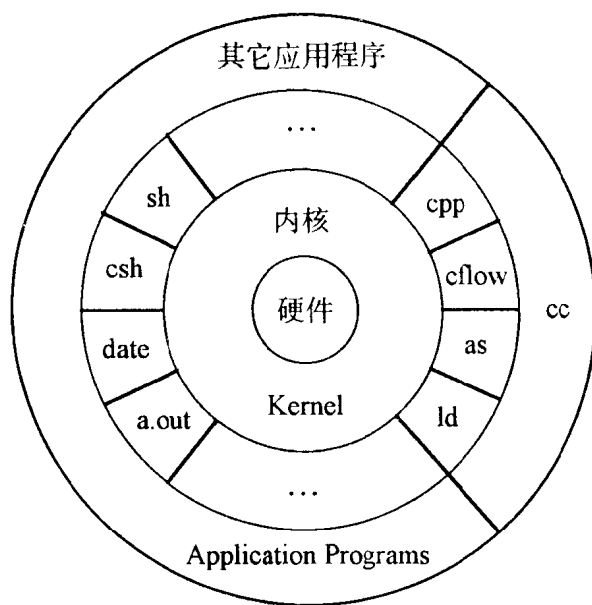


图 1.2 UNIX 系统的体系结构