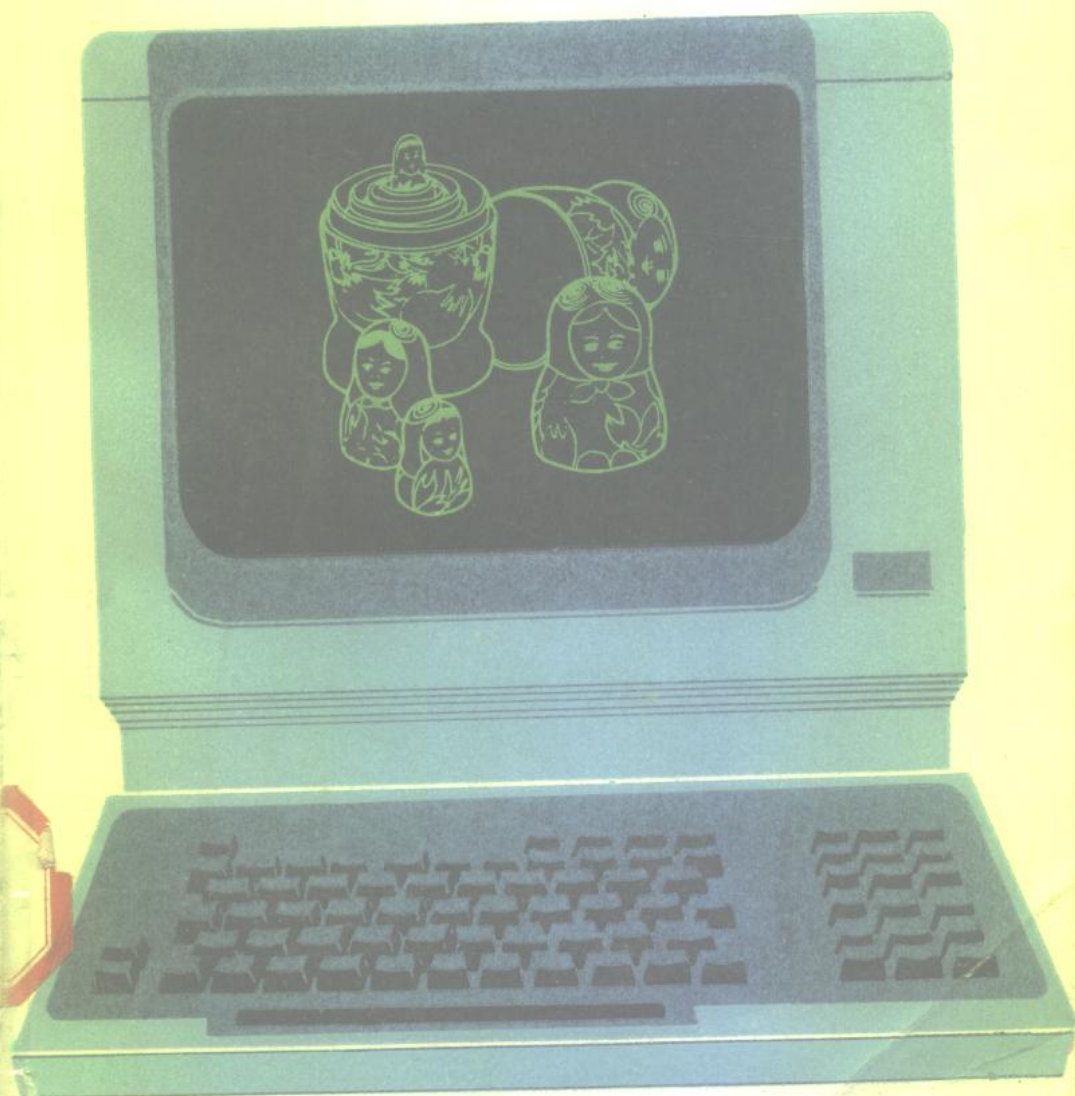


微型计算机应用丛书

UNIX 系统

〔美〕S.R.波恩 著



机 械 工 业 出 版 社

微型计算机应用丛书

UNIX 系统

〔美〕 S.R.波恩 著

朱亦梅 高传善 译

朱逸芬 校



机械工业出版社

UNIX 系统是当前最为流行的操作系统之一, 从大型计算机到高档微型机无不广泛使用。本书共 8 章, 内容适合各系统中大多数公共功能, 具有简洁通用的特点。书中还收集了大量实例, 起实用指南的作用, 并为进一步深入使用和开发系统提供方便。

本书可作为高等院校计算机专业师生或各种计算中心工作人员的参考教材。

The UNIX System

S. R. Bourne

Addison- Wesley Publishing Co.

1983

* * *

微型计算机应用丛书

UNIX 系统

〔美〕S. R. 波恩 著

朱亦梅 高传善 译

逸芬 校

机械工业出版社出版 (北京皇城根外大街万庄南里一号)

(北京市书刊出版业营业许可证出字第 117 号)

机械工业出版社印刷厂印刷

新华书店北京发行所发行 新华书店经售

*

开本 $850 \times 1168 \frac{1}{32}$ · 印张 $13 \frac{3}{4}$ · 字数 358 千字

1989 年 5 月北京第一版 · 1989 年 5 月北京第一次印刷

印数 0,001— 2930 · 定价: 12.00 元

*

ISBN 7- 111- 00668- 2 / TP · 46

出版者的话

计算机是现代化建设中不可缺少的先进工具,其应用正在向各个领域渗透,并以日新月异的面貌迅猛发展。

为了迅速普及计算机先进科学知识,大力推广计算机应用技术,积极提高技术管理干部的现代化管理水平和工程技术人员应用计算机的水平,中国机械工程学会自动化分会和机械工业出版社根据目前急需情况,先组织出版一套《微型计算机应用丛书》,分编著和翻译两个独立部分,均以应用为重点,内容反映微型计算机在机械、电气自动化、仪器仪表、办公自动化等方面的应用,供使用计算机的工程技术人员参考。

本丛书的出版得到原机械工业部计算机领导小组的大力支持,原机械工业部计算机与集成电路办公室,原北京机械工业自动化研究所等单位的有关同志给予了具体指导和帮助,其他兄弟单位提供了方便,对此一并表示感谢。

编委会名单

主任:沈烈初

副主任:蔡福元 罗命钧 郑仁贵

委员(按笔划排):

朱逸芬 李襄筠 严蕊琪

张长生 周 斌 李瑞芝

郑学坚 龚为廷 谢志良

葛林根

译者序

UNIX 系统是当前著名的交互式分时操作系统。由于它丰富、简便又易于移植,故已成为目前最为流行的操作系统之一。它在大型计算机到高档微型计算机中无不广泛使用。由于历史的原因,流行的 UNIX 系统存在许多不同的版本,相互间存在一些差异,给用户学习和使用带来某些不便。本书是根据贝尔实验室 S.R. 伯恩所著 1983 年版本译出,作者排除了个别版本之间的特异之处,力求内容适合各系统中大多数公共功能,具有简洁通用的特点。此外,本书是为实用指南目的而编写,故穿插大量实例,即方便初学者掌握,又为已经熟悉系统的科技人员提供一本完整、全面的参考手册,为进一步深入使用和开发系统提供方便。本书也可作为高等院校计算机专业师生或各种计算中心工作人员学习进修操作系统课程的参考教材。

本书的前言、第一、二、三、四、五章及附录 1、3、4、5、6 由上海大学朱亦梅副教授译,第六、七、八章及附录 2、7、8、9、10、11 由复旦大学高传善副教授译。全书由合肥工业大学朱逸芬教授校订。

由于译文内容广泛、名词繁多,以及译者水平有限,不当或欠妥之处敬请读者批评指正。

序

本书是 UNIX 系统的实用指南,不论是对初学者还是专家,本书都能提供帮助。各种实例,贯穿全书,用以说明使系统的使用具有吸引力的各种技术。用户可从书中给出的各种命令之间互相联系的实例中取得 UNIX 系统所提供的全部优点。

概论回顾了早期 UNIX 系统的历史背景,并介绍了包括文件和进程在内的 UNIX 系统的各组成部分。书的内容涉及 UNIX 操作系统的主要部分,包括文件管理、进程管理、系统调用、编辑程序、shell、C 语言、troff 和 nroff 以及数据管理工具。

第二章除了向用户介绍系统外,还同时介绍常用的命令。说明了请求注册的过程,并描述了可用的文本。在文本中介绍的二个主要题目是 shell 和文件系统,后面的章节则以已经具备第二章的基础进行叙述的。

在第三章中,我们介绍二个编辑程序 ed 和 vi。这些程序都是用于在终端上建立和修改文件的。在学习第二、第三章的时候,手边能有一台终端,并随时试验某些命令,则是颇为理想的。假如你不能取得进展,则应请求同事们的帮助,这往往是学习操作系统的捷径。

Shell 既为交互式作用的用户,又为个人或课题设配环境用的稿本提供 UNIX 操作系统的接口。Shell 的交互使用首先是在第二章中介绍。写 shell 稿本或程序,则在第四章中再深入论及。

在第五章中对 C 语言进行了相当细致的介绍,以致能编写适当大小的程序。读这一章时,若能熟悉另一种程序设计语言将是很有益的。用 make 为程序管理资源和用 adb 调试也安排在这章里介绍。

下一章描述 C 语言程序设计员所看到 UNIX 操作系统接口。着重在用由操作系统直接提供的工具来写程序。这一章是为用 C 语言写命令的用户而写的。第二章中尚未谈到的文件系统中更

先进的方面也将在这里介绍。

UNIX 操作系统的主要用途之一是正文处理和文本准备。程序集 `nroff`, `troff`, `egh`, 和 `tbl` 和正文编辑在一起将在第七章中介绍, 它们使人们能很容易地起草和修改文本。这一章中还有一个本书中所用的格式化程序包的工作举例。

最后一章涉及诸如 `awk`, `grep`, `sort` 和 `join` 这样一些程序所组成的数据处理“成套工具”, 这就为管理小型数据库和操作少量数据提供了一个灵活的方法。本章将例举用这些部件构成的完整系统。每个例子都将详细说明, 而标准 UNIX 操作系统所不具备的新型工具的结构也将在此出现。阅读第八章时, 将用到以前章节中所提供的全部材料。

附录总结了本书中所用到的各种命令。

本书涉及从用户观点所看到的 UNIX 操作系统的许多方面。我们假定本书的读者已在一定程度上熟悉现代计算技术的术语。

目 录

序

第一章 概论	1
1.1 历史回顾	1
1.2 程序设计环境	5
1.3 UNIX系统概念	6
1.3.1 文件系统	6
1.3.2 进 程	6
1.3.3 shell	7
第二章 启动	8
2.1 注 册	8
2.2 命 令	9
2.3 终端特性	10
2.4 文 本	12
2.5 文件系统	14
2.5.1 简单文件操作	15
2.5.2 对目录进行操作	17
2.6 shell	18
2.6.1 管道和筛选程序	19
2.6.2 文件名生成	20
2.6.3 引 述	21
2.6.4 提 示	22
2.6.5 shell 和 login	22
2.6.6 复 习	23
2.7 有用的命令	23
2.7.1 通 信	23
2.7.2 系统查询	26
2.7.3 进程管理	27
2.7.4 其他命令	29
2.7.5 复 习	33

第三章 编辑文件	34
3.1 编辑程序 ed	34
3.1.1 建立文件	34
3.1.2 行编辑	36
3.1.3 上下文搜索	39
3.1.4 上下文编辑	40
3.1.5 模式匹配	43
3.1.6 全局的编辑	45
3.1.7 其他	46
3.2 编辑程序 vi	49
3.2.1 窗口控制	50
3.2.2 光标控制	50
3.2.3 添加和删除	51
3.2.4 行编辑	52
3.2.5 移动内容	53
3.2.6 复习	54
3.2.7 更高级的性能	54
第四章 shell	60
4.1 shell过程	60
4.1.1 控制流 - for	62
4.1.2 控制流 - case	63
4.1.3 Here文本	64
4.1.4 shell变量	66
4.1.5 test命令	69
4.1.6 控制流 - while和 until	70
4.1.7 控制流 - if	71
4.1.8 命令组合	73
4.1.9 调试 shell过程	73
4.1.10 man命令	75
4.2 高级使用	75
4.2.1 参数发送	75
4.2.2 参量替代	76
4.2.3 命令替代	77

4.2.4 求值与引述	78
4.2.5 出错处理	81
4.2.6 故障处理	82
4.2.7 命令执行	84
4.2.8 输入-输出重定向	85
4.2.9 引用 shell	87
4.3 内部命令	87
第五章 C程序设计语言	91
5.1 C程序实例	92
5.1.1 一个简单程序	92
5.1.2 一个八进制转贮	93
5.1.3 平均距离	95
5.2 语言	98
5.2.1 词法考虑	98
5.2.2 表达式和运算符	101
5.2.3 控制流	106
5.2.4 函数	109
5.2.5 数组和指针	110
5.2.6 结构和联合	114
5.2.7 C预处理程序	120
5.2.8 语言结构	122
5.2.9 标准C程序库	124
5.2.10 结束语	127
5.3 程序组织和管理	128
5.3.1 编译程序	128
5.3.2 make命令	129
5.3.3 lint命令	131
5.3.4 程序库	132
5.3.5 性能测定	133
5.3.6 其他工具	134
5.4 调试C程序	134
5.4.1 调试内存映象	135
5.4.2 adb请求	136

5.4.3 adb 格式	137
5.4.4 在 adb 中设置断点	138
5.4.5 地址映射	139
第六章 UNIX 系统程序设计	141
6.1 参数约定	141
6.2 基本输入 - 输出	144
6.2.1 open 系统调用	145
6.2.2 读和写文件	146
6.3 再论文件系统	148
6.3.1 文件许可权	148
6.3.2 改变文件方式	149
6.3.3 目录访问	150
6.3.4 组	151
6.4 高级输入 - 输出	152
6.4.1 建立和撤消文件	152
6.4.2 随机访问输入 - 输出	156
6.4.3 文件的状态	159
6.4.4 终端输入 - 输出	161
6.4.5 管道	164
6.5 进程	165
6.5.1 进程执行	165
6.5.2 fork 系统调用	166
6.5.3 wait 系统调用	166
6.5.4 exec 系统调用	167
6.5.5 exit 系统调用	168
6.5.6 进程的环境	169
6.6 信号与中断	170
6.6.1 信号	170
6.6.2 发送信号	171
6.6.3 拦截信号	172
6.6.4 后台进程	173
第七章 文本准备	175
7.1 nroff 和 troff	175

7.1.1 准备文本的指南	176
7.1.2 简单的请求	178
7.1.3 高级请求	184
7.1.4 宏指令库	200
7.2 制作工具	210
7.3 文本处理工具	214
7.3.1 col 命令	215
7.3.2 diction 命令	215
7.3.3 eqn 命令	216
7.3.4 ptx- 排列索引的生成	218
7.3.5 refer 命令	219
7.3.6 spell 命令	221
7.3.7 style 命令	221
7.3.8 tbl 命令	221
第八章 数据处理工具	225
8.1 工具的简短描述	226
8.1.1 awk- 报告生成程序	227
8.1.2 cmp- 比较两个文件	232
8.1.3 comm- 选择公共行	233
8.1.4 diff- 文件区分	233
8.1.5 grep- 模式选择	234
8.1.6 join- 组合文件	235
8.1.7 sed- 流编辑程序	237
8.1.8 sort- 排序或合并文件	238
8.1.9 tail- 一个文件的最后几行	239
8.1.10 tr- 翻译字符	239
8.1.11 uniq- 移去重复的行	240
8.1.12 field- 选择栏	241
8.1.13 lex 和 yacc	241
8.2 简单的例子	241
8.2.1 维护一个简单的数据库	241
8.2.2 C 语言交叉对照程序	245
8.3 网球进级阶梯系统	254

8.3.1 组 织	255
8.3.2 输 入 结 果	256
8.3.3 每 周 的 活 动	261
8.3.4 进级阶梯的更新	263
8.3.5 按周打印进级阶梯	270
8.3.6 季度的开始和结束	273
8.4 实现 field 命令	277
附录1 命令集	281
附录2 系统调用	346
附录3 C 子程序	369
附录4 adb 请求	389
附录5 ed 请求	392
附录6 sh 请求	394
附录7 troff 请求	398
附录8 vi 请求	406
附录9 宏指令库	410
附录10 ms 宏定义库	417
附录11 ASCII 字符集	423
参考文献	424

第一章 概 论

UNIX 描述由贝尔实验室所开发的一系列计算机操作系统。UNIX 系统包括了操作系统和与其相关的命令。操作系统提供了一个层次结构的文件系统、进程管理和其他例行功能以管理计算环境的资源。所提供的命令包括基本的文件和数据管理、编辑、汇编、编译和文本格式化。功能很强的命令解释程序允许各个用户或课题用定义他们自己的命令来设置适合他们自己风格的环境。

导致首版 UNIX 系统的背景值得考察。60 年代, 计算科学界提出的主要问题包括程序设计语言和操作系统设计。在语言方面, 设计了诸如 PL/1、APL、SIMULA67、ALGOL68 和 COBOL, 对这些语言的相对优劣功过, 常有激烈的争论。在英国, 伦敦大学和剑桥大学曾联合承担研制组合程序设计语言计划(CPL), 但未得出成功的直接结果。然而, 它的确构成了 BCPL(基本 CPL)的基础, 成为它的一个组成部分。

这一时期的操作系统是为中、大型计算机设计的, 目的在于能使许多用户经济地共享计算机资源。引入了分时和交互使用(与批处理相对立)。探索了有关页面策略、保护设施、激活调度和文件系统设计这样一些问题。诸如 CTSS (Crisman, 1965) Multics (Feiertag, 1969) 和欧州的剑桥多路访问 (Cambridge Multiple Access System) (Hartley, 1968) 等系统被设计出来, 从而为 UNIX 系统提供了许多关键的思想。例如, 文件系统和与设备无关的输入 - 输出、进程和命令语言等均以这样或那样的形式出现在这些系统中。

1.1 历史回顾

事情要从 1968 年肯·汤普森(Ken Thompson)说起。当时汤普森刚从伯克利(Berkeley)回来, 在那里巴特里·兰普森(Butler Lampson)正在进行 SDS 930 操作系统的研究(多伊奇(Deutsch)

和兰普森(Lampson), 1965)。1967年在哈佛致力于应用数学研究的丹尼斯·理奇(Dennis Ritchie)也从哈佛来到了贝尔实验室。

汤普森参与了一个智能集团, 集团中许多人刚放弃了一项由贝尔实验室、通用电气公司、和麻萨诸塞理工学院联合研究的Multics课题。随着贝尔实验室退出Multics以及GE-645系统于1969年3月的搬迁, 该计算机科学研究组开始寻找新的替代计算环境。有关新设备购置的建议由于耗资太大而被拒绝。此外, 在Multics跨台以后, 操作系统的开发当时也不是一个很流行的研究方向。

汤普森本人的兴趣也倾向于建立一个文件系统而不想搞一个操作系统。经过拉得·卡纳德(Rudd Canaday)、汤普森和理奇之间的讨论, 草拟了设计方案, 并由汤普森写出了GECOS系统上该文件系统的早期模拟版本。

事情的另一方面是由汤普森和理奇在GECOS机上写了“空间旅行(Space travel)”程序。这个程序对分时计算机表现得功能贫乏, 并需要较好的响应。有一台抛弃的带有一台340显示器的PDP 7机可供使用, 但PDP 7仅提供了汇编程序和装入程序, 是一个单用户系统, 每个用户都需要独占一台机器。这种环境是原始的, 因而单用户UNIX系统的一部分不久就出现了。空间旅行程序为PDP 7进行了重写。还写了汇编程序和雏形的操作系统核心, 并在GECOS系统上对PDP 7进行了交叉汇编。这一早期系统并未提供分时功能。实际上, 很象现代的个人计算机, PDP 7的硬件很简单, 并不提供对这些活动的支持。不久, 出现了汇编程序和命令解释程序。这一文件系统提供了一个命名结构, 它是一有向图。所有的子目录使用一个单一的目录, 并由这个目录来链接。

交叉汇编意味着使用两个计算机系统, 并从一个机器到另一机器输送纸带程序, 而每送一次则完成一次转换。不久, 这一系统被装入PDP 7机上, 在这一重写的过程中, 把进程建立原语(fork)和进程映象(process images)都加进了这个系统。基本实用程序, 例如文件拷贝、编辑、迁移和打印等不久也都能提供了。这一系统提

供了两人同时工作的可能性,因而在 1970 年由布赖恩·勘涅根(Brian Kernighan)提出了 UNIX 这一名词。

计算研究小组仍没有自己的计算机。在一系列失败的尝试之后,乔·奥赛纳(Joe Ossanna)提议为文本预处理课题购置一台 PDP 11/20 机。在 1970 年的后期 PDP11 机到达,并开始将 UNIX 系统移植到这一功能更强的机器上去。

文本处理课题是成功的,而专利局则成为和研究小组共享设备的第一个 UNIX 用户。第一版系统由汤普森和理奇编写成一本手册,标记的日期是 1971 年 11 月。除了管道技术外这一版本中还包含了现代 UNIX 系统中所有重要的思想,包括文件系统、进程管理、系统接口和大多数命令实用程序。

1972 年 6 月在唐·麦克衣劳(Doug McIlroy)的推动下,结合管道技术的第二版系统出现了。这一系统及其实用程序仍是用汇编语言写的。汤普森也曾从事过 B 语言的研究,而这一系统的汇编程序是用 B 语言写成的。B 语言是直接由 BCPL 衍生而来的。但程序编译时是一次通过立即产生翻译码的。

B 和 BCPL 都是无数据类型的语言,只提供单一的数据对象,称为机器字。这使访问 PDP11 字节处理指令发生困难。于是在 B 语言中加入了由数据类型而产生 NB 语言,然而企图用 NB 语言来改写系统的尝试没有获得成功。理奇开始从事用于 NB 语言的编码来生成程序的研究,以便加快程序的执行速度。这种语言称为 C 语言。虽然它仍是没有结构和全局变量,然而这种语言还是很有吸引力的,许多新的实用程序都是直接用 C 语言编写的。

1973 年情况有了很大的进展。虽然系统仍是采用汇编语言编写,但在 C 语言中补充添加了结构后,UNIX 系统也随之用 C 语言写成。汤普森写了进程管理,而理奇写了输入-输出系统。

第六版的 UNIX 系统是第一个广泛应用的版本,在 1975 年 5 月问世并以极低的费用而流传。

以后的工作便是进一步改进系统。新的文件系统使能写出较大的文件,而 shell 的现代化为许多用这种语言编写的程序提供了