




L i n u x / U N I X 开 发 与 应 用 系 列



UNIX

程  
序  
设  
计  
教  
程

赵克佳 沈志宇 赵慧 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



# UNIX 程序设计教程

赵克佳 沈志宇 赵慧 编著

清华大学出版社

## (京)新登字 158 号

### 内 容 简 介

本书是 UNIX 系统程序设计方面较全面的著作,以最新 UNIX 98 程序设计界面为主导,兼顾 V UNIX 和 BSD UNIX 的特点,系统地讲述了 UNIX 程序设计 API 的各种函数及其编程方法。内容包括:UNIX 导论,标准输入输出,低级输入输出,文件与目录,进程环境与控制,日期和时间,信号处理,终端 I/O,高级 I/O,进程之间的通信,套接字与网络通信。

本书在介绍 UNIX API 各种函数的同时,深入浅出地讲述了它们所隐含的操作系统基本原理,书中给出的大量程序实例有助于读者深入、透彻地掌握这些函数的用途、使用方法以及程序设计技巧。

本书内容丰富,叙述系统、全面,适用于大专院校计算机和相关专业的教师、高年级学生、研究生以及计算机软件科研和工程技术人员,也适合作为自学人员的参考书,同时还可以作为一本 UNIX 实用编程手册供计算机程序员查阅。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

**书 名:** UNIX 程序设计教程

**作 者:** 赵克佳 沈志宇 赵慧

**出 版 者:** 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

**责任编辑:** 胡先福

**印 刷 者:** 世界知识印刷厂

**发 行 者:** 新华书店总店北京发行所

**开 本:** 787×1092 1/16 **印张:** 36.5 **字数:** 861 千字

**版 次:** 2001 年 4 月第 1 版 2001 年 4 月第 1 次印刷

**书 号:** ISBN 7-302-04341-8/TP·2544

**印 数:** 0001 ~ 5000

**定 价:** 50.00 元

# 前 言

本书介绍 UNIX 系统与程序设计的界面,这个界面即 UNIX 系统提供的各种系统调用和库函数。本书讲述它们的用途、使用方法以及所涉及的基本概念。

UNIX 是一个多用户多任务的操作系统,它是 20 世纪 60 年代末、70 年代初在 AT&T 作为计算机科学实验的结果而发明的。经过 30 年的历程,UNIX 已发展成为当代最富影响力且广为流传的计算机环境之一。如今,几乎在所有计算机商家、大专院校、科研机构以及国民经济的各行各业,如石油勘探、气象、银行、证券、保险、商务贸易等都可以看到使用 UNIX 系统的计算机,这些计算机类型各异,从资源有限的微机到高档工作站、大型机直至巨型机。特别是 90 年代以来,随着 Internet 的迅速发展,UNIX 由于其一直作为网络服务器的首选操作系统和它的开放环境以及其标准逐渐趋于统一等原因,使得它在计算机发展中的地位更加稳固,并且正在迅速地从小型机推广到初级用户中。

然而,UNIX 仍然是一种高性能且比较复杂的操作系统。使用 UNIX 开发应用程序不仅需要了解它的各种命令和它所提供的工具和实用程序,更重要的是还必须掌握 UNIX 编程方法,了解 UNIX 提供的编程界面。如果你是一个熟悉 C 语言、了解 UNIX 基本操作命令的程序员,并且希望精通 UNIX 提供给软件开发人员的各种编程接口,从而减少应用软件的开发时间,那么,本书正对你的胃口。本书由浅入深的介绍、清晰的叙述以及大量的例程序将帮你掌握 UNIX 系统编程的关键技术,学会正确使用 UNIX 的各种系统调用和库函数,理解隐藏在它们背后的 UNIX 基本概念和原理。

严格地说,UNIX 是由 X/Open 掌管的商标并且指的是与 X/Open 标准“单一 UNIX 标准版本 2(Single UNIX Specification Version 2)”一致的计算机操作系统。目前这个标准已归于 UNIX 98 标准之下,它定义了 UNIX 操作系统提供的所有函数的行为和接口。X/Open 标准在很大程度上是早期由 IEEE 定义的 POSIX 一系列标准的超集。本书的目标是介绍与 UNIX 98 标准一致的 UNIX 程序设计。UNIX 程序设计是一个很大的范畴,我们主要集中于该标准的应用程序编程界面(API)。

全书共分 12 章。第 1 章是 UNIX 导论。在这一章我们简单讲述 UNIX 的发展过程,介绍关于它的几种标准,讲解与系统有关的一些基础知识。同时,为了方便后续章节的叙述,我们将与 UNIX 全系统有关的一些内容也放置在本章,例如,系统信息、错误处理、系统配置参数等。

第 2、3 和 4 章讨论 UNIX 中文件的输入输出和管理。其中第 2 章介绍标准输入输出函数,对 C 语言非常熟悉的读者可以跳过这一章。第 3 章介绍低级输入输出函数,这些函数都是系统调用,它们是标准输入输出函数的低级原语,在一些特定的情况下会不可避免地需要使用这些函数。第 4 章讨论 UNIX 文件和目录的有关内容,介绍文件系统的基本概念以及管理、操纵文件和目录的函数。

第 5、6 章介绍与进程相关的内容。进程是 UNIX 系统的核心概念之一。第 5 章叙述进程的执行环境,如进程的启动点、环境变量、进程的地址空间、存储分配、进程的终止等。第 6 章讨论进程控制,介绍父进程、子进程、进程组、会晤期等概念,讲述创建进程、执行新程序、终止和控制子进程的各种函数,同时也较为详细地介绍了有关作业控制的内容。

第 7 章讲述与日期和时间有关的函数,讨论如何统计程序的执行时间、设置闹钟、获取日期和时钟以及怎样设置时区并在不同的时间表示之间进行转换。

第 8 章讲述 UNIX 的信号及其函数。信号是 UNIX 进程之间进行同步和协调的重要手段。在这一章我们不仅介绍了信号函数的用法,而且比较详细地介绍了 UNIX 的信号机制以及编程注意事项。

第 9 章的内容涉及低级终端 I/O。由于 X 窗口系统的发展,现在已很少有应用需要特别设置终端,但低级终端 I/O 涉及的是串行端口,我们这一章的目的是通过终端 I/O 来介绍串行通信的基本概念,描述通用终端界面函数以及如何利用这些函数进行串行端口程序设计。

第 10 章在高级 I/O 的标题下将一些比较深入然而离散的内容归集在一起,这些内容涉及文件锁、多路 I/O、存储映像 I/O 以及系统 V 的流等概念和函数。这一章有些内容是后面进程间通信和网络编程两章的基础。

第 11 章讨论进程间的通信,介绍管道、FIFOs 以及系统 V IPC 的消息队列、共享存储、信号灯等概念。在这一章我们主要讨论同一台计算机中不同进程之间实现数据交换和协同工作的方法。

第 12 章介绍套接字和与网络程序设计有关的函数。我们将讲述如何获取网络中的主机信息和地址,什么是客户/服务,如何获取通信端口号以及使用套接字函数进行网络通信的方法。

最后的附录列出了本书介绍的所有函数的清单和所在章号以便查阅。期望阅读完本书后记住每一个函数并完全了解它们的使用方法是现实的,重要的是从总体上熟悉这些函数的种类,在编写程序时能够知道在什么情况下需要使用哪一种函数以及知道在何处能够找到关于它们更专门、更详细的说明就可以了。

本书在介绍这些函数时,尽可能地给出了示例程序。所有示例程序都在计算机上实际运行过,随书附带的光盘中包含了这些程序。这些程序虽然不是实际的应用程序,但是它们小巧、简单,易于说明问题并且易于理解。

我们希望读者通过轻松地阅读本书而进入 UNIX 编程世界,熟悉和掌握 UNIX 编程的各种接口和方法。本书包括了 UNIX 程序设计与与操作系统关系密切的大部分函数并且尽可能地全面描述它们。因此,本书除了适合作为大专院校计算机专业学生的教材和自学人员的参考书之外,也可以作为一本实用编程手册供计算机程序员查阅。

本书的编写起因于在工作中经常需要用到这些函数,而 UNIX 系统中关于它们的联机手册却过于专业。这种手册对于专业人员而言起备忘的作用,对于初学者来说则用处不是很大,因为没有掌握隐藏在其背后的 UNIX 基本概念和术语,仅仅通过阅读这些函数的说明是无法透彻理解它们的。然而当时几乎找不到一本介于联机手册和 UNIX 操作系统内部实现技术之间的书,这种书能够从普通用户的角度介绍函数调用的方法,讲解在

UNIX 中关于它们的编程基本原理。由此,我们萌发了编写此书的念头。历经两年多的时间,本书终于成稿。由于 UNIX 本身非常庞大,UNIX 应用极其丰富以及我们水平所限,书中难免存在错误,诚恳欢迎读者指正。

在本书的编写中我们得到了亲人和同事的理解、关心和支持。首先要感谢我们的双亲,是他们的抚育和教养使我们在社会中能够尽自己的一份职责,直到今天他们仍在给予我们不断的支持。另外要感谢我们正在读中学的儿子,他十分关心本书的编写工作,他的懂事和逐渐成长的独立、自主能力使我们得以专心于本书的写作。最后,我们特别要感谢国防科大计算机学院计算机研究所系统软件研究室的同事们。这里有着浓厚的学术探讨和科研风气,有着真正的 UNIX 专家和高手,我们在工作上得到了他们的协作。该书的写作过程中,从机器设备到资料素材均得到了他们的帮助和指点,对此我们由衷地表示感谢。

作者

2000 年 12 月

### 作者简介

赵克佳(女) 国防科学技术大学计算机研究所研究员,政府特殊津贴享受者。从事计算机科研工作 20 多年,主要研究方向为程序设计语言及其编译、并行处理以及计算机系统软件。先后参与我国多项巨型机系统软件的研制工程,曾获部委级科技进步一等奖 2 项、二等奖 4 项。参与编写了以下教材和专著:

《超级计算中的依赖关系分析》	译著	湖南科技出版社	1991 年
《程序设计语言编译原理》	编著	国防工业出版社	2000 年
《并行编译方法》	著	国防工业出版社	2000 年

沈志宇 国防科学技术大学计算机学院教授。从事计算机教学与科研工作 20 多年,主要研究方向为并行处理以及计算机系统软件。先后参与我国多项巨型机系统软件的研制工程,曾获国家科技进步一等奖 1 项、部委级科技进步一等奖 1 项、部委级科技进步二等奖 4 项。主编了以下教材和专著:

《超级计算中的依赖关系分析》	译著	湖南科技出版社	1991 年
《并行程序设计》	编著	国防科大出版社	1997 年
《并行编译方法》	著	国防工业出版社	2000 年

联系地址:

湖南长沙国防科学技术大学计算机研究所,邮政编码: 410073

E-mail: kjzhao@nudt.edu.cn 或 zyshen@undt.edu.cn

电话: (0731)4573682, (0731)4573681

联系人: 赵克佳

# 目 录

<b>第 1 章 UNIX 导论</b> .....	1
1.1 UNIX 简史 .....	1
1.1.1 UNIX 的诞生 .....	1
1.1.2 UNIX 的早期发展 .....	2
1.1.3 BSD UNIX .....	3
1.1.4 系统 V UNIX .....	4
1.1.5 UNIX 的商业化 .....	4
1.2 标准 .....	5
1.2.1 SVID .....	5
1.2.2 POSIX .....	6
1.2.3 X/Open .....	6
1.2.4 C 标准 .....	8
1.3 UNIX 基本概念 .....	8
1.3.1 程序和进程 .....	8
1.3.2 内核 .....	10
1.3.3 shell .....	11
1.3.4 特权用户 .....	12
1.3.5 系统调用与库函数 .....	13
1.4 C 库 .....	15
1.4.1 头文件 .....	15
1.4.2 保留字 .....	17
1.4.3 特征测试宏 .....	18
1.5 系统信息 .....	19
1.5.1 机器标识 .....	19
1.5.2 硬件/软件类型识别 .....	21
1.6 系统能力限制 .....	22
1.6.1 一般能力限制值 .....	22
1.6.2 系统和文件特征选项 .....	26
1.6.3 sysconf, pathconf 和 fpathconf 函数 .....	28
1.7 错误处理 .....	32
<b>第 2 章 标准输入输出</b> .....	37
2.1 输入/输出基本概念 .....	37
2.2 流和 FILE 对象 .....	39
2.3 打开和关闭流 .....	40

2.4 读和写流 .....	43
2.4.1 字符 I/O .....	43
2.4.2 行 I/O .....	45
2.4.3 读回退 .....	49
2.4.4 块 I/O .....	50
2.5 文件定位 .....	52
2.6 文件结束和错误 .....	57
2.7 流缓冲 .....	58
2.8 格式 I/O .....	64
2.8.1 格式输出 .....	64
2.8.2 格式输入 .....	70
2.9 临时文件 .....	75
<b>第 3 章 低级输入/输出 .....</b>	<b>79</b>
3.1 文件描述字 .....	79
3.2 open、creat 和 close 函数 .....	80
3.3 文件输入输出 .....	84
3.3.1 read 函数 .....	84
3.3.2 write 函数 .....	85
3.4 设置描述字的文件位置 .....	87
3.5 dup 和 dup2 函数 .....	90
3.6 fdopen 和 fileno .....	91
3.7 文件控制函数 fcntl .....	92
3.7.1 重复文件描述字 .....	93
3.7.2 文件描述字标签 .....	94
3.7.3 文件状态标签 .....	95
3.8 非阻塞 I/O .....	99
<b>第 4 章 文件与目录 .....</b>	<b>102</b>
4.1 文件 .....	102
4.1.1 stat、fstat、lstat 函数 .....	103
4.2 文件类型 .....	105
4.2.1 普通文件 .....	105
4.2.2 目录 .....	106
4.2.3 链接与 link 函数 .....	106
4.2.4 符号链接与 symlink 和 readlink 函数 .....	108
4.2.5 特别文件 .....	110
4.2.6 测试文件的类型 .....	112
4.3 文件的用户和用户组 .....	113
4.3.1 chown、fchown 和 lchown 函数 .....	116
4.4 文件方式 .....	118
4.4.1 文件访问权限 .....	118
4.4.2 调整用户 ID 和调整组 ID .....	121



---

4.4.3	sticky 位	122
4.4.4	文件方式位小结	123
4.5	确定和改变文件方式	124
4.5.1	umask 函数	125
4.5.2	chmod 和 fchmod 函数	126
4.5.3	access 函数	128
4.6	文件大小	130
4.6.1	截断文件	132
4.7	文件时间	134
4.7.1	utime 和 utimes 函数	135
4.8	文件的删除与换名	138
4.8.1	删除文件和目录	138
4.8.2	文件改名	140
4.9	目录操作	142
4.9.1	工作目录	142
4.9.2	创建目录	145
4.9.3	读目录流	146
4.9.4	对目录流的随机访问	149
<b>第 5 章</b>	<b>UNIX 进程环境</b>	<b>151</b>
5.1	main 函数	151
5.2	命令行参数	151
5.2.1	命令行参数的语法约定	152
5.2.2	扫描命令行参数中的选项	153
5.3	环境变量	156
5.3.1	环境表	156
5.3.2	访问环境	158
5.4	终止进程	160
5.4.1	出口状态	160
5.4.2	终止前的清理	161
5.4.3	流产程序	162
5.5	存储分配	163
5.5.1	进程的地址空间	164
5.5.2	简单存储分配	165
5.5.3	释放分配的存储单元	168
5.5.4	其他存储分配函数	169
5.6	setjmp 和 longjmp 函数	170
5.7	进程资源	175
5.7.1	查看与设置资源限制	175
5.7.2	资源使用统计	178
5.7.3	进程优先权	179
5.8	用户信息	183

---

5.8.1 用户名 .....	183
5.8.2 用户数据基 .....	185
5.8.3 组数据基 .....	188
<b>第 6 章 进程控制</b> .....	<b>193</b>
6.1 进程创建 .....	193
6.2 执行一个新程序 .....	199
6.3 等待进程完成 .....	204
6.4 进程终止与僵死进程 .....	210
6.5 system 函数 .....	214
6.6 进程组 .....	217
6.7 会晤期 .....	219
6.8 控制终端 .....	220
6.9 作业控制 .....	222
6.9.1 作业控制的概念 .....	223
6.9.2 实现作业控制的 shell .....	224
6.9.3 示例 shell 中用到的数据结构 .....	226
6.9.4 shell 的初始化 .....	227
6.9.5 发送作业 .....	228
6.9.6 放置作业于前台或后台 .....	231
6.9.7 暂停的和终止的作业 .....	233
6.9.8 继续被暂停的作业 .....	235
6.10 进程的用户身份 .....	236
6.10.1 调整进程的身份 .....	238
<b>第 7 章 日期与时间</b> .....	<b>246</b>
7.1 处理机时间 .....	246
7.1.1 clock 函数 .....	246
7.1.2 times 函数 .....	248
7.2 定时与睡眠 .....	252
7.2.1 设置定时器 .....	252
7.2.2 睡眠 .....	256
7.3 日历时间 .....	256
7.3.1 取系统时间 .....	257
7.3.2 高分辨率时间函数 .....	258
7.3.3 分解的日历时间 .....	260
7.3.4 格式化日期与时间 .....	263
7.4 时区 .....	267
7.4.1 用 TZ 环境变量指定时区 .....	267
7.4.2 Tzset 函数与相关的变量 .....	269
<b>第 8 章 信号处理</b> .....	<b>270</b>
8.1 信号概念 .....	270

8.2	UNIX 信号	273
8.2.1	程序错误类信号	274
8.2.2	程序中止类信号	275
8.2.3	闹钟类信号	277
8.2.4	I/O 类信号	277
8.2.5	作业控制类信号	277
8.2.6	操作错误类信号	279
8.2.7	其他信号	279
8.2.8	sys_siglist[] 数组与 psignal 函数	280
8.3	指定和改变信号的动作	282
8.3.1	signal 函数	282
8.3.2	进程初启时的信号动作	285
8.3.3	不可靠信号	286
8.3.4	sigaction 函数	287
8.4	信号句柄	292
8.4.1	正常返回的信号句柄	292
8.4.2	终止进程的句柄	293
8.5	生成信号	294
8.5.1	raise 函数	294
8.5.2	kill 函数	295
8.6	阻塞信号	297
8.6.1	信号集操作	298
8.6.2	信号屏蔽	299
8.6.3	检查悬挂信号	302
8.7	等待信号	304
8.7.1	pause 函数	304
8.7.2	sigsuspend 函数	305
8.8	使用分开的信号栈	308
8.9	信号句柄编程技巧	312
8.9.1	句柄内非局部控制转移	313
8.9.2	可重入函数	317
8.9.3	紧相邻的多个信号	319
8.9.4	被信号中断了的系统调用	322
8.9.5	原子数据访问	324
8.10	SA_SIGINFO 标志	326
<b>第 9 章</b>	<b>低级终端 I/O</b>	<b>332</b>
9.1	需要改变终端设置的例子	332
9.2	低级终端 I/O 基础	335
9.2.1	终端 I/O 的历史	335
9.2.2	终端硬件模式	336
9.2.3	异步串行通信	337

9.2.4	加工和非加工输入方式 .....	339
9.2.5	输入输出队列 .....	339
9.2.6	流控制 .....	341
9.3	POSIX.1 GTT 控制界面 .....	341
9.3.1	termios 数据结构 .....	341
9.3.2	GTT 控制函数 .....	343
9.4	终端属性标志 .....	344
9.4.1	输入方式 .....	344
9.4.2	输出方式 .....	346
9.4.3	控制方式 .....	347
9.4.4	局部方式 .....	348
9.4.5	特殊字符 .....	350
9.4.6	stty 命令 .....	353
9.5	终端标识 .....	354
9.6	获取和设置终端属性 .....	358
9.7	加工方式输入 .....	360
9.8	非加工方式输入 .....	362
9.9	设置波特率 .....	369
9.10	行控制函数 .....	370
9.11	串行端口程序设计 .....	373
<b>第 10 章</b>	<b>高级 I/O .....</b>	<b>377</b>
10.1	文件锁 .....	377
10.1.1	fcntl 文件锁操作 .....	378
10.1.2	锁的请求、释放和测试 .....	381
10.1.3	文件锁与进程和文件的关系 .....	389
10.1.4	死锁 .....	390
10.1.5	建议锁与强制锁 .....	392
10.2	流 .....	394
10.2.1	流概貌 .....	394
10.2.2	流消息类型和优先级 .....	397
10.2.3	访问流 .....	399
10.2.4	流控制操作函数 ioctl .....	402
10.3	信号驱动的 I/O .....	404
10.4	多路转接 I/O .....	406
10.4.1	select 函数 .....	407
10.4.2	poll 函数 .....	410
10.5	异步 I/O .....	412
10.5.1	异步 I/O 控制块 .....	413
10.5.2	指定信号 .....	414
10.5.3	异步 I/O 函数 .....	414
10.5.4	异步 I/O 之例 .....	417

---

10.6	L/O 方式小结 .....	420
10.7	readv 和 writev .....	424
10.8	存储映射 I/O .....	425
<b>第 11 章</b>	<b>进程间通信 .....</b>	<b>433</b>
11.1	管道 .....	433
11.1.1	创建管道 .....	434
11.1.2	父子进程间管道通信 .....	435
11.1.3	连接标准输入和标准输出的管道 .....	440
11.1.4	popen 和 pclose 函数 .....	441
11.1.5	管道 I/O 的原子性 .....	443
11.2	FIFL 特别文件 .....	443
11.2.1	创建 FIFO .....	444
11.2.2	FIFO 操作 .....	446
11.2.3	FIFO 用于客户/服务通信 .....	450
11.3	系统 V IPC .....	455
11.3.1	关键字和标识 .....	456
11.3.2	ipc_perm 结构 .....	457
11.3.3	ipcs 和 ipcrm 命令 .....	458
11.4	消息队列 .....	459
11.4.1	初始化消息队列 .....	462
11.4.2	控制消息队列 .....	464
11.4.3	发送和接收消息 .....	467
11.5	信号量 .....	471
11.5.1	信号量的初始化 .....	473
11.5.2	信号量控制 .....	475
11.5.3	信号量操作 .....	478
11.6	共享存储 .....	483
11.6.1	共享存储段的访问和控制 .....	484
11.6.2	共享存储段的连接和分离 .....	487
<b>第 12 章</b>	<b>套接字与网络通信 .....</b>	<b>493</b>
12.1	套接字 .....	493
12.1.1	套接字的域、类型和协议 .....	493
12.1.2	创建套接字 .....	495
12.1.3	关闭套接字 .....	498
12.2	套接字地址 .....	499
12.2.1	主机的 IP 地址 .....	499
12.2.2	主机名 .....	502
12.2.3	服务与端口号 .....	506
12.2.4	套接字地址数据结构 .....	509
12.3	字节顺序 .....	511
12.4	协议数据库 .....	513

---

12.5	命名套接字 .....	516
12.6	套接字通信模式 .....	519
12.7	流套接字操作 .....	521
12.7.1	请求连接 .....	522
12.7.2	接收连接 .....	524
12.7.3	多客户 .....	528
12.7.4	getsockname 和 getpeername .....	534
12.7.5	send 和 recv .....	537
12.8	套接字选项 .....	539
12.9	带外数据 .....	541
12.9.1	TCP 带外数据 .....	541
12.9.2	带外数据的发送和接收 .....	543
12.9.3	带外数据标志 .....	547
12.10	数据报套接字操作 .....	550
12.10.1	sendto 和 recvfrom .....	551
12.10.2	数据报套接字客户/服务之例 .....	552
12.10.3	使用 connect .....	558
12.11	超时处理 .....	559
<b>附录</b>	<b>函数索引 .....</b>	<b>562</b>

# 第 1 章 UNIX 导论

在这一章,我们将首先讲述 UNIX 的诞生、成长和发展历程,介绍 UNIX 发展过程中出现的若干标准。回顾 UNIX 的历史,有助于我们了解它具有如此强大生命力的原因所在,并把握它未来的发展方向。UNIX 在发展中经历了由一个实验室的版本分裂为系统 V 和 BSD 两大阵营,并产生了好几种标准的过程。尽管目前在公认的 X/Open 标准下逐渐归于统一,但是仍然有些系统为向下兼容而保持非标准的特征。了解 UNIX 的这些标准,可以使我们理解和区分 UNIX 不同版本之间的区别,并编写出可移植性更好的程序。

然后,作为后继章节的铺垫,我们将简要讲述 UNIX 的一些基本概念并介绍与 UNIX 全系统有关的一些内容,例如系统信息、系统能力限制以及错误处理等。

## 1.1 UNIX 简史

在计算机发展史上,UNIX 是一个“历史悠久”的操作系统,早在 MS DOS、Windows 之前它就诞生了,到现在已经有 30 年的历史。在开始讲述 UNIX 环境程序设计方法之前,回顾一下它的历史和演变过程有助于我们更全面地了解 UNIX。

### 1.1.1 UNIX 的诞生

1965 年麻省理工学院的 MAC 课题组和通用电气公司一起开始了一个项目——开发一个新的称为 Multics 的多用户、交互式操作系统。Multics 的目的是向大的用户团体提供同时访问、计算和存储能力。在当时批处理系统为主流的时代,这是一个创新的概念。此后不久,贝尔实验室的计算科学研究中心也加入了这一计划。但在 1969 年,这个研究组认为开发 Multics 需要更长的时间,于是贝尔实验室退出了这个项目,Multics 也随之终止。

当 Multics 不再进行时,贝尔实验室的一些主要开发人员仍在继续研究另一个感兴趣的课题。他们中的一员 Ken Thompson 用 Fortran 编写了一个游戏程序“太空旅行”(Space Travel),这个游戏通过太阳系中行星的自转和公转模拟行星的位置和运动,并提供一个宇宙飞船可以来往于各个行星。

与此同时,Thompson、Dennis Ritchie 和贝尔实验室的其他人也在试图改善他们的程序设计环境,并提出了一种新的文件系统设想。此后不久 Thompson 写了一个内核并编写了模拟这个文件系统的程序,在 GE-645 计算机上将之实现。这个新的文件系统允许 Thompson 在类似于 Multics 的层次结构目录中存储他的游戏源代码文件。但是 Thompson 发现这仍不能满足“太空旅行”游戏的需要,因为 GE-645 是一个效率不高的分时系统,游戏的回

应很慢而且运行一次太贵(约 \$ 75),这迫使 Thompson 寻找另一台机器。他找到了一台闲置不用的 PDP-7。这是一台 18 位的机器,有 4096 字的内存和一台电传打字机。虽然在当时这并不是唯一适合运行该游戏的机器,但是它很廉价,并且有较好的图形显示器,然而它的程序设计环境和开发环境却不太好。因此 Thompson 不得不在另一台机器(GECOS 系统)作交叉汇编,然后再将程序记录在纸带上输入到 PDP-7。

在完成了“太空旅行”之后,Thompson 和 Dennis Ritchie 决定开始为 PDP-7 开发一种操作系统环境。Thompson 在这个环境中实现了他以前设计和模拟过的文件系统,该文件系统之后演变为著名的系统 V 文件系统(s5fs)的早期版本。不久,他们又加入了进程子系统、一个简单的命令解释程序(它后来演变成 Bourne shell)和一组管理文件系统的实用程序,并且实现了对两个用户分时使用的支持。当 PDP-7 上的汇编可以使用后,这个系统变成了自支持的而不再需要 GECOS 环境。

由这项工作诞生了 UNIX 的第一个版本,这就是最早的 UNIX 汇编版本,尽管当时这个操作系统还没有命名为 UNIX。一开始时,贝尔实验室该小组的成员 Brian Kernighan 幽默地将该操作系统取名为 UNICS,以隐喻它是一个两个用户的系统,同时,这也是“Uniplexed Information and Computing System”的缩写,因此它也是隐喻 Multics 项目的双关语。1970 年,这个系统正式取名为 UNIX。

### 1.1.2 UNIX 的早期发展

1971 年,UNIX 的第一个汇编版本被移植到了 PDP/11-20,它的第一个真正用户是贝尔实验室的专利部门。这次移植增加了几个文本处理实用程序,包括 ed 编辑器和 nroff 文本格式处理系统。

1971 年 11 月,Ritchie 和 Thompson 出版了《UNIX Programmer's Manual》第 1 版。这个手册整整出版了 10 版,分别对应于贝尔实验室发布的 10 个 UNIX 版本。

不久之后,1972 年发行了 UNIX 第 2 版。这个版本加入了用 B 语言写的管道和内核。B 语言是一种解释语言,因此受到了性能不高的困扰。最后,Ritchie 将 B 进化成 C 语言。C 语言是随 UNIX 诞生的,但它的成功却大大超过了 UNIX 本身。

前面几个版本严格限制在贝尔实验室内部。第 3 版出现在 1973 年 2 月,它包含了 C 编译器 cc。同一年,UNIX 用 C 重新写过,于 1973 年 11 月产生了第 4 版。这次重写为 UNIX 的可移植性打下了良好的基础,对 UNIX 后来的成功有着巨大的影响。

UNIX 发展过程中,一个重要的里程碑是 1973 年 10 月 Thompson 和 Ritchie 共同撰写的第一篇关于 UNIX 的论文:“The UNIX Time Sharing System”,它发表在《ACM Symposium on Operating System》杂志上,并于 1974 年 1 月发表在《Communication of ACM》上。这篇论文第一次让外界看到了 UNIX 的面貌,它标志着 UNIX 系统的突起。世界各地的学者开始对这个新的操作系统发生了兴趣,并导致了对 UNIX 软件和源程序代码的巨大需求。不久,UNIX 升级为第 5 版,这个版本在简单的许可协议下可以自由获取,包括给大学用于研究和教育目的的源代码。在这种情况下,UNIX 迅速地传播至全世界。第一个得到 UNIX 许可协议的是加州大学伯克利分校,它在 1973 年 12 月得到了 UNIX 系统。到 1975 年 UNIX



就已经扩散到了远至以色列的 Hebrew 大学和澳大利亚的 New South Wales 大学。这为后来增强和开发 UNIX 的各种项目铺平了道路。

而与此同时,贝尔实验室的另一个小组,程序工作组(Programmer's Workbench [PWB])则开发了增强了对大用户集支持的另一个版本。1975 年贝尔实验室第一次通过 Western Electric Company 对外发行了 UNIX 第 6 版,也称为 V6。这个版本进一步增强了对大用户集的支持。

UNIX 版本 V7 发行于 1979 年 1 月。这是第一个真正可移植的 UNIX 系统,并且在很大程度上影响了 UNIX 的后继发展。它包含了一个 C 编译器,这个编译器称为 PCC (Portable C Compiser),即可移植编译器,一个称为“Bourne Shell”的命令解释器,以及其他许多特征。

### 1.1.3 BSD UNIX

在 UNIX V7 期间,UNIX 已发展成熟为一个可运行于许多不同处理机上的相当稳定的操作系统。Bell 实验室广泛使用 UNIX,但没有任何技术服务。其父公司 AT&T 因为美国法律的原因,也没有提供服务(受 1956 年美国司法部对 AT&T 以及 Western Electrol 公司反托拉斯诉讼协议的约束,该协议的有效期限阻止 AT&T 生产任何与电话或电报无关的设备或从事其他非公共载体通信服务的商务)。这便为伯克利大学打开了一扇通往 UNIX 领域杰出领导者之门。

加州大学伯克利分校于 1974 年 12 月得到了第一个 UNIX 许可。在随后的几年中,贝尔实验室研究组的一些成员,包括 Ken Thompson 也利用休假年在那里教 UNIX,并参加一些研究工作。伯克利大学的研究生和教授中,很多人对 UNIX 系统产生了极大的兴趣,其中包括 Bill Joy 和 Chuck Holey。他们为这个 UNIX 开发了一些实用程序,包括一个 Pascal 编译器和一个 ed 编辑器,这个编辑器后来成为著名的 vi 编辑器。Bill Toy 将这些新增的内容与一些广泛发布的软件收集在一起打成一个软件包,形成了“Berkeley Software Distribution”(BSD),并在 1978 年春季将它以每个许可协议 50 美元售出。同年晚些时候又推出了 2BSD。

据说前面两个 BDS 版本只包含应用和实用程序,并没有修改也没有包含 UNIX 操作系统,第一个包含操作系统的版本是 1979 年发布的 3BSD。3BSD 运行于 32 位的 VAX-11/780,这个版本在内核加入了页式请求和虚拟内存等新功能。3BSD 虚拟内存功能引起了国防高级研究项目部 DARPA(the Defense Advanced Research Project Agency)的注意,并决定为伯克利 UNIX 系统的开发提供基金。DARPA 项目的主要目的之一是集成 TCP/IP 网络协议包。在 DARPA 的资助下,UNIX 开始蓬勃发展。1980 年伯克利产生了后来统称为 4BSD 的几个版本,4.0BSD(1980)、4.1CSD(1981)、4.2BSD(1983)、4.3BSD(1986)以及 4.4BSD(1993)。

伯克利 UNIX 的研究工作是在计算机科学研究组 CSRG(Computer Science Research Group)进行的。4.4BSD 之后,由于 UNIX 系统变得越来越大,以致很难由一个小组来维护和发展,同时也由于经费问题,CSRG 决定不再继续 UNIX 的开发。