

UNIX 操作系统 原理与应用

刘日升 孙玉方 编著

能源出版社

UNIX 操作系统 原理与应用

刘日升 孙玉方 编著



能源出版社

前 言

UNIX系统的产生是最近十几年来计算机操作系统发展中最重要成就。它对计算机界的影响可以比得上第一个可移植的高级语言FORTRAN,也可比得上第一个具有广泛性能的可兼容的计算机系列IBM/360。UNIX系统已经成为操作系统发展的一个里程碑。

UNIX在许多领域里都起着重要的作用。

在工业界,UNIX系统实际上已经成了16位以至32位微机操作系统的一个标准。作为一个可移植的操作系统,它已运行在各种不同类型的机器上。小到以Z80和M6809为CPU的8位微型机,大到32位的机器,如VAX-11/780,3B系列,IBM/370,IBM/4300,Honeywell Level 6,Univac V77,Amdahl470/V7,甚至于巨型机Cray-II等等。最流行的是以MC68000、Intel8086和Z8000为CPU的16位微机系统以及UNIX标准版本所运行的PDP-11系列。

在商业市场上,UNIX及其变种在16位微机系统中占统治地位,并进入8位机和32位机的市场。目前市场上出现的UNIX及其变种有:UNIX(MC68000,Z8000),XENIX(Z8000,Intel8086,MC68000)ZEUS(Z8000),Onyx(Z8000),Venix(MC68000),Uniplex(MC68000),Coherent(Z8000),IDRIS(MC68000,PDP-11),Jos(MC68000),Unos(MC68000),Cromix(Z80),Uniflex(MC6809),UTS(Amdahl VM/370)等等。

在大学里,由于UNIX短小精悍,包含了许多先进的设计思想和实现技术,并且是用高级程序语言编写的,从而使计算机操作系统的实例教学成为可能。在美国,90%以上的大学使用UNIX系统进行教学。在我国,高等理工院校的计算机系也已经或即将以UNIX为背景进行操作系统原理的教学。

在学术研究领域里,UNIX系统作为软件工程的支撑环境越来越受到重视。由于UNIX拥有丰富的软件工具和实用程序,以及对用户来说有良好的可适应性,从而为软件的开发提供了很好的工具系统和环境。1980年美国国防部为它的Ada语言提出了一个环境要求,即Stoneman计划。由于UNIX简单和通用,国内外已有不少机构把UNIX作为Ada语言环境的基础而进一步开发。UNIX被称为语言环境中的PASCAL。

近几年来,我国进口了一大批运行UNIX系统的微机系统。随着UNIX系统的不断普及,对它的应用逐步推广,势必要求对UNIX系统进行更深入的了解。UNIX系统本来是为程序员而使用的。它的资料主要是二卷三册的《UNIX Programmer's Manual》(UNIX程序员手册),其中大部分是UNIX作者和其它开发者的技术论文。近几年来国外出版了不少有关如何使用UNIX系统的书籍,我们也陆续翻译和编写了有关这方面的资料。但这些资料很少涉及到UNIX系统的内部结构和实现原理,因而在使用系统时,由于不了解它的外部用法与内部实现的关系,而大大限制了UNIX系统的深入开发。

根据国内各方面人员的要求,自1981年以来,我们先后在北京、成都、武汉、西安、大连、海天等地多次举办了UNIX系统学习班,主要讲解UNIX系统的实现原理。为了满足广大计算机用户、科技人员和高校师生学习UNIX系统的要求,为了进一步提高UNIX系统的应用和开发水平,我们根据近年来对UNIX系统的研究结果和教学经验编写了这本自学教

材。目的在于使广大未系统学过计算机操作系统原理的读者通过学习本书，既能掌握操作系统的基本原理，又能较深入地了解 UNIX 系统的内部实现过程，从而在使用 UNIX 系统时能更有效地使用它的 shell 命令和程序设计技术。

为了达到上述目标，我们在编写本书时着重注意了以下三点：

第一，注意通俗性和系统性。为了使未学过操作系统的读者比较容易接受 UNIX 的一些稍复杂的概念，书中适当安排了一些基本概念和预备知识。另一方面又对系统的程序模块作一些剪裁，忽略了一些与基本原理关系不大的实现细节，而较多地采用例子和图表对基本原理加以通俗的说明。

第二，本书主要讲授 UNIX 系统的实现原理。这主要体现在系统的核心程序中。但我们着重介绍核心中那些与用户程序设计和 shell 命令关系比较密切的部分，并说明它们与用户的关系，以便使读者在使用这个系统时不仅知其然，而且也知其所以然，从而更有效地利用 UNIX 的能力。

第三，关于 UNIX 的使用方法（包括 shell 语言、C 语言和程序设计环境），着重用例子说明，而不过多地涉及它们的语法细节。

本书共分十二章。其中第一章和第二章为基础部分，约占全书的 5%；第三章至第九章为系统原理部分，约占 70%；第十章至第十二章为使用部分，约占 25%。下面将各章的具体内容简介如下：

第一章介绍 UNIX 的历史和特征。

第二章针对初学的读者，介绍计算机软件和操作系统的一些基本知识。

第三章介绍支持标准 UNIX 系统的 PDP-11 机器的硬件结构、中断系统和 UNIX 系统的总体结构。

第四章开始正式涉及 UNIX 系统的原理。第四章与第五章分别介绍 UNIX 的进程管理和存储管理。

第六章介绍时钟管理和捕俘管理，其中还包括系统调用、软中断机构和跟踪机构的实现。

第七章叙述外部设备管理，包括块设备和字符设备，并分别用 RK-11 磁盘和 KL-11 控制台终端设备为例加以说明。

第八章介绍文件系统。

第九章为系统初启，说明被研制好的 UNIX 核心的可执行目标代码如何装入内存并把整个系统初始化。

第十章介绍命令程序设计语言 shell 的主要用法和实现思想。

第十一章介绍 UNIX 的基本语言，即 C 语言。

第十二章介绍 UNIX 程序设计环境，主要包括标准函数和系统调用的用法。本章可看成是上一章的继续。

由于水平所限，时间仓促，书中难免会有这样那样的缺点和不当之处，欢迎计算机界各方面人士批评指正。

目 录

第一章 UNIX操作系统的历史与特征	(1)
1.1 UNIX系统的发展历史与现状	(1)
1.1.1 UNIX系统的产生	(1)
1.1.2 UNIX系统的发展	(2)
1.1.3 UNIX系统的现状	(2)
1.2 UNIX系统的特征	(4)
1.2.1 UNIX系统的主要优点	(4)
1.2.2 对UNIX系统的批评	(6)
第二章 操作系统的基本知识	(8)
2.1 硬件与软件	(8)
2.2 操作系统的功能	(9)
2.3 核心程序与实用程序	(10)
2.4 操作系统的分类	(11)
习题.....	(12)
第三章 UNIX系统结构	(13)
3.1 PDP-11系统结构	(13)
3.2 中断系统	(14)
3.2.1 中断概念的引入与分类	(14)
3.2.2 中断与中断向量	(15)
3.2.3 捕俘与捕俘向量	(17)
3.2.4 中断保留区	(18)
3.2.5 中断捕俘总控程序	(19)
3.3 UNIX系统的程序结构	(20)
3.3.1 程序的调用与返回	(20)
3.3.2 再谈中断保留区	(22)
3.4 UNIX系统的整体结构	(23)
习题.....	(25)
第四章 进程管理	(26)
4.1 进程的概念	(26)
4.1.1 进程概念的引入	(26)
4.1.2 进程的状态	(28)
4.2 进程映象	(29)
4.2.1 进程的程序与数据	(29)
4.2.2 proc结构与ppda	(30)
4.3 进程调度	(33)
4.3.1 调度算法	(33)

4.3.2	进程现场的保留	(34)
4.3.3	低级调度程序	(35)
4.4	进程通信	(37)
4.4.1	概述	(37)
4.4.2	通信操作的实现	(37)
4.4.3	进程通信举例	(39)
4.5	进程的建立	(40)
4.5.1	建立进程映象	(41)
4.5.2	子进程的运行	(42)
4.5.3	系统调用fork	(43)
4.6	执行一个文件	(45)
4.6.1	更换进程映象	(45)
4.6.2	可执行文件的格式	(45)
4.6.2	自变量传递	(46)
4.6.4	exec的实现	(48)
4.6.5	fork-exec的应用	(50)
4.7	进程的终止与等待	(51)
4.7.1	进程终止	(51)
4.7.2	等待子进程	(53)
4.8	进程的资源	(55)
习 题		(56)
第五章 存储管理		(57)
5.1	引 言	(57)
5.1.1	逻辑地址与物理地址	(57)
5.1.2	连接编辑	(58)
5.1.3	地址重定位	(59)
5.1.4	内存的扩充	(59)
5.2	PDP-11的地址转换机构	(60)
5.2.1	页表	(60)
5.2.2	地址转换算法	(62)
5.2.3	用户栈的表示	(63)
5.3	逻辑空间的分配	(64)
5.3.1	用户逻辑空间	(64)
5.3.2	核心逻辑空间	(65)
5.3.3	样本页表	(66)
5.3.4	样本页表与硬件页表的装配	(67)
5.3.5	estabur与sureg的应用	(70)
5.4	物理空间的分配	(70)
5.4.1	分配算法	(70)
5.4.2	应用举例	(72)

5.5 程序对换	(74)
5.5.1 概述	(74)
5.5.2 对换驱动程序	(74)
5.5.3 映象换出程序	(76)
5.5.4 进程 0	(77)
5.6 有关数据段的操作	(78)
5.6.1 改变数据段的大小	(78)
5.6.2 用户栈的扩大	(79)
5.6.3 系统调用brk	(79)
习题	(82)
第六章 时钟处理与捕俘处理	(84)
6.1 时钟处理	(84)
6.1.1 硬件时钟与软件计时	(84)
6.1.2 时钟处理程序	(85)
6.2 捕俘总控与系统调用	(86)
6.2.1 捕俘总控程序	(86)
6.2.2 系统调用的转入与参数复制	(87)
6.2.3 系统调用的分类	(91)
6.3 软中断	(92)
6.3.1 软中断的引入	(92)
6.3.2 软中断处理程序的设置	(93)
6.3.3 软中断的产生与发送	(94)
6.3.4 软中断的发现与处理	(95)
6.3.5 软中断处理举例	(100)
6.4 跟踪	(101)
6.4.1 跟踪机构的用法	(101)
6.4.2 跟踪机构的实现	(102)
6.4.3 跟踪过程	(106)
习题	(108)
第七章 外部设备管理	(109)
7.1 引言	(109)
7.1.1 块设备与字符设备	(109)
7.1.2 主设备号与次设备号	(109)
7.1.3 UNIX设备管理的特点	(110)
7.2 设备配置与开关表	(111)
7.2.1 设备的重新配置	(111)
7.2.2 开关表与驱动程序的联系	(111)
7.2.3 特别文件的建立	(114)
7.3 块设备的缓冲区管理	(116)
7.3.1 块设备系统的结构	(116)

7.3.2	缓冲区与缓冲首部	(116)
7.3.3	块设备表与缓冲首部队列	(118)
7.3.4	缓冲区的申请与释放	(119)
7.4	块设备的驱动	(112)
7.4.1	硬件背景	(112)
7.4.2	RK磁盘的驱动	(114)
7.4.3	块设备系统与文件系统的接口	(127)
7.5	字符设备的缓冲管理与tty结构	(131)
7.5.1	引言	(131)
7.5.2	缓冲管理	(131)
7.5.3	终端设备的tty结构与寄存器组	(133)
7.6	终端设备的驱动	(137)
7.6.1	终端的打开与关闭	(137)
7.6.2	终端输入过程	(138)
7.6.3	终端输出过程	(141)
7.7	原始磁盘的管理	(147)
7.7.1	原始磁盘的用法	(147)
7.7.2	原始磁盘的启动	(147)
	习题	(148)
第八章	文件系统	(151)
8.1	引言	(151)
8.1.1	文件与文件系统	(151)
8.1.2	文件的逻辑结构与物理结构	(142)
8.1.3	UNIX文件系统的特 点	(143)
8.2	文件目录	(144)
8.2.1	简单的文件目录	(154)
8.2.2	多级目录与UNIX树形目录结构	(155)
8.2.3	索引节点	(157)
8.2.4	文件的联结	(151)
8.2.5	目录的检索	(162)
8.3	文件卷的安装与拆卸	(163)
8.3.1	实现原理	(163)
8.3.2	实现过程	(166)
8.4	文件卷存储空间的管理	(167)
8.4.1	一些常用的方法	(168)
8.4.2	文件卷的专用块与空间块的管理	(168)
8.4.3	i节点的分配与释放	(172)
8.5	文件的访问与存取控制	(174)
8.5.1	文件的访问与共享	(174)
8.5.2	文件的存取控制	(176)

8.6 关于文件的系统调用.....	(177)
8.6.1 数据结构.....	(177)
8.6.2 文件的创建.....	(179)
8.6.3 文件的联结与删除.....	(182)
8.6.4 文件的打开与关闭.....	(185)
8.6.5 用户打开文件表、系统打开文件表与活动i节点表的关系	(187)
8.6.6 文件的读写.....	(188)
8.6.7 管道文件.....	(197)
习题	(202)
第九章 系统初启	(204)
9.1 初始引导.....	(204)
9.2 核心初始化.....	(205)
9.2.1 核心页表寄存器的初置.....	(205)
9.2.2 进程 0 和进程 1 的建立.....	(207)
9.2.3 核心初始化过程.....	(210)
9.3 系统初始化.....	(210)
9.3.1 单用户环境.....	(210)
9.3.2 若干文件.....	(211)
9.3.3 多用户环境与文件getty	(212)
9.3.4 进程 1 的生命周期.....	(213)
习题	(214)
第十章 shell命令语言及其程序设计	(215)
10.1 UNIX使用基础	(215)
10.1.1 注册(login)与注销(logout)	(215)
10.1.2 终端特性	(216)
10.2 编辑程序	(216)
10.2.1 ed的进入与退出	(217)
10.2.2 建立文件	(217)
10.2.3 行编辑	(218)
10.2.4 上下文编辑	(220)
10.2.5 模式匹配	(222)
10.2.6 全局编辑	(223)
10.2.7 其它用法	(223)
10.3 常用的shell命令	(224)
10.3.1 有关目录和文件的命令	(224)
10.3.2 文件打印命令	(227)
10.3.3 状态询问命令	(230)
10.3.4 用户通信	(231)
10.3.5 进程管理	(233)
10.4 shell的交互特征	(234)

10.4.1	后台执行与重新定向	(234)
10.4.2	管道与过滤器	(235)
10.4.3	文件名字生成	(235)
10.4.4	引号的使用	(236)
10.5	shell程序设计	(236)
10.5.1	shell变量	(236)
10.5.2	几个专用的shell变量	(238)
10.5.3	简单的条件	(239)
10.5.4	简单命令、管道线与命令表	(240)
10.5.5	if条件	(240)
10.5.6	shell程序的自变量	(241)
10.5.7	while和until条件循环	(242)
10.5.8	命令替换	(243)
10.5.9	for结构	(244)
10.5.10	case结构	(245)
10.5.11	break和continue语句	(245)
10.6	shell程序的执行	(246)
10.7	shell解释程序的实现	(248)
10.7.1	初始化	(248)
10.7.2	命令行的读入与分析	(249)
10.7.3	命令树的执行	(250)
第十一章	C语言	(255)
11.1	C语言的演变历史及其特点	(255)
11.1.1	C语言的演变历史	(255)
11.1.2	C语言的特点	(256)
11.2	C程序的编写及其运行	(257)
11.2.1	样本C程序	(257)
11.2.2	C程序的处理流程	(260)
11.3	C语言的基本成分	(263)
11.3.1	C程序的元素	(263)
11.3.2	基本数据类型	(264)
11.3.3	常量	(265)
11.3.4	运算符与表达式	(266)
11.4	语句与控制流	(270)
11.4.1	若干简单语句	(271)
11.4.2	条件语句与开关语句	(271)
11.4.3	循环语句	(275)
11.4.4	其它语句	(277)
11.5	函数与程序结构	(278)
11.5.1	函数的一般形式	(278)

11.5.2	函数的递归调用	(280)
11.5.3	程序结构与存储类	(281)
11.5.4	初始化	(284)
11.6	构造数据类型	(285)
11.6.1	数组	(286)
11.6.2	指针与数组	(287)
11.6.3	结构	(289)
11.6.4	联合	(293)
11.7	C 预处理程序	(293)
第十二章	UNIX程序设计环境	(296)
12.1	自变量约定	(299)
12.2	输入输出程序库	(297)
12.2.1	输入输出标准函数	(297)
12.2.2	格式输入输出	(298)
12.2.3	文件的存取	(299)
12.2.4	错误处理与出口	(302)
12.2.5	成行的输入输出	(303)
12.3	其它函数	(303)
12.4	低级输入输出	(304)
12.4.1	文件的读写与打开关闭	(304)
12.4.2	文件的随机存取	(307)
12.4.3	fopen的实现	(307)
12.5	进程管理	(309)
12.5.1	低级进程执行——execlp 和 execlvp	(309)
12.5.2	创建并等待子进程——fork和wait	(311)
12.5.3	进程终止——exit	(313)
12.6	信号处理	(314)
参考文献		(317)

第一章 UNIX操作系统的历史与特征

1.1 UNIX系统的发展历史与现状

1.1.1 UNIX系统的产生

UNIX系统是由美国电话电报公司(AT&T)下属的Bell实验室的两名程序员K·汤普逊(Ken Thompson)和D·里奇(Dennis Ritchie)于1969~1970年研制出来的。这是一个交互式的分时系统。虽然Bell实验室不是一个出售计算机软件的团体,但由于UNIX的可移植性及其简单、通用和精巧的设计,使得它首先占领了学术界和教育界,之后又在短短几年内占领了工业界和商业界,成为现今16位微机事实上的标准操作系统。UNIX为什么能取得这样巨大的成功呢?首先让我们来回顾一下它的产生背景与发展历史。

到60年代中后期,计算机操作系统已经发展到比较成熟的阶段。当时国际上最大的计算机公司IBM就推出了一个空前庞大的操作系统OS/360(用去5000人年)。由于该系统过于复杂而不可靠,用户使用也不方便,因而受到广泛的批评。1965年,美国麻省理工学院(MIT)、通用电气公司(GE)和Bell实验室开始共同研制一个大型分时系统MULTICS(MULTiplexed Information and Computing Service),该系统用去200人年,于1969年开始交付使用。Bell实验室的K·汤普逊和D·里奇参加了这一工程。

MULTICS仍然过于庞大而复杂,未能达到它原定的野心勃勃的目标,然而它在设计和实现中提出了许多有价值的思想和技术。这对后来的操作系统发展产生了很大的影响。该系统是最早提出和实现进程概念的系统之一,它所实现的段页式存储器是当时最先进的技术。它首次提出和实现了软硬件结合的保护环的存储保护技术。对于外部设备,它向用户提供了与设备独立的、简单抽象的接口。尤其重要的是,它所实现的功能完善的分级结构文件系统为后来的许多操作系统所效法。它还为用户提供了有程序设计能力的命令语言shell。在设计方法上,MULTICS采用了结构化的方法,又采用高级程序设计语言PL/I编写了其中90%的代码。在研制MULTICS的过程中采用了MIT的CTSS为研制工具,而后又将MULTICS用作自己的研制工具。

1969年,Bell实验室的K·汤普逊和D·里奇退出MULTICS工程。由于他们感到自己没有有一个可用的分时系统,于是就在一台废弃的机器PDP-7上建立一个操作系统,以便能够在程序设计的研究环境中支持一批程序员协调一致地工作。他们于1970年完成了这个操作系统,这就是UNIX的第一个版本。这个版本开始只有汇编语言和几个实用程序,之后他们又为资料准备软件系统进行了进一步地开发。

UNIX系统继承和发展了MULTICS系统的许多成功的经验,例如,分级结构的文件系统,与设备独立的用户接口,功能完善的命令程序设计语言,采用高级语言作为系统研制工具。然而,UNIX的规模比起MULTICS来大大缩小了。MULTICS用去200人年,而UNIX只用了2个人年。为了反映它与MULTICS又继承又背离的关系,UNIX的作者把它取名为UNIX,其中UNI和MULTI相对立,而X与CS的读音相同。

1983年,Bell实验室UNIX系统规划部主任拉里·克雷姆(Larry Crame)就UNIX的成功

这一事实说道：“UNIX的成功并非来自什么崭新的设计概念，而是由于对操作系统所应具备的功能作了一番仔细的斟酌。也就是说，要确定赋予它哪些功能，而且更重要的是，要确定放弃哪些功能。过去的操作系统常常由于庞杂而带来许多问题，有所失才能有所得。UNIX的成功就在于它恰当地作了选择。”

1.1.2 UNIX系统的发展

1970年，汤普逊实现了B语言，用它重写了UNIX系统。这就出现了UNIX第二版。由于B是一种解释性语言，速度慢，1971年里奇开始研制C语言。

1971和1972年又分别推出了UNIX的第三版和第四版。

1973年UNIX系统用C语言重写，其规模增加了1/3，并提供了多道程序设计功能。这就是UNIX第五版。

同年，在第四届ACM操作系统原理会议上，里奇和汤普逊发表了题为“The UNIX Time-Sharing System”的论文。UNIX从此开始为外界所认识。

这篇论文修改后发表在1974年7月的CACM杂志上，之后它又获得ACM协会1974年最优论文奖。

UNIX操作系统的产生不是为了商业上的目的，主要在Bell系统内部使用，后来逐渐引起了一些大学和研究机构的兴趣。到1975年，Bell实验室又推出了UNIX第六版，同属AT&T的西方电气公司对这一版本颁发了许可证。但对于大学和其它非赢利的机构只象征性地收取少量费用，以鼓励它们使用和开发。就从这时开始，美国加利福尼亚大学伯克利（Berkeley）分校对UNIX系统进行的独立开发，与Bell系统以后的版本相平行。该校研制了一系列版本，它们均以BSD（Berkeley Software Distribution）为标志。

在Bell方面，1979年推出第七版，1981年推出System III，1983年推出System V。

在伯克利方面，1977年推出1.0BSD，以后又陆续推出2.0BSD（1978），3.0BSD（1979），4.0BSD（1980），4.1BSD（1981），4.2BSD（1983）。

从80年代开始，商业界看到了UNIX的价值，他们也开始研制自己的UNIX版本。这些版本基本上都是以第七版为基础的，例如Xenix、Onyx、Vnix等。

鉴于UNIX对于计算机工程所作的重大贡献，1983年美国ACM协会授予UNIX的两名作者里奇和汤普逊以图灵（A.M.Turing）奖。这是国际计算机界所公认的关于计算机科学与工程的最高奖赏。

到1983年为止，UNIX各种版本的发展历史如图1-1所示。

1.1.3 UNIX系统的现状

对UNIX系统的各种开发带来的一个不利后果是版本的不统一，这给UNIX系统的进一步开发和应用带来很大困难。为了解决这一问题，国际微处理机市场的四大巨头——Intel、Motorola、Zilog和National Semiconductor公司在AT&A所属西方电气公司的监督下联合开发UNIX的System V版本。它们以这个版本为标准来装配各档的芯片、插件、组件产品和开发系统。虽然其它工业标准操作系统还继续使用，但System V将成为这些公司微机的实际标准操作系统。

Intel公司在iAPX-286上采用System V，同时也采用Xenix。

Motorola公司已经提出了为它的EXORmacs开发系统设计的System V版本，即System

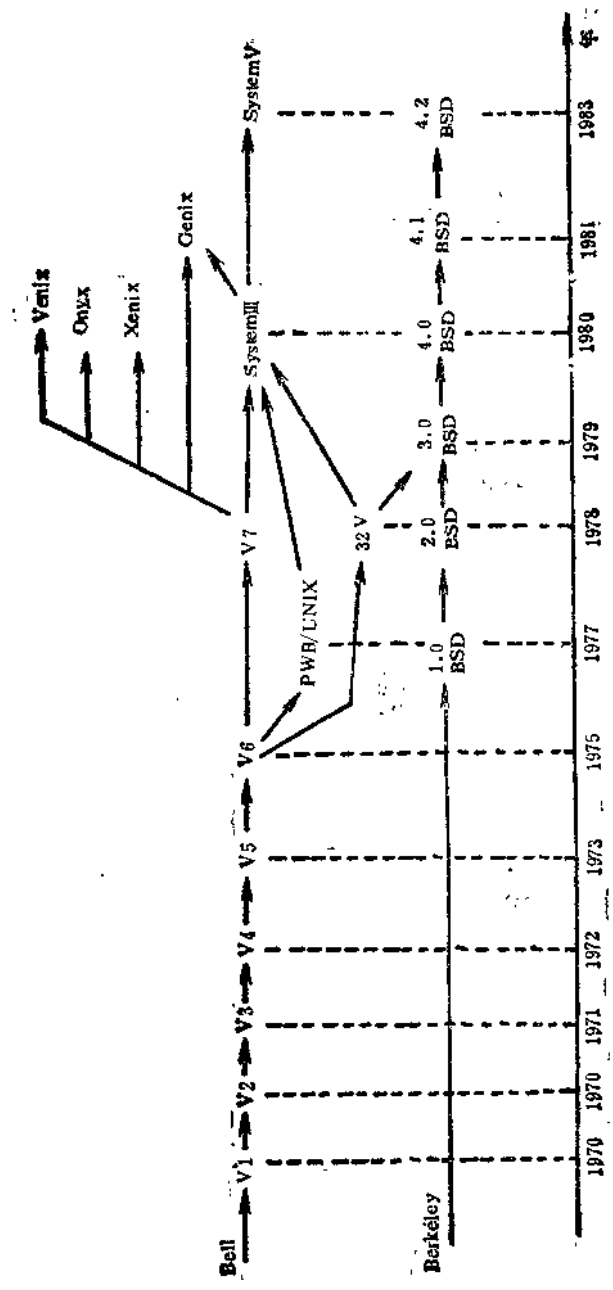


图 1-1 UNIX系统的演变

V/86。

Zilog公司准备把System V用于Z8000和以后的Z80000上。

National公司将在NS16000系列上运行System V。

AT&A公司甚至亲自出马进入计算机市场。在经过重大改组之后，AT&T于1984年1月获准经营计算机工业。同年，Bell实验室就生产出32位的CPU芯片WE32000，并以此构成AT&T的微型、小型计算机系列，即3B系列。AT&T以其强大的财力和技术力量为后盾成为国际上较大的计算机公司，成为IBM公司最大的竞争对手。AT&T的竞争王牌依靠的就是它的UNIX系统。在3B系列上已经安装了最新的UNIX版本System V，该系列已投入国际市场（包括我国市场）。

这四家公司与AT&T合作开发UNIX标准版本，对UNIX的标准化无疑会产生重大影响。其它一些打算继续开发自己操作系统的公司不得不在自己系统的未来版本上加上UNIX的特征。作为8位微机上的标准操作系统的CP/M就从UNIX中吸取了不少特征。CP/M的开发者Digital Research公司宣布在它的Concurrent CP/M-86的未来版本中将加入更多的UNIX特征。开发了Xenix的Microsoft公司除了继续推广Xenix外还宣布在它的另一个操作系统MS-DOS的未来版本中将加入更多的UNIX特征。就连实力强大的IBM公司也于1984年初宣布在PC个人计算机上配上了UNIX的最佳版本PC/IX。

可以预料，由于市场的吸引和程序员的推动，UNIX会以这种或那种形式存在和发展，而它的主要特征和新的思想将会对今后软件的发展产生深刻影响。

1.2 UNIX系统的特征

1.2.1 UNIX系统的主要优点

UNIX系统问世以来，受到来自各个方面的称赞。有人说UNIX的最重要优点是它的可移植性；有人说它成功的关键是分级构造的文件系统；有人说UNIX的最主要优点是简单通用；有的认为UNIX的最宝贵的财富是它的200多个实用程序；有的说UNIX的主要特点在于它是一个可剪裁的系统，适于各种用户的个别需要；也有的认为UNIX软件工具的思想是它的重要贡献。

美国ACM协会图灵奖评选委员会在对UNIX两位作者颁发1983年图灵奖时，对UNIX作出了如下评价：

“UNIX系统的成功在于它对少数关键思想所作的恰如其分的选择和精美的实现。UNIX系统关于程序设计的新的思想方法成了整整一代软件设计师的楷模。UNIX系统的天才就在于它为程序员提供了一种可以利用他人工作成果的机制。”

具体地说，根据UNIX作者的观点，UNIX系统的主要优点在于以下六个方面。

第一，分级结构的文件系统。

文件系统有一个根目录，在它之下可以有若干文件或其它目录。每个目录下面又可以有文件或目录。如此无限延伸下去。UNIX鼓励用户把他的一组文件或目录组织在一个目录里。文件系统为不同用户的文件提供了共享和存取控制的能力，保证用户之间安全有效的合作。

UNIX的用户还可以把与UNIX文件系统相容的文件系统（卷）安装到已有的文件系统

中的某个目录上，从而形成一个更大的文件系统。用户就象访问原有文件系统一样访问自己安装上去的文件系统。用户也可随时卸下自己的文件系统。

第二，简单划一的文件概念。

文件是无结构的字节序列。文件不分记录，也没有存取方法。这种做法大大减小了系统的规模。在用户面前，文件的概念简单了，使用也方便了。当然，用户如果需要的话，可以为文件建立自己需要的结构。文件也没有顺序文件和随机文件之分。在缺省情况下文件都是顺序存取的，但用户可通过改变读写指针而对文件实现随机存取。

UNIX 系统对文件概念的最大改进可以说是把普通文件、目录和外部设备统一在文件之下。在许多操作系统中，由于目录含有关于文件的重要信息，所以要由操作系统隐蔽和保护。而在UNIX中，目录是可由任何程序读取的文件。

UNIX把外部设备当作文件处理是对外部设备所作的最重要的抽象。外部设备就如同磁盘上的普通文件那样被访问、共享和保护。用户不必知道设备的物理特性就能访问它。例如行式打印机对应的文件名是/dev/lp。用户只要用写文件的操作（write）就能把它的数据从打印机上输出。

第三，建立异步进程的能力。

UNIX系统为用户在终端上打入的每条命令都建立一个子进程，以执行对应的实用程序。如果用户在命令后而打入符号“&”，则这个子进程就要与后面建立的子进程并发地运行。由于UNIX维护一个进程的代价很小，所以操作系统的许多功能都可以从核心中移出去作为普通进程运行。这就使得一个大型的程序可以通过与它对应的进程与其它程序合作。例如若干程序可以用管道线连接起来使得前一程序的输出就是后一进程的输入。

第四，使用灵活的命令程序设计语言shell。

shell这一名称来自MULTICS。shell首先是一种命令语言。UNIX的200多条命令对应着200个实用程序。它们可以用管道线连接起来。不仅如此，shell还是一种程序设计语言。它具有许多高级语言所拥有的控制流能力，如if、for、while、until、case等语句，以及对字符串变量的赋值、替换、传递参数、命令替换等能力。用户可以利用这些功能用shell语言写出“shell程序”存入文件。以后用户只要打入相应的文件名就能执行它。这种程序易于阅读和修改，很容易用于系统的扩充。

与其它操作系统的命令解释程序不同的是，shell是一个普通的程序，而不是操作系统的一部分。甚至用户可以在进入系统之前预先指定自己的命令解释程序。由于它的这一特点，用户可以在UNIX的任一子系统中直接引用shell命令而不必退出（然后再进入）这个子系统。例如在编辑程序ed、调试程序Adb或C程序中都可以执行shell命令。命令执行完毕，用户仍然处在原有子系统中原来的位置上。

第五，UNIX系统支持十几种高级语言和200多个实用程序。

UNIX支持的高级语言有C、FORTRAN77、PASCAL、APL、SNOBOL、ALGOL68、COBOL、BASIC等。UNIX的实用程序小而简洁，可以把它们组合起来完成复杂的工作。所以比起其它系统中许多庞大而复杂的实用程序来要容易维护得多。在UNIX实用程序中，有一些其它系统常见的实用程序，如编辑程序、调试程序、有关系统状态监控和文件管理的实用程序等。但UNIX还有一组强有力的软件工具，用户能够比较容易地使用它们开发可靠的软件。由于正文文件（ASCII码文件）是所有软件的原始形式，所以在UNIX的软件工具库中有许多是用于处理正文文件的实用程序。例如，用于正文格式加工的nroff和troff，用

于表格加工的tbl, 用于数学方程式加工的eqn, 用于大型程序开发的维护组合程序make, 源代码控制系统sccs, 命令语言的词法分析程序和语法分析程序的生成程序lex和yacc, 等等。

第六, 使用了C语言。

UNIX系统的几乎所有代码(包括系统核心的90%)都是用C语言写的。这使UNIX成为一个可移植的操作系统。以前的操作系统一直是汇编语言的阵地, 但这种语言是依赖于机器的。用C语言编写的UNIX系统容易修改和移植。1976年, 里奇和S.约翰逊(S. Johnson)只用两个月就把UNIX从PDP-11移植到Interdata/32上, 以后UNIX又被迅速移植到十几种机器上。操作系统的可移植性带来了应用程序的可移植性, 因而用户的应用程序既可用于小型机又可用于其它的大型机和微型机, 从而大大提高了用户的工作效率。

C语言与UNIX是互相依赖的。UNIX是用C语言写的, C语言又是在UNIX支持下运行的。目前C语言已经成为可与PASCAL相比的通用的程序设计语言。

1.2.2 对UNIX系统的批评

虽然UNIX系统取得了巨大的成功, 但是如同任何事物一样, UNIX系统也不是没有缺点的。在这一方面, 不少文章已经指出了。在批评者所指出的问题中, 有的确属UNIX设计的缺点, 有的是UNIX迅速发展所带来的, 也有的是属于批评者的观点。概括起来, 对UNIX的批评有以下几种:

第一, UNIX的版本太多而不统一, 因而应用程序的可移植性不能完全实现。

在UNIX的发源地Bell实验室, 先后产生了十几个版本, 即V1~V7、32V、System III和System V。与之平行, 加州大学伯克利分校也提出了六个版本。八十年代以来, 许多软件公司又推出五花八门的UNIX变种。这种情况的出现一方面反映了UNIX受人欢迎的优点, 另一方面也反映了UNIX的设计目标。UNIX是用C语言写的, 容易修改和移植。UNIX鼓励用户用UNIX的工具开发出适合自己需要的环境, 这进一步增加了UNIX的版本。为了统一这些版本, Microsoft公司曾试图把Xenix作为16位微机操作系统的标准。目前AT&T已与四家重要的微机厂家(Intel、Motorola、Zilog和National Semiconductor)合作制订统一的UNIX System V版本, 这将会给UNIX的标准化以重大影响。

第二, UNIX缺乏如下一些能力, 如实时控制、分布式处理、网络能力、文件记录结构等。

UNIX的主要设计宗旨是简单、通用。这是它得以成功的根本原因。与早期的大型操作系统, 如OS/360、MULTICS等相反, 它为用户的开发提供了一个良好的基础和环境。从目前大多数用户的观点来看, 这种系统更受欢迎。当然, UNIX也在不断发展, 例如提供实时功能、增加网络能力等。

第三, UNIX的核心是无序模块结构。

UNIX核心虽然90%是用C语言写的, 但由于其结构复杂, 要作较大修改也决非易事。近几年来, 国内外有不少机构试图对核心加以改造, 但保持它原有功能和对外接口。如有的借助于更结构化的语言(如并发Euclid、XCY), 有的借助于硬件功能(如微程序设计), 也有的就用C语言重写。我们也于1982年曾用XCY语言把UNIX系统的核心改造为层次的、模块化的结构。

第四, UNIX的用户界面复杂。