

第一章 UNIX / XENIX 系统概述

使用过计算机操作系统的人已经很少有人不知道 UNIX 了。全世界有 30 万个 UNIX 的用户单位,使用 UNIX 的人员超过 50 万,UNIX 产品的产值达 55 亿美元(软件占 10%)。到 1987 年为止,光是获得 UNIX 源代码专利的用户单位就有 6000 多个。据美国 Infocorp 公司调查,目前在美国已有包括 IBM 和 DEC 在内的 44 家大公司销售配有 Unix System V 的计算机,在 1987 年这类机器的销售量达 80 万台,而在 1986 年只有 24 家公司销售这类机器。据该公司预测,到 1992 年,在 12,000 到 350,000 美元的范围内,UNIX 机将占全世界计算机市场的 36%,而在 1986 年仅为 24%。在美国,90% 多的大学使用 UNIX,在这些学校的计算机科学系开设 UNIX 操作系统的课程。1986 年,配有 UNIX 系统的工程工作站已达到 65,700 个,占工程工作站市场的 91%。现在,从 s/09 这样的 8 位微型机、IBM PC / AT 及其兼容机到 IBM / 370 这样的大型机、CRAY 巨型机,从工程工作站到科学计算、事务处理、办公室自动化的各个领域,都有 UNIX 系统在运行。正因为如此,UNIX 已成为 1987 年计算机领域里四项最热门的技术之一。美国著名的市场研究公司 IDC 把 UNIX 活力的增强评为 1987 年美国计算机工业的十项大事之一,并把 UNIX 广为接受已成为定局作为 1988 年十项重要发展展望的第一项。专家们认为,UNIX 有可能成为操作系统的标准,成为多用户、多任务应用领域中的主流操作系统之一。

第一节 什么是 UNIX 系统?

UNIX 系统是称为软件的一组程序,它的作用就是把计算机和计算机的用户联系起来。UNIX 系统软件可以完成三项工作:

- 控制计算机的运行。
- 在用户和计算机之间作为解释程序。
- 提供一系列程序和工具包,使你能够完成自己的工作。

控制计算机运行的 UNIX 软件称为操作系统。简单地说,操作系统的作用就像是一个教室中的教师。这个教师负责布置作业、安排各种仪器设备的使用并协调和管理学生的各种活动。操作系统的作用就是协调计算机内部的工作,指挥各个部件的操作。根据计算机内部的时钟,操作系统作出各种调度的决定,例如什么时候应该发送信息给打印机,什么时候可以装人和运行用户的程序等等。UNIX 系统的核心(操作系统)称为内核(kernel)。

在 UNIX 系统中,在用户和计算机之间起联系作用的软件称为外壳(shell)。shell 对用户的要求进行解释,如果这些要求是合理的,就从计算机的存储器中取出程序,然后执行。

使用户能够完成自己工作的 UNIX 系统软件称为应用程序(application)或称为实用程序(utility),也有人把它称为工具(tool)。这些实用程序包括编辑程序(使你能够编写、修改和存储文件及数据的程序)、电子邮件、各种程序设计语言(如 C、Pascal)等等。

UNIX 是由两种软件组成的。因此,UNIX 这个术语实际上涉及了操作系统和实用程序这两部分的内容。这本书主要介绍 UNIX 一些有用的实用程序的使用方法,对 UNIX 的操作系统部分只作简单描述。

操作系统也可以定义为计算机和计算机用户之间的连接。用户通过使用操作系统来使用计算机。操作系统的目的是使用户能够以更灵活、更容易掌握的方式控制计算机的资源。UNIX 系统软件的操作系统部分也和一般的操作系统(例如 PC-DOS)一样,能够完成:

- 建立和管理文件系统。
- 装入和执行用户的程序。
- 管理和控制输入/输出设备。

除了这三项基本功能外,UNIX 还提供了在一般的微型计算机操作系统中没有的功能,主要是:

- 提供了多用户分时的功能。这就意味着,在不同终端上的几个人可以同时使用计算机。计算机把时间单位分成许多时间小片。每个时间片只为一个用户服务,然后迅速地从一时间片转到下一个时间片,从为一个用户服务转到为另一个用户服务。由于计算机运行的速度极快,每个用户感觉不到他的工作有停顿或等待,就好像只有他一个人使用计算机一样。一个操作系统能够分时为几个用户服务取决于计算机能够每秒钟操作数百万次。慢速的计算机是不可能分时的。同样,快速的计算机在用户数目过多,或每个用户提出的任务过多时,当计算机把主要的时间花在从一个任务转换到另一个任务上,而实际上没有时间去完成这些任务时,系统的效率就会骤然下降。用户就会感到花在等待上的时间过长,以至不能忍耐。

PC-DOS 是属于单用户的操作系统,而 UNIX 则是多用户的操作系统,这正是这两个操作系统的主要区别。

- 提供多任务功能。这个特性使每个用户可以同时进行几项工作,允许用户按照他的要求给每个任务设置不同的优先权。

UNIX 操作系统就是一个多用户、多任务、分时的操作系统。

第二节 UNIX 系统的历史和发展

UNIX 系统已经渡过了它的孩提时代,正在生气勃勃、充满活力地步入它的青年时期。在我们学习使用 UNIX 系统之前,回顾一下 UNIX 系统发展的历史是很有益处的。

1969 年 3 月,美国电话与电报公司(AT&T)贝尔实验室(Bell Labs)的肯·汤普森(Ken Thompson)和丹尼斯·里奇(Dennis Ritchie)在数字设备公司(DEC)的 PDP-7 小型机上开始着手开发一个小型通用的分时系统。设计这个系统的目的是为了在廉价的小型机上开发一个良好的程序设计环境,供贝尔实验室内部使用。汤普森和里奇都参加过在通用电器公司的大型机 GE645 上运行的交互式分时系统 Multics 的设计工作。他们把 Multics 系统上的一些优秀成果都带到了新系统的设计工作中,如文件系统、与设备独立的输入/输出、进程控制和命令语言等,这些成果对后来 UNIX 系统的发展起着重要的作用。

1970 年,这个命名为 UNIX 的系统开始在 PDP-7 机上运行。当时只能支持两个人同

时使用。同年,贝尔实验室添置了 PDP 11/20 小型机。汤普森和里奇立即在这个新的机器上建立起 UNIX 系统,并且为了对贝尔实验室的专利组织的资料准备工作提供支持,他们对系统进行了进一步的开发。

1971 年 11 月,汤普森和里奇合作编写了称为第一版的 UNIX 系统程序设计文件。实验室的专利组织成了 UNIX 的第一个用户。除了管道技术外,现代 UNIX 系统的许多重要思想,包括文件系统、进程管理、系统接口和主要的实用程序在这个版本中都可以找到。特别值得一提的是,UNIX 系统从一开始就把程序设计环境和文本处理能力作为系统的两大基石,一直保持至今。

1972 年 6 月,在增加了管道技术后,诞生了 UNIX 的第二版。这时候的系统 and 实用程序仍然是用汇编语言编写的。

1973 年是取得重要进展的一年。里奇在 B 语言的基础上研制成了 C 语言。汤普森和里奇用 C 语言改写了 UNIX 系统,由此诞生了第五版(V5)UNIX。后来的事实证明,这次用 C 语言改写系统,对于把 UNIX 从一个机器移植到另一个机器,对于 UNIX 的迅速普及和推广是具有决定意义的一步。同年,汤普森和里奇合作写了“UNIX 分时系统”一文,提交给美国计算机协会(ACM)第四次操作系统原理讨论会,并于 1974 年 7 月正式发表在《ACM 通讯》杂志上。这是 UNIX 系统第一次正式向外披露。

1975 年 3 月,第六版(V6)UNIX 系统发表。在这两年中,美国各大专院校陆续配置了数字设备公司(DEC)的 PDP 小型机,这为 UNIX 系统的普及奠定了硬件基础。在这一年,AT&T 所属的西方电气公司(Western Electric)开始颁发使用 UNIX 的许可证,并对高等院校实行优惠,目的是鼓励他们使用并继续开发 UNIX 系统。教育界安装 UNIX 系统所需的费用大约只是政府和商业界所付费用的 2%。从这时开始,美国各大学开始陆续使用 UNIX 系统。到目前为止,使用 UNIX 系统的大学已达到 90%。许多大学在使用 UNIX 系统的过程中,不断地对它进行修改、扩充和再开发。最著名的是加利福尼亚大学伯克利分校开发的 4.1BSD 和 4.2BSD 等系统。到七十年代后期,一个支持 UNIX 系统的硬件、软件和相关服务的工业就逐渐形成了。

1976 年 6 月,西方电气公司对获得 UNIX 使用许可证的用户公布了第六版 UNIX 系统内核的源程序。同年,新南威尔士大学首次以 UNIX 作为操作系统课程的实例进行讲解,收到了良好效果。从那以后,美国大多数大学的计算机科学系都以 UNIX 作为操作系统课程的主要内容。现在,每年都有数千名在运行和修改 UNIX 方面有丰富经验的大学生从各个大学毕业,走向计算机领域的各个方面。业已证明,对学校实行优惠、公布 UNIX 源程序、开设 UNIX 系统课程等措施,对于 UNIX 系统的发展都起了决定性的作用。在这一年中,里奇和 S·约翰逊(S·Johnson)把 UNIX 从 PDP-11 移植到 Interdata 8/32 机(一种类似于 IBM370 机的 32 位计算机)上。这是第一次把 UNIX 移植到不同类型的机器上。从那以后,UNIX 已经成功地移植到几十种计算机上了。

1978 年,第七版(V7)UNIX 发表。

1979 年,约翰·李瑟(John Reiser)和汤姆·伦敦(Tom London)开发了在 VAX 11/780 上运行的 UNIX 32V。

1981 年,AT&T 发表了 UNIX System III (S3)。从这时开始,UNIX 不再采用版本 (Version)号排列,而改为按系统(System)号排列。

1983年,UNIX System V(S5)发表。同一年,汤普森和里奇因发明了UNIX系统而获得美国计算机协会(ACM)图灵奖。

进入八十年代以来,微型计算机大量涌现。1980年,为了满足微型计算机使用UNIX系统的需要,美国享有盛誉的软件公司Microsoft根据微型机的特点对UNIX系统第七版(V7)进行了修改和扩充,并易名为XENIX。这是在UNIX诞生十年之后,首次进入微型机市场。1983年,Microsoft公司根据UNIX系统Ⅲ(S3)相应地改写了XENIX,发表了XENIX 3.0。1984年,在IBM推出IBM PC/AT机后,XENIX开始有了更适合应用的场合。Microsoft及时发表了PC/AT XENIX 1.0,这是与UNIX系统Ⅲ相当版本。1986年,根据UNIX系统V,Microsoft公司发表了PC/AT XENIX V。另一家微型计算机软件公司Santa Cruz Operation也发表了SCO XENIX 5.2/2。1987年,AT&T公司和Intel公司联合推出UNIX系统V/386 3.0版。与此同时,Microsoft公司也发表了XENIX V/386。这将是同时在以Intel 80386为CPU的32位超级微机上运行的两种UNIX版本。据报道,到目前为止,XENIX系统已经发行了20万份。

XENIX与UNIX比较虽然有了一些差别,但是这些差别主要体现在操作系统软件上,实用程序软件差别不大,核内差别大而核外差别小。从用户使用的角度看,shell命令解释程序、基本命令和主要实用程序的用法几乎完全一样,这一点从UNIX和XENIX的命令手册中的许多命令能够通用就可以看出来。会用UNIX系统的人可以毫无困难地使用XENIX;会使用XENIX的人使用UNIX也很容易。这是因为XENIX就是在微型计算机上运行的UNIX,它们并不是本质上完全不同的两种操作系统。

本书以在IBM PC/AT及其兼容机上使用的Microsoft公司的PC/AT XENIX 1.0(相当UNIX系统Ⅲ)为依据,同时参考SCO公司的XENIX 5.2/2(相当UNIX系统V),向你介绍XENIX系统的基本使用方法。

第三节 UNIX系统的特点

UNIX系统在不太长的时间内取得很大成功的根本原因在于UNIX本身的性能 and 特点。

1. UNIX是一个分时、多用户、多任务的系统,这是UNIX有别于PC-DOS的根本特征。说它是分时的,即它不是一个实时的系统。尽管有些扩展的UNIX已经具有实时控制的能力,但是标准的UNIX系统版本是不包含这些扩展功能的,UNIX系统也不是为实时应用而设计的。

2. 内核和核外程序的有机结合。UNIX系统在结构上分成两大层:内核和核外程序,即我们在前边提到的操作系统和实用程序两部分。UNIX的系统设计特别注意了内核和核外程序的有机结合。内核向核外程序提供了强有力的支持;核外程序则以内核为基础,充分利用了内核的支持。两者结合起来作为一个整体,向用户提供了各种良好的服务。

3. 良好的用户界面。UNIX向用户提供了两种界面(或称接口)。一种是用户在终端上通过使用命令和系统进行交互作用的界面,称为用户界面;另一种是面向用户程序的界面,称为系统调用。

UNIX系统的用户界面就是操作系统的外壳shell,它起着命令解释程序和全功能编程

语言的双重作用。这种语言并不难学,使用方便、灵活。使用它不仅能提高在系统上进行工作的效率,而且便于扩充系统的功能。UNIX 系统的有些实用程序就是后来的用户用 shell 语言编写的。

系统调用是用户编写程序时可使用的界面。和很多操作系统只在汇编语言级向用户提供系统调用的接口不同,UNIX 不仅在汇编语言级,而且在程序设计语言 C 中提供了这种接口。这给程序员设计程序带来很大方便。

4. 树形结构的文件系统。UNIX 的文件系统是分级的树形结构,这一点和 PC-DOS 相同。但是 UNIX 的文件系统是可装卸的,这却是 PC-DOS 所不能的。它不仅能够扩大文件存储空间,而且有利于安全和保密。

5. 文件和设备统一处理。在 UNIX 系统中,普通的文件、文件的目录表和输入/输出设备都是作为文件统一处理的。它们在用户的面前有相同语法和语义,使用相同的保护机制。这样既简化了系统的设计又便于用户使用。

6. 丰富的核外系统程序。UNIX 系统的核外部分包含了非常丰富的软件开发工具、文本处理程序、高级语言处理程序和系统实用程序。据 AT&T 公司称,到 1986 年,在 UNIX 下运行的微型计算机软件包已达 1000 多个。正是这些应用程序大大加强了 UNIX 的功能,也给它增添了不少光彩。

7. 系统用 C 语言写成,具有良好的可移植性。和完全用汇编语言写成的 PC-DOS 不同,UNIX 系统的两部分软件基本上都是用 C 语言写成的,这使 UNIX 系统很容易移植。这不仅意味着 UNIX 系统易于移植到别的硬件系统上,例如从 Motorola 的 68 系列移植到 Intel 的 80 系列,而且在 UNIX 系统下开发的应用程序也易于移植到其它配置了 UNIX 的系统上去。这正是 UNIX 系统得以普及和取得成功的重要原因之一。

第四节 UNIX 系统的结构

我们从以下两个方面来看一看 UNIX 系统的结构。

一、UNIX 系统的基本结构

如前所述,UNIX 系统就件由两部分组成:操作系统软件和系统实用程序软件,即内核部分和核外部分。UNIX 系统的基本结构如图 1-1 所示。

UNIX 的内核(kernel)是常驻内存的部分,是不能由用户随意变化的部分,这就是通常所说的 UNIX 操作系统。它包括进程管理、存储管理、设备管理和文件系统管理四大管理技术和操作系统的其它重要技术。这一部分内容一直是美国、目前也是我国大专院校计算机科学系操作系统课程的实例和研究课题。UNIX 系统核心的主要特点就是短小、简洁、精干(第六版(V6)的核心才只有大约 40K 字节)。UNIX 的设计者对核心的功能作了大胆的取舍,使它既具有较强的功能,又不包罗万象、庞大繁杂。内核的主要算法经过反复推敲,其中的数据结构和程序都经过精心设计。这使得 UNIX 的核心已成为操作系统设计的典范。

对普通用户来说,UNIX 系统的核心部分是看不见的。这正是为了使用户能把精力集中于自己的工作,不必为计算机内部怎样工作而操心。

所有不必常驻内存的程序都从内核中分离了出来,而以核外程序的形式在用户环境下运行。用户使用 UNIX,主要就是使用这一部分。这里包括各种程序开发工具、高级语言处理程序、文本处理程序、文件和信息管理程序、电子通讯程序等许多实用程序。用户编写的经过编译、汇编、连接而成的可执行程序也属于这一类。UNIX 系统的这一部分软件远比核心部分大得多、丰富得多,并且是可以不断修改和扩充的。正是由于 UNIX 的用户不断地把一些优秀的实用程序加到 UNIX 系统的核外部分,才使得 UNIX 系统的功能不断加强,日益完善。

在 UNIX 系统内核和系统实用程序之间的是 UNIX 向用户提供的两种界面。系统调用是专供 UNIX 系统的程序员设计和开发应用程序时使用的。这些人是 UNIX 系统的高级用户,至少是对 UNIX 非常熟悉而又精通于程序设计的用户。不管是普通用户,还是程序员,都是以命令的形式通过使用核外的实用程序来使用 UNIX 系统的。对用户所发出的命令进行解释,并调系统内核的相应部分执行命令的是称为 shell 的命令解释程序。它是 UNIX 系统内核面向所有用户的接口。

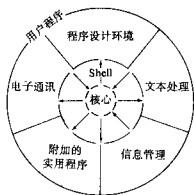


图 1-1 UNIX 系统的结构

从 UNIX 系统的基本结构图可以看出, UNIX 系统是由不同的层次、不同的部分组成的。不同的用户使用 UNIX 的不同的部分,重点放在不同的方面。对于大院校学习操作系统的人来说,他们研究的是 UNIX 系统的核心(kernel)部分;对系统程序员来说,使用的则是系统调用、shell 和实用程序部分中的高级语言处理程序和程序开发工具等;而对广大普通用户来说,使用的却是 shell 和其它系统实用程序以及用户自己开发的程序。本书是面向上述三种人的,并且由于是本人门书,而不是一本手册,书中的内容只能涉及 UNIX 系统实用程序中的一小部分,但却是常用的、基本的部分。

二、UNIX 核心的源程序结构

UNIX 系统内核的源程序代码约有一万行,分成 44 个文件。这些文件按编译方式大致可分成三类。

“.c”文件:这是 C 语言程序文件,共有 28 个文件,190 个子程序。虽然一个文件可能包含了若干个子程序,但是这些文件都是可以单独编译的。

“.h”文件: C 语言程序的全局变量、符号常数和数据结构文件。共有 14 个“.h”文件。这些文件只能和 C 程序文件一起编译。

“.s”文件:这是汇编语言程序文件,共约一千条,即占内核总程序量的大约 10%,分成 2 个文件,包含 33 个子程序。这些汇编语句大部分都与硬件关系密切,很难用 C 语言描述,或者是为了提高工作效率而没有用 C 语言描述。

上述内核源程序的情况只是美国公布的 UNIX 第六版(V6)的情况。

到 UNIX 第七版(V7)时,内核源程序代码已增加到约一万八千行(汇编程序约 1050

行)。程序文件增加到 77 个,其中“.s”文件 3 个,“.h”文件 33 个,“.c”文件 41 个。

以后各版本 UNIX 系统的情况也不尽相同。但总的趋势却是系统的内核逐渐加大了。

第五节 UNIX 系统的现状与未来

一、UNIX 的标准化

截止本书完稿之时,UNIX 系统的发展又取得了许多新的成就:

- Microsoft 和其它一些公司先后推出了 XENIX V/386 系统。
- AT&T 与 Microsoft 公司将 XENIX 系统 V 和 UNIX 系统 V 合并成新的 UNIX 系统 V/386 的 3.2 版。
- AT&T 公司宣布将在 1989 年推出 UNIX 系统 V 的 4.0 版。
- Apple 公司推出了相当 UNIX 系统 V 的 AU/X 系统。

除了这些最新成就外,UNIX 系统最引人注目的进展就是 UNIX 系统的标准化工作。

随着 UNIX 系统的迅速推广,与 UNIX 系统兼容的版本越来越多。这就使为 UNIX 开发的应用软件的移植工作出现了困难,也使用户难以选择。为了解决这一问题,早在 1980 年美国的 UNIX 用户协会就开始了 UNIX 系统的标准化工作。与此同时,美国的 IEEE 学会也成立了一个 p1003 标准化委员会。1985 年,这两个组织将 UNIX 系统的标准化工作合并起来共同开展。p1003 委员会在 UNIX 系统原有成果的基础上制订出了一个可移植的操作系统的规范,称为 POSIX 规范。POSIX 规范包括三个部分:

1. 1003.1 标准:可移植操作系统的规范。
2. 1003.2 标准:命令语言 shell 和其它工具。
3. 1003.3 标准:验证测试方法。

1986 年 4 月,IEEE 提出了“试用 POSIX 规范”的标准草案。到 1987 年 2 月,已有几十家大公司(包括 IBM 在内)和国际上很有影响的 x/open 组织都表示支持这个规范。同时,国际标准化组织 ISO 的 TC97 技术委员会正在准备以 IEEE 的 POSIX 规范为基础,制订一个操作系统的国际标准。美国国家标准局 NBS 也已提名将 POSIX 作为联邦信息处理的标准。现在,无论是新推出的还是即将推出的 UNIX 系统,几乎都毫无例外地宣布采用 POSIX 标准。这就将使以 POSIX 为标准开发的软件可以不加修改地用于不同来源的 POSIX 系统上。预计不久,正式的 POSIX 标准将获得通过并颁布推行。到那时,UNIX 将成为操作系统的工业标准。虽然今后 UNIX 可能会易名为 POSIX,但它的精华和实体将长期存在下去。

二、UNIX 系统的国产化

随着我国国内对 UNIX 系统的引进、使用和推广,我国已有相当一批专家和技术人员在 UNIX 系统上进行了多年的研究和开发工作,并取得了一系列成果。早在 1982 年,中科院软件所的技术人员就已对 UNIX 第 6 版进行了深入的分析和解剖,并写出了很有水平的分析报告。1983 年,合肥工业大学从 UNIX 第 7 版复原出 C 语言的源程序。1984 年,广

东省计算中心在 UNIX 系统基础上研制成 16 位多用户微型计算机汉字系统。1985 年,北京大学开发成功 C-XENIX 系统。成都电讯工程学院完成了 UNIX 第 7 版的分析与移植这一国家“六五”科技攻关项目,并复原出了 UNIX 系统源程序。航天部 710 所将 UNIX 系统移植到国产 LS-84 微型计算机上。同年,我国出版了第一本高等学校用的教材《UNIX 操作系统教程》,我国各大专院校的计算机系陆续开设了这门课程。1986 年,从 ALTOS、TOWER、DUAL、Dimension、3B2、Universe 一直到 IBM PC/AT,几乎任何一种引进的高档微型机上的 UNIX 系统的中文版本如雨后春笋般大量涌现。1987 年,中科院软件所完成了 UNIX 系统 V 和 XENIX 的汉化改造工作,他们推出的中西文兼容的 C-UNIX 系统 V 得到美国 AT&T 公司的认可和好评。AT&T 公司还与中科院签订了提供 UNIX 系统 V3.1 版源程序的协议。

现在,我国已经加入了国际 UNIX 用户组织,中国的 UNIX 用户协会也已成立。UNIX 系统国产化的工作正在逐步展开。

三、UNIX 与 OS/2

自从 IBM 公司 1987 年 4 月推出 PS/2 新机型以后,不仅原来的 PC 机市场产生了很大震动,而且对多用户的微型机市场也产生了一定影响。PS/2 机器上运行的 OS/2 操作系统会不会取代 UNIX? 这是人们普遍关注的一个问题。

尽管 PS/2 将要取代 PC 机,PS/2 上运行的操作系统 OS/2 将要取代 PC-DOS,但是 PS/2 仍然是面向个人使用的系统(Personal System),而不是多用户系统。IBM 的战略意图是要以它作为低档的产品,以便与 IBM 将来的小型机和大型机一起组成一个综合的向上兼容的系统体系。即便把 PS/2 用于多用户场合,它也只能像 PC/AT 那样带两个终端。而且由于 OS/2 是一个多任务而非多用户的操作系统,因此,当 PS/2 用于多用户时,它目前也只能运行 IBM PC/RT 机上的多用户操作系统 AIX,而 AIX 是一个基于 UNIX 的操作系统。同时,IBM 宣布将支持 UNIX 的工业标准。由此看来,PS/2 不会独占多用户的微型机市场,OS/2 不会取代 UNIX。IBM 在未来多用户微型机上运行的操作系统将不是与 UNIX 对立的,而可能是与 UNIX 一致的。

对于广大熟悉 PC-DOS 的用户应该怎么办? 我们都面临着一个适时转向的问题,或者转向多用户的 UNIX,或者转向单用户的 OS/2。如果你想加入到 UNIX 用户的行列中,应该何时转向? 从什么地方起步? 我们建议你从现在开始,从学习这本书起。

第二章 怎样开始

学习 UNIX 最好的方法就是使用 UNIX。与其看完书再试,或者没看书就试,不如边看书边试。本书中的每一条命令、每一个例子你都可以亲手试用一下。

在这一章中,我们将介绍怎样在 UNIX/XENIX 系统上开始工作,包括怎样进入和退出系统,怎样修改口令,怎样使用键盘上的一些特殊按键,以及几个简单的但是常用的命令。

第一节 注册和注销

现在要讲的是进入和退出 UNIX/XENIX 系统的方法。

一、注册(login)

如果使用的是 PC-DOS,开机后就可以直接使用了。但是使用 UNIX/XENIX 时却不同。因为有可能是几个人同时使用系统,每个人就必须有自己的工作环境,必须各自保存自己的文件。只有各就其位,才能各得其所,否则就乱套了。因此,使用 UNIX/XENIX 的人必须是该系统的合法用户。他(或她)必须有自己的用户名,这个名字在系统中是记录在案的。这就好像如果你是 UNIX/XENIX 系统这个“家庭”的一员,这个“家庭”的“户口本”上就应该有你的名字一样。

在 UNIX/XENIX 系统运行的过程中,如果打开一台与系统相连的终端,准备进入和使用系统时,屏幕上就会出现:

```
login:
```

这是系统在询问用户名。这个用户名可以是名字的汉语拼音缩写、小名、外号、笔名等等,并不一定是真名。这个名字是由系统管理员(这个“家庭”的“家长”)在为你建立一个单独的用户目录(这个工作称为建立用户)时,根据你的意见告诉系统的。这个名字和其他用户的名字一起,保存在一个名为“passwd”的文件中,这个文件驻留在系统的“/etc”目录中。这就是我们所说的“户口本”了。

如果你回答的名字能在“/etc/passwd”文件中找到,你就是这个系统的合法用户,就有可能进入并使用这个系统。如果在“/etc/passwd”文件中并没有你回答的名字,那你就被拒绝于系统的大门之外,因为在这个系统中并没有你的一席之地。

在注册回答用户名时,应当用小写字母键入(建立用户时只允许用小写)。如果使用了大写字母,或者无意中按下了<Caps Lock>(大写锁定)键,系统会显示出一个信息,建议你尽可能用小写字母,并请你再次注册。否则,系统将全部采用大写显示和操作,直到你注销退出系统。

回答用户名(以及下一节介绍的“口令”)进入系统的过程称为注册,也有人把它称为登录,其意思都是取自英文的“login”。这是进入和使用系统必不可少的第一步。

二、注销

当你在系统上结束工作以后,不能象在 PC-DOS 上那样关机了事。要时时想着你使用的是多用户的系统。你干完了,别人可能还在干着。你可以关机离开。但是,任何人打开你用过的终端时,他还是在你的目录之下。他可以不经任何注册过程,就在你的名下使用系统。因此,应当养成这样的习惯:每当工作结束时,就从系统中退出。退出系统的方法非常简单,只要按下 <Ctrl-d> 键(即同时按下“Ctrl”和字母“d”键)就可以了。屏幕上将重新出现“login:”,这时就可以关闭终端,然后离开。任何人要再次使用这个终端,都必须重新注册了。

有时候,当按下 <Ctrl-d> 键时,可能并没有出现“login:”。这往往是因为你进入了系统的某个程序或另外一种状态。例如,在使用电子邮件 mail 程序时,按下 <Ctrl-d> 键只能退出 mail 程序,如果想要退出系统,还需要再次按下 <Ctrl-d> 键。在一般情况下,<Ctrl-d> 键的作用是终止输入,通常每按一次 <Ctrl-d> 键,系统就会退出一层,直到出现“login:”,才退出了系统。

用户退出系统的过程称为注销(logout)。

第二节 设置和更改口令

一、回答口令

当对“login:”的询问回答了用户名并按下回车键时,注册过程还没有结束。屏幕上会接着出现:

Password:

这是系统请你回答口令,也有人称之为通行字或保密字。这是防止别人冒用你的名字进入系统的一种保护措施。如果没有口令的保护,任何人只要知道系统中任何一个用户的名字就可以进入系统。

口令也是系统管理员为你建立用户目录(用 mkuser 命令)时告诉系统的。如果那时是你自己输入口令的,那就有可能连系统管理员也不知道你的口令,除非你让他为你输入,或者你把自己的口令告诉了他。口令也保存在“/etc/passwd”文件中,但却是以一种加密的算法转换了形式显示出来的。下面就是“/etc/passwd”文件中的几行:

```
xiaod: j96sCiGhz6h2: 207: 51: : /usr/xiaodi: /bin/csh
student: j9lF2owJz/1f6: 210: 50: : /usr/student: /bin/sh
stu: 212: 50: : /usr/stu: /bin/sh
```

/etc/passwd 文件是每个人都可以看的,但是只有系统管理员才能修改。在这个文件中,每个用户占一行,一行中有若干项,每一项之间用冒号“:”分隔。第一项是用户名,这是用户注册时要由系统进行检查的项目。第二项就是口令,但是已经转换了显示形式,目的就是让人看不出来。如果口令这一项两边的冒号是连着的,如上边所列的第三行,说明该用户取消了口令保护,这种用户目录使用起来是不安全的。不要企图按照这个文件中显示的形式回答某个用户的口令,打算进入别人的目录。这样做是徒劳的。

在回答口令时,你所按的每一个键在屏幕上都不显示出来。这是为了防止有人在旁边偷偷看你的口令。当按错了一个键时,仍然可以用退格键删除这个看不见的但却是错误的字符,然后继续键入正确的字符。

如果你回答的口令不正确,系统就会显示:

```
Login incorrect
login:
```

系统并不是立即对你回答“login:”的用户名进行检查的,而是在你回答了口令之后对用户名和口令一起进行检查。因此,这两项中任何一项回答错了,系统都会给出一个出错信息,并再次显示“login:”,你就不得不重新开始注册过程。如果你始终回答不正确,就别想使用系统了。

二、修改口令

千万不要忘记了自己的口令,否则就不能进入系统了。如果真的出现了这种情况,可以请求系统管理员把 /etc/passwd 文件中你的口令项删掉,因为只有他可以修改这个文件。这样你就能够只回答用户名而直接进入系统了。进入系统以后,可以重新设置一个新的口令。

如果别人已经知道了你的口令,从而失去了口令的保护作用时,你可以随时用“passwd”命令修改口令。当键入“passwd”并按下回车键时,系统首先要问你老的口令。如果回答得不正确,系统就会显示出“Sorry”,你就不能修改它。在你正确地回答了老的口令以后,系统要让你两次键入新的口令。这是为了在键入一遍新口令之后,给你一次修改和确认的机会,也为了加深记忆。只要这两次键入的不一致,系统就要求你重新输入。直到连续两次键入的口令完全一致,系统便接受这个新的口令,并保存在 /etc/passwd 文件中。修改口令的过程如下所示:

\$ passwd	←“\$”是提示符,“passwd”是你发出的命令
Changing password for xiaodi	←由系统显示的信息,“xiaodi”是用户名
Old password:	←键入的字符并不显示出来
Enter new password (minimum of 5 characters)	
Please use a combination of upper and lowercase letters and numbers.	
New password:	←键入新的口令
Re-enter new password:	←再次键入新的口令
They don't match; try again.	←两次键入的口令不一致,系统要求重新键入。
New password:	
Re-enter new password:	
\$ █	←系统已接受新的口令,“█”表示光标。

UNIX / XENIX 系统默认要求口令的长度最少 5 个字符(可由系统管理员改变),最多 11 个字符,最好采用大小写字母与数字的组合。长度越长、组合越复杂的口令越不容易被入破密,使用系统也就越安全。

特权用户(即系统管理员)可以用 passwd 命令删除自己的口令,即当要求输入新的口令时,直接按回车键。新的口令是空的,他的口令自然就取消了。但是普通用户一旦设置了

要状态的提示符。

系统默认每个用户在标准 shell (Bshell) 下的提示符都是“\$”。但是这个提示符不是不可改变的。用一个编辑程序修改在你的注册目录下的“.profile”文件中的 `PS1=$`, 就可以把这个提示符改成别的符号。例如, 有的人就把这个提示符改成了“C>”, 以使人觉得好像他正在使用 PC-DOS 一样。但是, 在刚开始学习使用 UNIX / XENIX 系统时, 不要把精力放在这样的细枝末节上。

第四节 系统用到的特殊键

在开始工作之前, 首先让我们介绍一下在系统运行的过程中可能会用到的一些特殊键及其功能。

UNIX / XENIX 系统允许在一个命令行中使用多个命令。不管一个命令行中有几个命令, 最后都要以回车键(有的键盘标的是 Enter, 有的标的是 Return, 有的标的是 ↵)作为命令行的结束符。只有在按下回车键以后, 系统才能对命令行中出现的命令进行解释, 然后执行。在按下回车键之前, 可以对命令行中出现的按键错误加以纠正。

表 2-2 系统用到的特殊键及其功能

按 键	功 能
Return	执行刚键入的命令行
Ctrl-d	停止输入或注销
Ctrl-i	水平跳过一个制表区, 与<Tab>键作用相同
Ctrl-h	删除光标左边的一个字符
←	删除光标左边的一个字符
Ctrl-u	使刚键入的一行作废, 光标回到下行开始, 重新键入
Del	中断命令的执行, 与 PC-DOS 的<Ctrl-Break>作用相同
Ctrl-s	暂停屏幕的滚动显示
Ctrl-q	继续屏幕的滚动显示

表 2-2 是 XENIX 系统运行过程中可以使用的特殊键及其功能。

表 2-2 是针对监控(主控)台所用键盘的情况。当使用终端键盘时, 个别按键的情况可能与表中所列的情况不符。这或许是由于终端的仿真功能不够完善, 或许是由于终端把有些功能定义到其它键上了。例如, 当使用 GW220-CT 终端时, 如果想中断程序的执行, 就得按 F5(Break)键; 如果想继续滚屏显示文本内容, 就得按 F1 键, 等等。在这种情况下, 应当查阅一下终端说明书, 然后再决定如何使用。

上机练习:

- 1) 用“cat / etc / termcap”命令显示一个长文件, 试一试暂停屏幕滚动和继续屏幕滚动显示, 以及不等显示完就中断显示的按键功能。
- 2) 随便键入一行信息, 不要回车就使该行作废, 重新键入 passwd 命令, 不等回答口令

就终止该命令的执行。

3) 在回车前试一试纠正刚键入的前一两个字符。

第五节 简单的 shell 命令

现在一切都准备就绪, 可以开始使用 XENIX 系统了。首先介绍几个简单的但是常用的命令。

一、显示或设置当前的日期和时间(date)

在使用系统的过程中, 人们经常想要知道今天的日期或现在的时间。在 PC-DOS 中, 用 `date` 命令显示或设置日期, 用 `time` 命令显示或设置时间。在 UNIX / XENIX 系统中, 把日期和时间的显示或设置合成了一个命令—`date`。

1. 显示日期和时间

`date` 命令的最常用的用法就是只显示今天的日期和当前的时间。显示的情况如下:

```
$ date                                ←显示系统当前的日期和时间
Mon Feb 22 15:18:17 BJT 1988         ←北京时间 1988 年 2 月 22 日 星期一 15:18:17
$ █                                  ←命令执行完毕
```

`date` 程序首先以标准的英文缩写显示星期和月份, 然后是几号, 接着是以 24 小时制显示的时间, 精确到秒。随后出现的三个字母无关紧要, 它是你所在时区的名称的缩写, 是在系统安装过程中设置时区和时差时由系统管理员设置的。这里的“BJT”表示北京时间。最后显示的信息是年份。

由于安装 UNIX / XENIX 的系统几乎毫无疑问都装有电池为机器内的时钟电路供电, 只要电池有电, 机内的“电子表”是始终走着的。因此, 每次你使用 `date` 命令时, 它所显示的时间应当与当前标准的北京时间相吻合。

2. 设置日期和时间

如果 `date` 命令显示的时间已经不准确, 可以重新进行设置。日期和时间的设置不能分开进行。设置的方法全部采用数字式, 其格式如下:

```
date mmddhhmm[yy]
```

其中前两个 `mm` 是两位数字表示的月份, `dd` 是日, `hh` 是小时 (24 小时制), 第二个 `mm` 是分, 最后的 `yy` 表示年 (可以省略)。例如, 设置日期和时间为二月 22 号 15 点 50 分, 可采用如下形式:

```
$ date 02221550                      ←设置日期和时间
date: no permission                  ←没有修改的权利
Mon Feb 22 15:50:00 BJT 1988         ←这是你想设置的日期和时间
$ date                                ←显示系统当前的日期和时间
Mon Feb 22 15:20:26 BJT 1988         ←时间并未改变
$ █
```

上面显示的第三行就出现了出错信息, 从中可以看到 UNIX / XENIX 系统命令显示出错信息的一般格式。几乎所有 UNIX / XENIX 命令的出错信息都是首先显示出错的命

令名,在冒号之后接着显示有关的出错信息。了解 UNIX / XENIX 系统的各个命令在什么情况下会出现出错信息,对不同的出错情况应该用什么样的方法加以纠正,这也是使用好 UNIX / XENIX 系统命令的一个基本要求。

上面的出错信息表示你没有权利修改日期和时间,接着把你想要设置的日期和时间以标准的 date 格式显示了一遍。但是当你再次使用 date 命令时,会发现时间并没有改变。这是因为 date 命令只允许系统管理员设置日期和时间,而不允许普通用户随意修改。我们将在下边介绍怎样才能使自己成为系统管理员,但是用 date 命令设置日期和时间的方法你已经掌握了。

用 date 命令修改系统的时间有一个很大的局限性:用 date 命令设置的时间只在这一次系统运行的过程中起作用。下次开机时会发现,日期和时间并没有改变。要想从根本上改变系统的日期和时间,需要由系统管理员使用 setclock 命令,设置的方法与 date 命令设置的方法一样(请参考第十三章)。但是 setclock 命令也有一个缺点:用它设置的时间当时并不起作用,只有到下次启动系统时才有效。因此,就设置或修改系统的日期和时间这一功能来说,date 和 setclock 这两个命令的生效时间是不同的。date 命令是当时有效,关机失效;而 setclock 命令是当时无效,再开机才有效。这无形中给用户的使用带来了不便。

3. 改变日期和时间显示的格式

如果说 date 命令用于设置日期和时间并不好用,用它来改变日期和时间的显示形式却给用户带来了很大的灵活性。假如 date 命令的标准输出格式不合你的意,可以随意把它改成你所喜欢的形式。

改变 date 命令显示格式的方法是: date '+格式'。

在加号“+”之后就是由你所确定的格式。这个格式可以包含着若干个字段,每个字段必须用一个百分号“%”开头。在 date 命令的自定义格式中允许使用的字段如表 2-3 所示。例如:

```
$ date '+%a %h %d %T BJT 19%y/'
```

←改变 date 命令显示的格式

```
Tue Feb 23 13:52:59 BJT 1988
```

←显示的结果

```
$ █
```

你看,这不正是 date 命令缺省的格式吗?利用 date 命令的自定义格式,不仅可以随意安排显示的顺序和内容,而且可以显示更多一些的信息。在自定义格式中,可以随意插入空格和换行,也可以插入其它字符,例如上例中的“BJT 19”。但是字段描述符(表 2-3 中表示字段的字母)必须跟在字段开始符“%”之后,在它的前边不能插入空格。否则,date 命令将无法识别究竟是字段描述符还是要照原样显示的字符。例如:

```
$ date '+%D% j'
```

←在字段描述符“j”前多了一个空格

```
date: bad format character -
```

←格式字符不正确

```
$ date '+%D %j'
```

←正确的格式

```
02/23/88 054
```

←结果

```
$ █
```

上述第一个例子在字段标识符“%”和字段描述符“j”之间插入了一个空格,因此出现了出错信息“bad format character”,表示字段描述符不可识别。第二个例子在字段描述符“D”之后插入空格,问题就解决了(如果不插入空格,两个字段显示的内容就会连起来)。

在自定义格式中,每个字段所显示的数字的位数都是填满的,不足的部分补零。例如在

表 2-3 date 命令自定义格式中各字段的作用表

字段	显示的内容
n	在格式中插入一个换行符
t	在格式中插入一个制表符
m	显示年中的月份——从 01 到 12
d	显示月中的日期——从 01 到 31
y	显示年份的最后两位数字——从 00 到 99
D	显示月 / 日 / 年式日期, 形式为 mm / dd / yy
H	显示小时——从 00 到 23
M	显示分钟——从 00 到 59
S	显示秒钟——从 00 到 59
T	显示时 / 分 / 秒式时间, 形式为 HH:MM:SS
j	显示一年中第几天——从 001 到 365
w	显示星期几——从 0 到 6(星期日 为 0)
a	显示星期的英文缩写——从 Sun 到 Sat
h	显示月份的英文缩写——从 Jan 到 Dec
r	显示上午或下午——AM 或 PM

上例中, 表示今天是今年的第几天用的是 054 而不是 54, 表示月份用的是 02 而不是 2 月。

如果每次显示日期和时间时都要这样重新定义, 使用起来就太麻烦了, 这样也就失去了使用 date 命令自定义格式方法的意义。比较好的用法是: 把自定义的 date 命令的格式存在一个文件中, 例如太件名为 “d”, 然后把这个太件变成可执行文件。每当需要查看日期和时间时, 键入 “d” 并按回车键, 就能够按照预先定义的格式显示出所需要的信息, 如下所示:

```
$ cat d                                ←显示文件 d 的内容
```

```
date '+今天的日期: 19%y 年%m 月%d 日 %t 星期%w %t 今年的第 %j 天 %n 现在的时间: %T'
```

```
$ d                                    ←运行新的 d 命令
```

```
今天的日期: 1988 年 02 月 23 日      星期 2      今年的第 054 天
```

```
现在的时间: 15:40:05
```

```
$ █
```

这个例子是按照中国人的习惯以汉字显示日期和时间。如果所使用的系统已经过汉化, 这个例子在你的系统中也完全可以实现。

你一定会问: 怎样建立一个文件? 如何把文件变成可执行的? 在读完下两章之后自然会明白了。

二、显示日历 (cal)

UNIX / XENIX 系统提供了一个显示日历的命令。它可以显示公元 1-9999 年中任意一年或任意一个月的日历。这个命令的名字是 “cal”, 它取自英文 “calendar” (日历) 这个词的前三个字母。可以不带任何参数直接使用这个命令, 如下所示:

```
$ cal                                ←显示上个月、这个月和下一个月的日历
```

```
Mon Feb 22 15:57:23 1988           ←当前的日期和时间
```

Jan							Feb							Mar								
S	M	Tu	W	Th	F	S	S	M	Tu	W	Th	F	S	S	M	Tu	W	Th	F	S		
					1	2			1	2	3	4	5	6				1	2	3	4	5
3	4	5	6	7	8	9	7	8	9	10	11	12	13	6	7	8	9	10	11	12		
10	11	12	13	14	15	16	14	15	16	17	18	19	20	13	14	15	16	17	18	19		
17	18	19	20	21	22	23	21	22	23	24	25	26	27	20	21	22	23	24	25	26		
24	25	26	27	28	29	30	28	29	27	28	29	30	31									
31																						

\$ █

cal 程序首先显示出当前的日期和时间,然后显示出上一个月、这个月和下一个月这三个月的日历。月份和星期都是用英文的缩写表示的。

在 cal 命令之后可以跟着一个表示年份的数字,指定显示某一年全年的日历。这时将以年号取代当前的日期和时间,接着显示该年每一个月的日历。例如,“cal 1897”将显示 1897 年全年的日历。由于篇幅的关系,显示出什么情况留待你自己去观察。在指定年份时不能偷懒,如果想看 1988 年的日历,必须用“cal 1988”,不能用“cal 88”。这是因为“cal 88”将显示公元 88 年的日历,而不是公元 1988 年的日历。

也可以指定显示某一年某一月的日历。月份是 cal 命令允许指定的最小显示单位。指定的月份应在年份之前,中间用一个空格分开。指定月份时,可以使用数字式,也可以使用英文缩写式。用英文缩写形式时,首字母可以大写,也可以小写。用数字式时,月份数字后边必须指定年份。否则,指定的数字只作为年份解释。例如:

```
$ cal Mar 1953          ←显示 1953 年 3 月的日历
March 1953             ←结果如下
S M Tu W Th F S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
$ █
```

上述命令也可以用“cal mar 1953”或“cal 3 1953”,但是不能用“cal 3”或“cal Mar”表示。为什么?请想一想。

cal 程序是根据标准的格林威治时间推算日历的。因此,如果在系统安装时设置的本地时区距格林威治时间的时差不准确,将会使显示的日历不准。最好先用 cal 命令显示本月的日历核对一下,证明确实无误后,就可以放心地使用了。

现在,你可以借助于 cal 命令来查阅任何历史名人、你的家庭成员、亲戚朋友出生于星期几了。

三、谁在系统上(who, who am I, logname)

如果你想用“write”命令以通信的方式同某一个用户对话(详见第八章),首先必须知道