

第一部分

综　　述

第一章

BSD 系统的历史和目标

1.1 UNIX 系统的历史

20 多年来，UNIX 操作系统被广泛应用在计算机界的各个领域。不计其数的公司和团体为 UNIX 系统的发展和完善而努力，从而使 UNIX 系统涌现出许多不同的版本。而本书主要集中对 UNIX 系统发展历史中的一个分支——BSD 系统进行讨论。UNIX 在这个分支上的发展主要由以下几个事件组成：

- 贝尔实验室（Bell Laboratories）创造出 UNIX。
- 加州大学伯克利分校（University of California at Berkeley）的计算机系统研究小组（CSRG）为 UNIX 系统实现虚拟内存机制和 TCP/IP 体系。
- 伯克利软件设计公司（BSDI）在 CSRG 研究的基础上，提出了 FreeBSD 和 NetBSD 的计划并进行开发。

UNIX 系统的起源

第一个 UNIX 系统的诞生是在 1969 年，贝尔实验室的 Ken Thompson 在一台已经过时的 PDP-7 上开始开发这个系统（作为个人研究项目）。很快， Dennis Ritchie，这位 C 语言的创始人加入了进来，用 C 语言重写整个系统，几乎所有的汇编语言程序都被 C 程序所代替。由于 UNIX 系统初始设计得简单、精练，以及随后 15 年内的不断完善和发展 [Ritchie，1984；Comptoa，1985]，该系统成为计算机界最重要和功能最完备的系统之一 [Ritchie，1987]。

Ritchie、Thompson 和一些 UNIX 早期的开发者们曾经参与贝尔实验室的一个名为 Multics 的研究项目，而该项目对新一代操作系统的发展起着不可磨灭的作用。 *UNIX* 从某种角度也可以看作是一个 *Multics*。Multics 所关注的是功能的齐全性，而 UNIX 致力于各项功能的完善。UNIX 中的很多设计方案和系统特征，都直接源于 Multics。例如 UNIX 系统的基本框架，分配一个用户进程作为命令行解释器的思想，文件系统界面的基本结构等。

许多其他操作系统（像美国麻省理工学院的 CTSS） 的思想，也都融入了 UNIX 中。例如创建进程的 *fork* 操作，它源于伯克利分校的 GENIE 系统。用户可以轻易地创建进程，这样每个语句都使用一个进程，而不是被当做一个过程调用，这一想法是源于 Multics。

在 UNIX 发展的历史中至少有三条主线图 1.1 描述了它们早期的形成过程，而图 1.2

则介绍了它们在近些年来的发展情况，特别是对那些后来发展为 4.4 BSD 和 UNIX System V 的分支给予了详细的介绍。在这两张图表中的时间只是一个大概时间，我们也不需要知道一个非常确切的发展过程。图中一些系统的名字我们在下文中并没有提到。它们被收入这里只是为了让读者更清楚其与我们所要具体分析的 BSD 系统之间的关系。

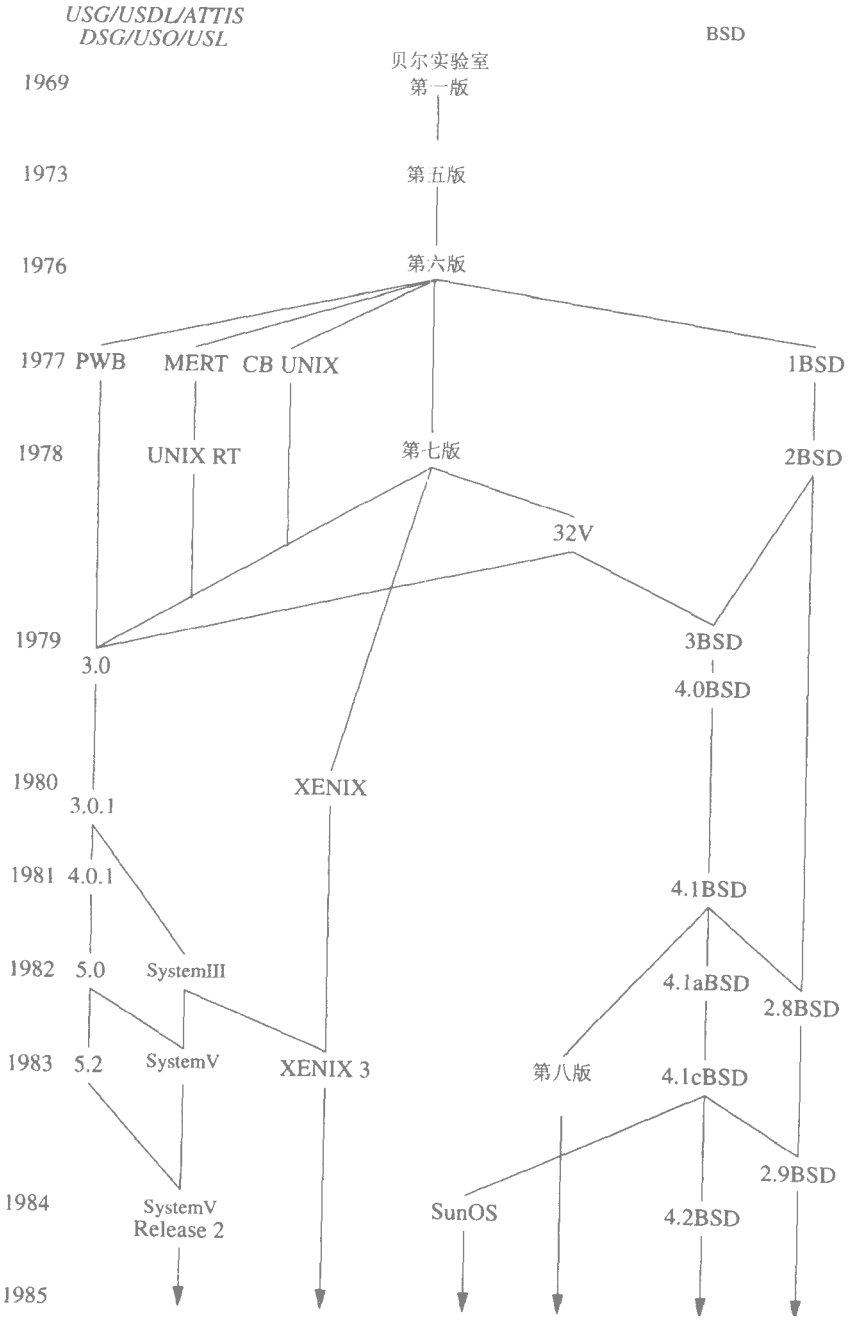


图 1.1　UNIX 系统家族的树型结构（1969～1985）

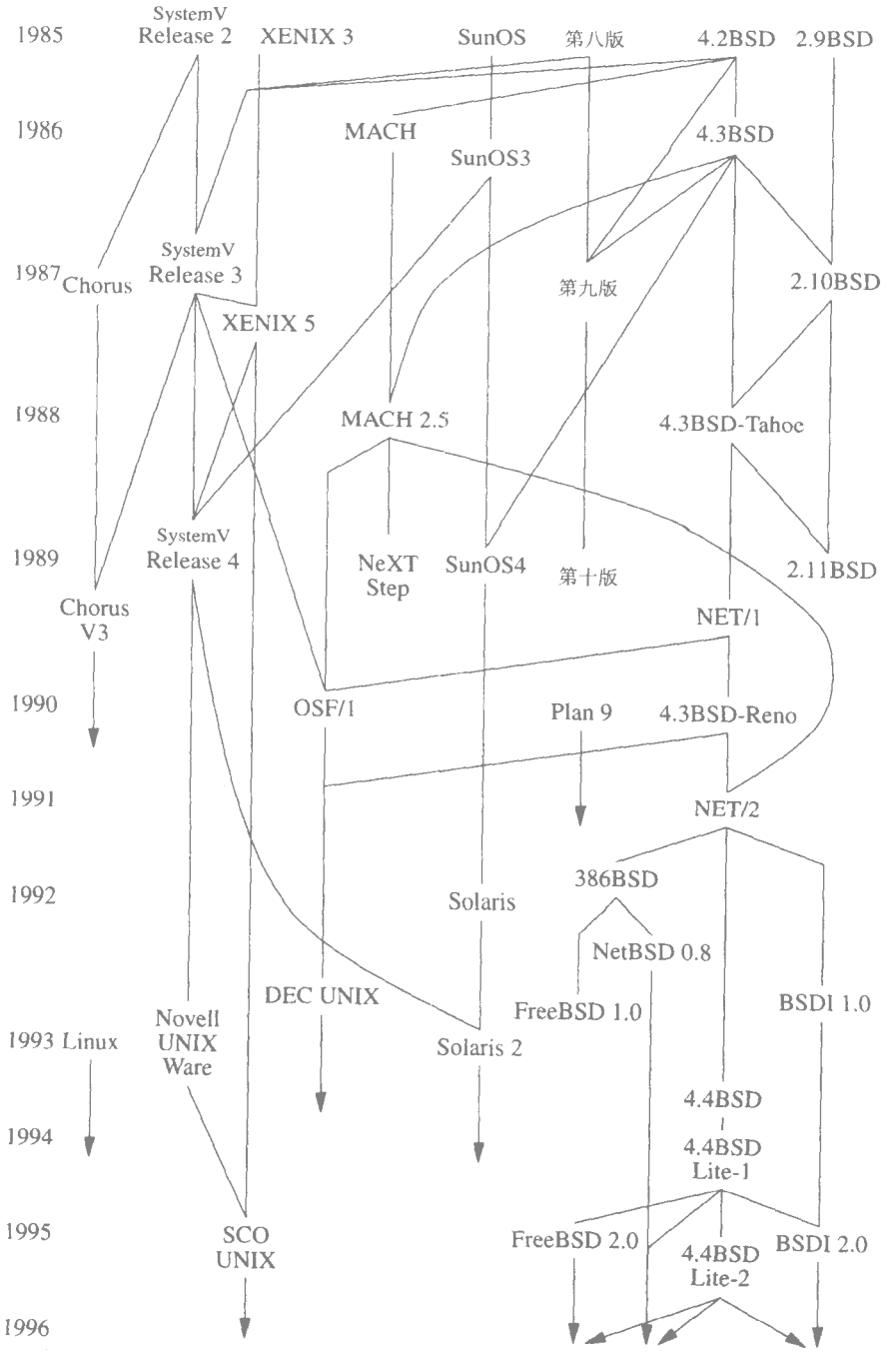


图 1.2　UNIX 系统家族的树型结构（1986～1996）

UNIX 系统的研究与发展

UNIX 第一个重要的成型版本是在贝尔实验室开发成功的。它在该系统的早期版本上新

增了分时系统（UNIX Time-Sharing System）第六版，即我们所熟知的 V6。1976 年，这项技术的第一个版本在贝尔实验室外被广泛使用系统是以《UNIX 程序员手册》（*Programmer's Manual*）发布时的版本号来标识的。

UNIX 和当时的其他操作系统在以下三个方面有重要区别：

（1）UNIX 是用高级语言（C 语言）写成的。

（2）UNIX 是以源代码的形式发布的。

（3）UNIX 系统有很多强大的功能，这些功能在其他的操作系统中只有依托于昂贵的硬件设备才能实现。

UNIX 的源代码绝大多数用 C 语言写成而不是用汇编语言。在当时比较流行的一种思想是操作系统一定要用汇编语言写，这样才能保证它对硬件的有效访问及其执行效率。C 语言本身具有高度的可读性和高效的特点，程序员用它可以很方便地编写那些对硬件进行操作的程序，而不用去摆弄那些复杂而严格的汇编程序。UNIX 中只有 3%的程序在必须用汇编进行硬件访问时使用了汇编语言，例如上下文切换。虽然 UNIX 的成功不仅仅取决于它是用 C 程序写的，但这个选择为它的成功迈出了第一步。Ritchie 的 C 语言是 Thompson 所创造的 B语言的延续和发展，而 B 语言又起源于 BCPL 语言。C 语言自身也在不断地完善和发展中，以至于随后又出现了 C 语言的变种——C++语言。它用数据封装的办法使代码具有高度的可读性。

UNIX 与其他操作系统的另一个重要的区别是代码开放。这起源于贝尔实验室在最初发布 UNIX 给其他研究机构时，就是以源代码的形式发放的。UNIX 的设计者们考虑到这种形式的好处是让其他研究者不仅可以使用这个系统，还可以深入地研究和修改该系统的内部结构。这种对用户的友好关系使得一些新的技术和思想可以很快地融入 UNIX 中来，而这也是这个系统飞速发展的原因之一。一旦当某个系统在某方面超越了 UNIX，总是会有人开始研究它的特性，并利用它的新技术来修改 UNIX 而使之更加完善。这种小巧的，易于理解的设计使得 UNIX 总是能得到最新思想和技术的补充，从而远远超越它最初的设计版本。

第三个重要的区别是 UNIX 提供给一般用户以自由地并发运行多个进程的功能，并且，这些进程与命令行通过管道线（pipeline）技术密切相连。在当时的情况下，一般只有大型的、价格昂贵的机器上运行的操作系统才能提供多进程的并行方式，而且其进程的数量总会严格受到系统性能的限制。

大多数早期的 UNIX 程序都在 PDP-11 上运行，这种机器便宜，并且在当时也算是功能强大。尽管如此，至少也有一个早期的 UNIX 第六版移植到完全不同的体系结构——Interdata 7/32 上运行。PDP-11 的寻址空间太小，而且使用不方便。而新一代的 32 位寻址空间的机器如 VAX-11/780，为 UNIX 创造了拓展功能的空间，使虚拟内存和网络支持的实现成为可能。早期的实验是在不同体系结构的机器上做一些与 UNIX 类似的系统，其结果是令人满意的。它证明了 UNIX 可以很方便地整个移植到其他体系结构的机器上去，就像在其他操作系统中模仿 UNIX 的功能一样容易。第一个可移植的 UNIX 系统是 UNIX 分时系统第七版（V7）。它可以在 PDP-11 和 Interdata 8/32 上运行，并且在 VAX 机器上有一个变种系统 UNIX/32V 分时系统 1.0 版（32V）。贝尔实验室中的研究组又相继研制出系统第八、九、十版（V8、V9、V10）。他们在 1996 年开发出了 Plan 91。

AT&T 的 UNIX SystemIII 和 SystemV

在 1978 年发布系统第七版后，研究组将对外发布权转让给 UNIX 支持小组(UNIX Support Group，USG)。USG 曾经在内部发布过一个名为 UNIX 程序员工作平台（UNIX Programmer's Work Bench，PWB)的版本,而该版本也曾经对外发布过。

在 UNIX 第七版后，USG 的第一个对外发布版本是 1982 年的 UNIX SystemIII。该系统融合了第七版、32V 以及其他研究组以外的机构所开发的 UNIX 系统。UNIX/RT（一个实时的 UNIX系统）与 PWB 原先做的一些工作都被包含进来。USG 随后于 1983 年发行了 UNIX SystemV ,这个系统由 SystemIII 发展而来。而后，贝尔从 AT&T中分离了出来，而 AT&T 得到了发行 SystemV 的许可。

USG 发展成为 UNIX 系统发展实验室（UNIX System Development Laboratory，USDL)。它们在 1984 年发行了 UNIX SystemV，Release 2。UNIX System V，Release 2，Version 4 中引进了分页技术，包括了“写时复制技术”（copy-on-write）和共享内存技术。SystemV 的实现不是基于伯克利分校的页面系统。 USDL 在 AT&T 信息系统（ AT&T Information Systems，ATTIS ）上获得成功，并于 1987 年发布了 UNIX System V，Release 3 。这个版本包括了流（STREAMS）——一个以 V8 中继承而来的 IPC 机制的概念。ATTIS 被 UNIX 系统实验室（UNIX System Laboratories,USL ）开发成功，而这个实验室在 1933 年被出售给 Novell 公司。Novell 公司将 UNIX 的市场份额转让给 X/OPEN 协会，给予他们惟一为 UNIX 产品命名建立标准的权利。两年后，Novell 将 UNIX 出售给 Santa Cruz Operation（SCO).

其他组织

由于 UNIX 易于修改和移植的特性，出现了大量致力于发展 UNIX 的组织。这其中包括 Rand 公司，在第十一章中将提及该公司所开发的 Rand ports 技术；Bolt Beranek and Newman（BBN)，我们将在第十三章中讨论的 4.2 BSD 网络实现的原型系统即是由他们研制成功的；美国伊利诺斯大学（University of Illinois)，他们致力于早期的网络开发工作；还有美国哈佛大学(Harvard)、普渡大学(Purdue)，以及数字设备公司（Digital Equipment Corporation，DEC）等。

如果单纯从运行该系统的机器数量来算的话，由微软公司和 SCO（The Santa Cruz Operation ）合力研制的 XENIX 可以说是使用最为广泛的 UNIX 系统。XENIX最初是基于第七版,但随后又以 SystemV 为基础。最近，SCO 从 Novell 公司收购了 UNIX 并声明要将 UNIX 与 XENIX 合并起来。

也有一些著名的操作系统不是起源于 UNIX 系统，包括 IBM 公司的 OS/2 和微软公司的 Windows 95 和 Windows NT 。这些系统都声称自己是“ UNIX 杀手”，但是至今为止也未成为事 实。

关 于BSD 系统

在既非贝尔实验室也非 AT&T 开发的 UNIX 系统中，最具影响力的应该是加州大学伯克利分校发布的 Berkeley Software Distributions(BSD)例如 4.3BSD 。第一个伯克利的 VAX UNIX 工作是在 32V 版本上增加了虚拟内存，按需调页（demand paging ）和页面替换技术，这项工

作是由 William Joy 和 Ozalp Babaoglu 于 1979 年为开发 3BSD 完成的。当时程序的规模越来越大，如伯克利的 *Franz* LISP，为了解决这个问题，开发了虚拟内存机制。由于这项内存管理工作的成绩，美国国防先进项目研究委员会（Defense Advanced Research Project Agency,DARPA）决定资助伯克利的开发小组完成后面开发标准版本（4BSD）的工作。

4BSD 项目的一个目标是为 DARPA 的 Internet 网络协议 TCP/IP 提供支持。这个网络体系的范围很广，包括各种分散网络设备的互联、局域网（例如以太网 Ethernet 和令牌网 token ring）与远程网络（例如 DARPA 的 ARPANET）之间的连接。

尽管在 3 BSD 后出现了许多 BSD 版本,例如 4.0BSD、4.1BSD、4.2BSD、4.3BSD、4.3BSD Tahoe 和 4.3BSD Reno ,我们还是将所有在 3BSD之后的伯克利 VAX UNIX 系统统称为 4BSD。4BSD 是 VAX 从 1977 年首次出现起，到 1983 年 UNIX 发行 SystemV 为止这段时间内 VAX 专用的操作系统。大多数公司购买一套 32V 的使用许可，但却向伯克利定购 4BSD 使用。很多贝尔实验室内部的机器都开始使用 4.1BSD 系统（当 4.3BSD 系统发布以后，又转为使用 4.3BSD）。一个新的虚拟内存系统随着 4.4BSD 系统一起发布了。VAX 机在当时已经接近淘汰，故而 4.4BSD 不再移植到这种机型上。取而代之的是新一代的 68000、SPARC、MIPS、还有 Intel PC 体系。

4BSD 的开发是在一个由商界和学术界的知名人士组成的专家委员会的指导下进行的。原 DARPA UNIX 系统的顶峰是 1983 年推出的 4.2BSD。经过进一步开发和改进，伯克利分校于 1986 年中期推出 4.3BSD。接下来的版本包括了 1988 年 6 月发布的 4.3BSD Tahoe 和 1990 年 6 月发布的 4.3BSD Reno。这些版本主要是移植到 Computer Consoles Incorporated 的硬件平台上。几乎同时开发的两个自由代码网络版本是： 1989 年 3 月的 4.3BSD Net1 和 1991 年 6 月的 4.3BSD Net2 。这两个版本建立在 4.3BSD 开放代码的基础上，他们的源代码和可执行代码均可自由发布和使用，个人用户与公司都不需要购买 UNIX 的代码许可。CSRG 最后发布的是两个 4.4BSD 的版本，发布时间是 1993 年 6 月。其中一个版本是传统的版本，称为 4.4BSD-Encumbered，包括完整的源代码和可执行代码，用户必须经过 UNIX 代码授权才能使用该版本。另一个版本含有部分 4.4BSD 的源代码，称为 4.4BSD-Lite ，它不含有任何需要 UNIX 授权的代码，所以可以自由地使用和拷贝。在这两个版本发布后，CSRG 宣告解散。4.4BSD-Encumbered 版本按照计划发行了，而 4.4BSD-Lite 版本被 USL 起诉侵权而不能发布。该官司打了一年之久，最后以 CSRG 修改了 4.4BSD-Lite 的部分代码并于 1995 年 6 月发布 4.4BSD-Lite release 2 版本而告终。于是这个版本称为 CSRG 真正的 4.4BSD 最终版。

尽管 CSRG 已经解散,可 4BSD 还是继续同众多的 UNIX 版本和众多其他操作系统并列占有一席之地。

UNIX 世界

世界上许多公司和厂家，包括几乎所有那些占巨大市场份额的商家，都纷纷推出以 UNIX 或其相关、派生系统作为操作平台的计算机。而其他一些厂商则提供大量的相关的外围设备、软件包、技术支持、培训和文档。硬件支持包括微机、袖珍机、多用户机、大型机乃至超级计算机。绝大多数厂家都移植了 SystemV、4.2BSD、4.3BSD、4.4BSD 中的一个，或者兼而有之。我们估计现在大概没有机器还在使用 SystemIII、4.1BSD 或者第七版，当然也不排除

一些地方还在使用 PDP-11，运行 2BSD 或者其他过时操作系统平台的可能性。如果有哪个地方还在使用第六版，那就显得太可笑了（在前言的最后部分可以找到我们的联系方法，如果你还在使用，可以告诉我们）。

UNIX 在学术界也是一个很有前途的领域。Thompson 和 Ritchie 因为开发了 UNIX 系统而得到了图灵奖。UNIX 系统和相关的，特别是专门为教学设计的系统（如 Tunis、XINU 和 MINIX）一起广泛用于操作系统的教学中。Linus Torvalds 在他的自由软件 Linux 中重新实现了 UNIX 的界面。与商业和工程领域相比，UNIX 系统更加广泛地应用在世界各地的教学和科研机构中。

虽然 CSRG 已经解散了，但 4.4BSD 系统仍然生机勃勃。在自由软件领域，FreeBSD 和 NetBSD 小组继续从事基于 4.4BSD 系统的研究和开发工作。其中 FreeBSD 致力于将 4.4BSD 系统应用于个人 PC 的硬件平台上。而 NetBSD 使 4.4BSD 尽可能地适应更多的硬件体系结构，可以在各种平台上运行。这两个小组都是从 Net2 版本做起，但在 4.4BSD-Lite 发布后，又都开始基于这个版本进行开发。

在商业领域中，与 4.4BSD 最接近的变体是由伯克利软件设计公司（Berkeley Software Design.Inc.，BSDI）开发的 BSD/OS。早期的 BSDI 软件是基于 Net2 版本，而后来转为基于 4.4BSD-Lite。

1.2 BSD 和其他系统

CSRG 在他的系统中不仅融入了 UNIX 系统的技术，同时也包括了其他操作系统中先进的思想。许多 4BSD 中与终端驱动程序有关的结构来源于 TENEX/TOPS-20。从 TOPS-20 继承下来的任务控制体系（概念上，不是实现上）源于 MIT Incompatible Timesharing System（ITS）。而最早在 4.2BSD 上提出的虚拟内存机制（由 CSRG 和其他几个商家共同实现）建立的基础——文件映射和页面级界面也是最早出现于 TENEX/TOPS-20。现有的 4.4BSD 虚拟内存系统（详见第五章）是借鉴于 MACH——一个 4.3BSD 的分支系统。还有许多新的模块在设计时都从 Multics 的项目中得到启发。

在 CSRG 的设计中，高效总是被当做一个最主要的因素而优先考虑。而效率的改进方法总是在与 VAX 上原有的操作系统——VMS 进行比较中而得到的。

其他一些与 UNIX 相关的系统也借鉴了 4BSD 的先进技术。例如 AT&T UNIX SystemV、IEEE POSIX.1 标准，以及相关的国家标准局（National Bureau of Standards，NBS）联邦信息处理标准（Federal Information Processing Standard，FIPS），它们都从 4BSD 中汲取了一些思想，例如：

- 任务控制（详见第二章）。
- 可靠信号（详见第四章）。
- 多文件访问许可组（详见第六章）。
- 文件系统界面（详见第七章）。

X/OPEN 小组（原先只是由欧洲的一些商家组成，而现在却包括了大部分美国的 UNIX 商家）发布了《*X/OPEN Portability Guide*》，并于最近又发布了《*Spec 1170 Guide*》。这些文

档制定了内核的标准界面，并提供给用户大量的 UNIX 标准应用程序。当 1993 年 Novell 从 AT&T 手中收购 UNIX 时，以 UNIX 的惟一命名权将 UNIX 的名字改为 X/OPEN。这样一来，任何企图标榜自己为 UNIX 的系统必须参照 X/OPEN 制定的界面规则。X/OPEN 手册从 POSIX 中借鉴了不少成分。POSIX.1 标准是一个 ISO 国际标准，编号为 SC22 WG15。因此，世界上众多的类 UNIX 系统（UNIX-like systems）都接受了 POSIX 标准。

4BSD 中的 *socket* 进程间通信机制（详见第十一章）具有高度可移植性，虽然从没有在该系统上发布过，但它很快就被移植到了 AT&T SystemIII 上。4BSD 中关于 TCP/IP 的网络协议体系（详见第十三章）被广泛用作各种平台上实现 TCP/IP 体系的基础：从 AT&T 3B 的机器上运行的 SystemV，到 VMS，再到 IBM PC。

CSRG 与那些基于 4.2BSD 和 4.3BSD 系统上开发自己操作系统的商家保持着紧密的合作关系。这种共同发展的关系使 4.3BSD 的移植变得更加畅通无阻，并极大地促进了目前正在进行的系统开发工作。

用户交流对系统发展的影响

伯克利版本所做的大量发展工作都建立在用户反馈的基础上。除了给予资助的 DARPA 外，分布在世界各地的广大 BSD 用户也纷纷对研究组提出大量的意见和建议。

除此之外，伯克利研究组还从现有的其他软件中得到启发。分布在澳大利亚、加拿大、欧洲和美国的高校和组织都对 4BSD 的发展起到了极大的促进作用。这其中包括一些重要的功能，如自动配置与硬盘配额体制。虽然代码版权阻止了对 SystemIII 和 SystemV 的照搬照抄，但还是有一些创意（如 fcntl 系统调用）是来源于 SystemV 的。此外，CSRG 还发布了大量的用户提供的外围软件程序。

一个通过交流发展而成的工具实例就是在 4.3BSD Tahoe 中采用的公共时区处理软件包（public-domain time-zone-handling package）。一个国际研究小组（包括 Arthur Olson，Robert Elz 和 Guy Harris）参与了该软件的设计和实现，部分原因是在 USENET 新闻组 comp.std.unix 里的讨论。此软件包可以完全不用 C 语言库文件来实现时区转换规则，将它们存入文件中，并不需要改动任何系统代码就可以实现时区规则的转换。因此这个软件在只有可执行代码而没有源代码的 UNIX 系统中变得尤为有效。它还使用户可以自行制定时区规则，而避免被系统的时区规则所限制。这个软件包包含一个巨大的数据库，在其中存放着世界各地的时区规则，从中国到澳大利亚到欧洲。只要有了它，4.4BSD 系统就不用再对某一个地区设置特定的时间，系统的实现也就简化了。这种时区软件包在 BSD 中的应用吸引了一些商家的注意力，如 Sun 公司，他们将这种技术融入了自己的系统中。

伯克利研究组征求了一些漏洞及其修改办法的电子邮件。UNIX 软件之家 MT XINU 基于这些意见发布了一个漏洞列表。许多漏洞修正程序被融入了随后的版本中。在 USENET 的 comp.unix 新闻组中，经常会有一些关于 UNIX（包括 4.4BSD）的讨论，这些讨论都在 Internet 上发布。Internet 和 USENET 都是国际性的。除此以外，USENET 上还有一个专门为 4BSD 修正漏洞的讨论组 comp.bugs.4bsd。只有少数的意见会被伯克利开发机构所采纳，因为这些讨论组上的意见太多了，以至于绝大多数无法得到处理。于是在此后又成立了一个专门发布 CSRG 对漏洞处理意见的新闻组 comp.bugs.4bsd.bugfixes。以上这些新闻组中的讨论常常会使一些新的功能得到实现，最终，它们将融入 4BSD 系统中。

1.3 4BSD 的设计目标

4BSD 是一个主要由研究机构发布，供科研使用的系统，并且最近开始介入商业的应用中。人们在系统的开发过程中提出了许多非传统性的思想，并把它们加入系统之中。而这些思想或是应用总会产生实际的商业效应。

早期的 4BSD 系统是技术驱动的。这些技术使设计者们能够充分地利用那些其他 UNIX 系统还无法利用起来的硬件设备。这其中包括：

- 虚拟内存的支持。
- 第三方（非 DEC）设备的驱动程序。
- 超平台的终端屏幕支持应用程序库。大量的应用程序由这个程序库中派生出来，其中包括著名的全屏幕编辑器 vi。

AT&T 32V 版本只给予少数硬件支持，而 4BSD 却提供了大量的第三方设备驱动程序。这可以说是该系统能如此流行的重要因素之一。对于一些不得不节约硬件开支的大学，在一些厂家开始提供他们自己的类 4.2BSD 系统以前，4.2BSD 几乎是他们惟一的选择。

超平台的终端屏幕支持虽然看上去比较呆板，但在当时也为 4BSD 系统得以流行起到了非常重要的作用。

4.2BSD 设计目标

DARPA 希望伯克利将 4.2BSD 发展为一个在 VAX 机上的标准科研系统。为此，开发者们设计了很多新的技术，其中包括全新的支持进程在巨大而稀疏的地址空间中寻址的虚拟内存系统、高速的文件系统、进程间通信技术，以及网络支持。新的虚拟内存系统和高速文件系统是应一些研究者在计算机辅助设计与制造（CAD/CAM）、图像处理和人工智能（AI）方面的需求而产生的。而进程间通信技术则是来源于对分布式系统的研究工作的需求。网络的研究主要起源于 DARPA 希望他们的人员可以使用 56kbps 的 ARPA Internet 进行网络互联(并且伯克利研究组还对高速局域网的研究很感兴趣）。

开发者并没有尝试把 4.2BSD 做成一个完全分布式系统，而是采用了传统的 ARPANET 资源共享体系。这主要有三个原因：

1. 系统大量分散于各处，并且需要高度自治。而在那时的分布式数据库的技术要求有一个中央管理授权机构。
2. 当时著名的紧密连接系统算法还不成熟。
3. 伯克利的原则是尽可能使用现有的已经成熟的技术，而不是去发明一些新技术。

这样，系统提供了许多方便的网络工具如远程登录（*rlogin*、*telnet*）、文件传输(*rcp*、*ftp*)，以及远程指令执行（*rsh*），但是所有的机器都仍然保持着自己显式的身份验证体系，而并没有像分布式系统那样被封装起来。

由于发布时间的限制，4.2BSD 并没有把设计时所考虑到的所有新技术都写在系统中。新的虚拟内存系统也不在 4.2BSD 发布版中。CSRG 把他们的注意力集中在一些高速发展的硬件技术领域。其网络系统支持众多的硬件设备，包括对 10Mbps 以太网、令牌网和 NSC Hyperchannel 的多重界面支持。内核的源代码被模块化并重新搭建，以适合在各种新型的体

系结构（从微处理器到大型机）中进行移植。

4.3BSD 设计目标

4.2BSD 的种种不足是继续开发 4.3BSD 的原因之一。由于 4.2BSD 中新增了许多功能，导致其代码量加大，从而在某些方面的性能上甚至不如 4.1BSD。在系统中还发现了不少漏洞，特别是在 TCP 协议的实现中。另一些内容由于缺乏足够的时间而没有包含进来。还有一些内容，例如 TCP/IP 的子网和路由支持，由于没有迅速得到外界的认可，从而没有来得及加入到 4.2BSD 系统中。

商业软件一般都有各版本向下兼容的特性，这样使现有的应用程序不至于因过时而作废。然而，维持这种兼容性是十分困难的事情，所以大多数科研系统中几乎都对它弃之不顾。BSD 版本采用一个折衷的解决方法，即尽可能地做到向下兼容，并在不能兼容的部件上清楚地标明出来。这种方法的好处是可以让软件自由地升级，而不用考虑兼容性的种种条条框框。实际上，从 4.2BSD 到 4.3BSD 的兼容性也非常令人满意，大量 4.2BSD 的应用程序可以顺利地移植到 4.3BSD 系统中。

与 4.2BSD 相比，4.3BSD 的 C 语言界面只有少量的几个关于终端界面的语句和一个 IPC 系统调用的参数（*select*；详见 6.4 节）被改变。4.3BSD 中增加了一个标志参数，它创建一个信号句柄，并允许进程直接获取 4.1BSD 的信号字符，而不是 4.2BSD 的（详见 4.7 节）。增加这个标志的惟一目的就是让那些建立在旧的系统字符表上的应用程序得以延续运行，而不至于过时作废。

用户看不到 4.3BSD 在实现上与 4.2BSD 的不同之处，然而确实有很多地方做了改动。例如开发者对多套网络协议体系进行了改进，除了 TCP/IP 体系之外，又加入了 XEROX NS。

4.3BSD 的第三个发布版（下文中称作 4.3BSD Tahoe）除了支持 VAX 之外，还增加了对 Computer Consoles 公司（CCI）Power 6（Tahoe）系列微型机的支持。虽然它与在 VAX 上的版本大部分都相似，但仍有一些不同之处。

4.3BSD 的第三个发布版（下文中称作 4.3BSD-Reno）加入了 ISO/OSI 网络支持、一个全新的 NFS 版本，以及对 POSIX.1 的转换和补充。

4.4BSD 设计目标

4.4BSD 拓展了 4.3BSD 的硬件基础，现在已经支持众多的体系结构，包括摩托罗拉 68K、Sun SPARC、MIPS 和 Intel PC。

4.4BSD 系统弥补了 4.3BSD 中的很多不足之处。特别是虚拟内存机制已被完全替换。新的虚拟内存机制不再以 VAX 机为基础，它为现行的大内存体系结构提供了有效的算法。4.4BSD 系统还增加了 ISO 标准网络协议的实现，并大人提高了 TCP/IP 协议体系的性能。

终端驱动程序不仅兼容第七版，而且也兼容第六版。这部分内容曾经一度十分有用，但现在却日益变得无足轻重。在 4.4BSD 中，CSRG 将其置换为一个与 POSIX 兼容的终端驱动程序——由于 SystemV 支持 POSIX，故而这个程序支持 SystemV。兼容 POSIX 变成一个目标。对 POSIX 的支持并不只限于内核部件（如 termios 和会话（session）），还包括了对绝大多数 POSIX 工具的支持。

4.3BSD 最大的缺陷就是不支持多种文件系统。在网络协议中，不可能存在着某种能够适

应各种情况的单一的文件系统。故而一般要求操作系统支持多种不同的文件系统，就像要支持多种不同的网络协议同时运行一样。4.4BSD 设计了一种类似于 Sun 公司 vnode 体系的面向对象文件系统界面。这种框架支持与 4.3BSD 中网络协议数相当的多个本地或者远程的文件系统。使用 vnode 界面建立了一个可以动态添加和修改的文件系统管理机制。在这种机制下，4.4BSD 支持众多的文件系统类型，包括回滚、联合、 uid/gid 映射层结构，再加上一个在 CD-ROM 上特别适用的 ISO9660 文件系统。除此以外，系统还支持 Sun 的网络文件系统（NFS）第二版和第三版，还有一个新的基于本地磁盘的日志文件系统。

对于可动态调配的 IPC 进程模块的研究始于贝尔实验室的 UNIX 第八版。基于流（stream）的 I/O 系统建立在 UNIX 字符 I/O 系统的基础上。它允许一个用户进程开启一个新的未知端口，并在其上运行相关的内核处理模块，就像在做普通的终端行编辑一样。用来处理网络协议的进程模块也可以如此处理。在一个网络处理模块的基础上运行终端处理模块，这样就在内核上实现了高效动态的网络虚拟终端（*network virtual terminal*）。然而 stream 模块也有一个缺陷，因为它们本身是线性的，所以不能充分处理那些基于网络的由各种设备提供的多路数据的输入和输出。UNIX 第八版的流 I/O 系统作为 STREAMS 系统被 UNIX SystemV 的第三个版本所采用。

在 4.2BSD 中对网络部件的设计使用了一种不同的方法，它基于 *socket* 界面和一套动态多层网络体系。这就允许在同一个系统中同时运行若干套基于流、数据报或者其他各种不同访问方式的网络协议。协议模块用一套机制处理各种连接和设备中得到的多路不同数据。4.4BSD 还对 socket 界面做了少量的补充，使之允许 ISO 网络协议的实现。

1.4 系统的发布

CSRG 一直都是一个小型的软件开发者小组。这种资源方面的缺乏，致使软件工程的管理显得犹为重要。CSRG 的工作人员和全世界致力于该系统的组织都注意在系统开发过程中进行仔细认真的协调。虽然现在 CSRG 已经不复存在，但这些组织仍然活跃在 BSD 领域中。他们继承了 BSD 的传统，继续从事着 FreeBSD、NetBSD 和 BSDI 的研究。

CSRG 发布的版本主要分两种：

- 以新内容为主的系统：3BSD、4.0BSD、4.2BSD、4.4BSD。
- 以修正漏洞和提高性能为主的系统： 4.1BSD、4.3BSD。

这种结构保证了系统得以及时发布、修改、完善和提高，及时发布以修改为主的版本充分反应了 CSRG 致力于开发一个高性能可靠系统的原则。

表 1.1 4.2BSD 的发布步骤

描述	发布步骤			
	alpha	internal	beta	final
名称	4.1aBSD	4.1bBSD	4.1cBSD	4.2BSD
所做主要更新	网络	快速文件系统	IPC	新的信号机制

CSRG 的版本发行分三个步骤：alpha 版、beta 版和 final 版，如表 1.1 所示。alpha 版和

beta 版一般来说并不是真正的发布版——它们只是用作测试的版本。alpha 版一般只发布在为数不多的几个地方，其中大部分都是在大学中。 beta 版的发布点相对多一些，但是人们不能直接得到这个版本，而是通过一个树形结构传播开去，并且通过该结构反馈漏洞报告，修改办法和测试评估收集到 CSRG 手中。例如，4.1aBSD 在 100 多个站点上运行，但是 beta 版却只发布给 15 个主要站点。beta 版测试的树形结构使 CSRG 的开发者们可以把精力集中在软件的开发上，而不是整天从浩如烟海的 beta 版测试报告中找资料。本书在检验技术准确与否的工作上也是采取类似的办法。

许多参与 beta 版测试工作的主要成员不仅拥有软件的拷贝，还拥有可以登录到伯克利研究组主机的账号。这样的用户经常通过 Internet 或者通过电话拨号登录到伯克利主机上，他们大多数分布在世界各地，如澳大利亚、英国，美国麻省尤他州、马里兰州、德克萨斯州、伊利诺斯州，或者一些相对近一点的地方，例如斯坦福。这样的用户被授权可以直接更改主机上 4.3BSD 和 4.4BSD 的源代码拷贝。许多系统组件，例如 C 语言和 Fortran 语言的编译器，还有一些重要的网络应用程序，例如 *telnet* 和 *ftp*，其中都包含着这些非 CSRG 人士所做出的重要贡献。有一个例外，就是所有内核相关的修改都是由 CSRG 人员完成的，虽然这些修改建议经常是由外部人员提出。

虽然这些被授权直接修改代码的人总是在动手之前经过认真的分析设计，但他们还不能做到完全没有失误。于是，在整个 beta 版测试的末期，CSRG 人员通过与以前的版本（例如 4.2BSD 和 4.3BSD）相比较来检验他们所写程序的正确性。一些被认为不适用的代码，例如对列目录命令增加的参数和对 fseek() 系统调用返回值的修改，最终都从发布版中删除了。

这个过程显示了一个工程只有少数几个主要开发者的好处：这些开发者都会从头到尾地熟悉整个系统的设计、运作过程，他们知道自己的代码怎样更好地与整个系统结合起来，从而形成一个和谐的最终版本。那些拥有大量开发人员的大公司在这一点上就显得欠缺。

CSRG 所做的工作完全不含有任何商业目的。这样就促使系统的所有重要设计都从技术上出发，而不用考虑任何市场效益问题。BSD 的开发者们一直站在这个立场上，故而从不承诺版本发行的具体日期。

参考文献

AT&T, 1987.
AT&T, *The System V Interface Definition (SVID)*, Issue 2, American Telephone and Telegraph, Murray Hill, NJ, January 1987.

Babaoğlu & Joy, 1981.
O. Babaoğlu & W. N. Joy, "Converting a Swap-Based System to Do Paging in an Architecture Lacking Page-Referenced Bits," *Proceedings of the Eighth Symposium on Operating Systems Principles*, pp. 78–86, December 1981.

Bach, 1986.
M. J. Bach, *The Design of the UNIX Operating System*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

Cerf & Cain, 1983.

V. Cerf & E. Cain, *The DoD Internet Architecture Model,* pp. 307–318, Elsevier Science, Amsterdam, Netherlands, 1983.

Chambers & Quarterman, 1983.

J. B. Chambers & J. S. Quarterman, "UNIX System V and 4.1C BSD," *USENIX Association Conference Proceedings*, pp. 267–291, June 1983.

Comer, 1984.

D. Comer, *Operating System Design: The Xinu Approach,* Prentice-Hall, Englewood Cliffs, NJ, 1984.

Compton, 1985.

M. Compton, editor, "The Evolution of UNIX," *UNIX Review*, vol. 3, no. 1, January 1985.

Ewens et al, 1985.

P. Ewens, D. R. Blythe, M. Funkenhauser, & R. C. Holt, "Tunis: A Distributed Multiprocessor Operating System," *USENIX Association Conference Proceedings*, pp. 247–254, June 1985.

Holt, 1983.

R. C. Holt, *Concurrent Euclid, the UNIX System, and Tunis,* Addison-Wesley, Reading, MA, 1983.

Joy, 1980.

W. N. Joy, "Comments on the Performance of UNIX on the VAX," Technical Report, University of California Computer System Research Group, Berkeley, CA, April 1980.

Jung, 1985.

R. S. Jung, "Porting the AT&T Demand Paged UNIX Implementation to Microcomputers," *USENIX Association Conference Proceedings*, pp. 361–370, June 1985.

Kashtan, 1980.

D. L. Kashtan, "UNIX and VMS: Some Performance Comparisons," Technical Report, SRI International, Menlo Park, CA, February 1980.

Kernighan & Ritchie, 1978.

B. W. Kernighan & D. M. Ritchie, *The C Programming Language,* Prentice-Hall, Englewood Cliffs, NJ, 1978.

Kernighan & Ritchie, 1988.

B. W. Kernighan & D. M. Ritchie, *The C Programming Language,* 2nd ed, Prentice-Hall, Englewood Cliffs, NJ, 1988.

McKusick, 1985.

M. K. McKusick, "A Berkeley Odyssey," *UNIX Review*, vol. 3, no. 1, p. 30, January 1985.

Miller, 1978.

R. Miller, "UNIX—A Portable Operating System," *ACM Operating System Review*, vol. 12, no. 3, pp. 32–37, July 1978.

Miller, 1984.

R. Miller, "A Demand Paging Virtual Memory Manager for System V," *USENIX Association Conference Proceedings*, p. 178–182, June 1984.

Mohr, 1985.

A. Mohr, "The Genesis Story," *UNIX Review*, vol. 3, no. 1, p. 18, January 1985.

Organick, 1975.

E. I. Organick, *The Multics System: An Examination of Its Structure,* MIT

Press, Cambridge, MA, 1975.

P1003.1, 1988.

P1003.1, *IEEE P1003.1 Portable Operating System Interface for Computer Environments (POSIX),* Institute of Electrical and Electronic Engineers, Piscataway, NJ, 1988.

Peirce, 1985.

N. Peirce, "Putting UNIX In Perspective: An Interview with Victor Vyssotsky," *UNIX Review*, vol. 3, no. 1, p. 58, January 1985.

Popek, 1981.

B. Popek, "Locus: A Network Transparent, High Reliability Distributed System," *Proceedings of the Eighth Symposium on Operating Systems Principles*, p. 169–177, December 1981.

Presotto & Ritchie, 1985.

D. L. Presotto & D. M. Ritchie, "Interprocess Communication in the Eighth Edition UNIX System," *USENIX Association Conference Proceedings*, p. 309–316, June 1985.

Richards & Whitby-Strevens, 1980.

M. Richards & C. Whitby-Strevens, *BCPL: The Language and Its Compiler,* Cambridge University Press, Cambridge, U.K., 1980, 1982.

Ritchie, 1978.

D. M. Ritchie, "A Retrospective," *Bell System Technical Journal*, vol. 57, no. 6, p. 1947–1969, July–August 1978.

Ritchie, 1984a.

D. M. Ritchie, "The Evolution of the UNIX Time-Sharing System," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 8, p. 1577–1593, October 1984.

Ritchie, 1984b.

D. M. Ritchie, "Reflections on Software Research," *Comm ACM*, vol. 27, no. 8, p. 758–760, 1984.

Ritchie, 1987.

D. M. Ritchie, "Unix: A Dialectic," *USENIX Association Conference Proceedings*, p. 29–34, January 1987.

Ritchie et al, 1978.

D. M. Ritchie, S. C. Johnson, M. E. Lesk, & B. W. Kernighan, "The C Programming Language," *Bell System Technical Journal*, vol. 57, no. 6, p. 1991–2019, July–August 1978.

Rosler, 1984.

L. Rosler, "The Evolution of C—Past and Future," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 8, pp. 1685–1699, October 1984.

Sandberg et al, 1985.

R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, & B. Lyon, "Design and Implementation of the Sun Network Filesystem," *USENIX Association Conference Proceedings*, pp. 119–130, June 1985.

Stroustrup, 1984.

B. Stroustrup, "Data Abstraction in C," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 8, pp. 1701–1732, October 1984.

Tanenbaum, 1987.

A. S. Tanenbaum, *Operating Systems: Design and Implementation,* Prentice-Hall, Englewood Cliffs, NJ, 1987.

Tuthill, 1985.
B. Tuthill, "The Evolution of C: Heresy and Prophecy," *UNIX Review*, vol. 3, no. 1, p. 80, January 1985.

Uniejewski, 1985.
J. Uniejewski, *UNIX System V and BSD4.2 Compatibility Study,* Apollo Computer, Chelmsford, MA, March 1985.

USENIX, 1987.
USENIX, *Proceedings of the C++ Workshop,* USENIX Association, Berkeley, CA, November 1987.

Wilson, 1985.
O. Wilson, "The Business Evolution of the UNIX System," *UNIX Review*, vol. 3, no. 1, p. 46, January 1985.

X3J11, 1988.
X3J11, *X3.159 Programming Language C Standard,* Global Press, Santa Ana, CA, 1988.

X/OPEN, 1987.
X/OPEN, *The X/OPEN Portability Guide (XPG),* Issue 2, Elsevier Science, Amsterdam, Netherlands, 1987.