

第 1 章 简 介

1.1 简 介

1994 年计算机界迎来了 UNIX 操作系统的第 25 个生日。自 UNIX 系统于 1969 年诞生以来，它已被移植到数十种硬件平台上，商业公司、大学、研究机构都以各种各样的形式发行 UNIX。它最初仅是由若干程序组成的一个小系统，现已成长为应用领域广泛、功能丰富的操作系统。今天，各种版本的 UNIX 已应用于从嵌入式处理器到工作站、桌面系统，乃至服务于大群用户的高性能多处理机系统。

UNIX 系统包括运行 UNIX 操作系统之上的用户程序集，库，以及其他实用工具，而 UNIX 操作系统本身则为这些应用提供运行环境和各种各样的系统服务。本书讲解操作系统自身的设计与实现，不涉及运行于其上的应用和工具。自 UNIX 降生于贝尔电话实验室(BTL)以来，除 BTL 主要完成了它的早期发行之外，UNIX 一直受到许多公司和大学的青睐。之后，UNIX 的变体在市场上繁衍起来。所有这些变体都不很严格地支持一套基本的正常“UNIX 系统”所拥有的接口、应用及其特征。这些变体在各自的内部实现，接口的具体语义，以及它们提供的“增值”特征方面千差万别。本书将主要集中介绍几个主线发行，如 Novell 公司的 System V Release 4(SVR4)，加州大学伯克利分校的伯克利软件发行(4.x BSD)，以及卡内基·梅隆大学的 Mach。此外还涉及一些商业实现版本，如 Sun Microsystems 公司的 SunOS 和 Solaris，数据设备公司(DEC)的 Digital UNIX，以及惠普公司的 HP-UX。

本章简要介绍 UNIX 操作系统。首先是它的简史，包括 UNIX 的诞生，成长，最终为业界接受。然后是影响 UNIX 系统发展的诸多因素。最后探讨一下 UNIX 的未来发展之路。

1.1.1 简史

在深入地探讨 UNIX 系统设计之前，很有必要回顾一下它的历史和发展历程。在以下的小节中，我们将追溯 UNIX 系统的成长之路。从其在 AT&T 的朴素开端，到它现在略显混乱，号称为多平台，多厂商，多型的操作系统。此外，还有不少资料提供了更详尽的史料，比如 Peter Salus 的 A Quarter Century of UNIX[Salu 94]。本章将着重介绍其中的重要事件。

1.1.2 创始之初

60 年代末，贝尔电话实验室与通用电气、麻省理工学院合作开发了一个称为 Multics 的操作系统[Orga 72]。1969 年 3 月，Multics 项目撤销了。来自 BTL 的主要开发人员开始寻找其他有趣的项目。这其中，Ken Thompson 写了一个名叫“太空漫游”的游戏程序。他找到了一台基本无人使用的 PDP-7 计算机(数据设备公司产品)运行他的程序。PDP-7 缺少程序开发环境。Thompson 不得不在另一台运行(GECOS 操作系统的 Honeywell 635 上交叉汇编他的程序，然后再通过纸带加载到 PDP-7 上。

为了方便太空漫游的开发，Thompson 和 Dennis Ritchie 一起着手开发 PDP-7 上的操作环境。第一部分是一个简单的文件系统，它后来演化成称之为 System V 文件系统(s5fs)的早期版本。他们很快又加入了一个进程子系统，一个简单的称为 shell 的命令解释器(后来发展为 Bourn Shell[Bour 78])和一些实用工具。现在这个系统是自足的了，不再需要 GECOS 环境了。他们将它命名为 UNIX，是 Multics 的双关语。

接下来的一年，Thompson，Ritchie 和 Joseph Ossanna 向 BTL 中请购买 Digital 的 PDP-11，作为 BTL 专利部门的字处理系统。他们将 UNIX 移植到 PDP-11 上，增加了几个包括 ed 编辑器、runoff 文字照排工具等在内的字处理工具。Thompson 还开发了一种称为 B

的新语言(早期语言 BCPL 的一个分支「Rich 82」),并且用它编写了早期的汇编器和一些实用工具。B 语言是一种解释型语言,性能不佳。后来,Ritchie 又继续开发出 C 语言。这是一种编译型语言并且支持数据类型和数据结构。C 语言甚至比 UNIX 系统还要成功。

UNIX 在 BTL 内部开始流行起来,许多人都投入到它的开发中来。1971 年 11 月,Ritchie 和 Thompson 由 Doug McIlroy 牵头出版了第 1 版《UNIX 程序员手册》。这之后 BTL 又相继出版 10 个版本的 UNIX 和相应的 10 个版本手册。

起初的几个版本仅限于 BTL 内部发行。1973 年 2 月,第 3 版中加入了 cc,C 语言编译器。同年,UNIX 用 C 语言进行了彻底的重写(这就是 1973 年 11 月的版本 4)。这是 UNIX 迈向成功之路的关键一步。Thompson 和 Ritchie 共同发表了首篇 UNIX 论文,The UNIX Time Sharing System[Thom 74]。它刊登在 1973 年 10 月的 ACM Symposium on Operating Systems(SOSP)和 1974 年 7 月的 Communication of the ACM 这是 UNIX 首次对外界亮相。

1.1.3 繁衍

1956 年美国司法部依据反托拉斯法案向 AT&T 和西电公司 提起诉讼。AT&T 与联邦政府签署了“consent decree”。该协议不允许 AT&T 生产与电话或电报服务无关的设备,也不允许从事除“公用通信服务”外的任何商业活动。

结果;AT&T 不得不面对不能经营计算机产品的尴尬境地。但自从在 SOSP 上发表论文以来,越来越多的人希望能够获得 UNIX 软件和源代码。在签署了简单的许可证协议后,AT&T 无偿地向大学提供 UNIX 系统用于教学和研究。AT&T 既不为 UNIX 做广告,也不出售 UNIX,更不对它的发行版本提供支持。在那些最早的领取许可证的用户之中就有加州大学伯克利分校,它在 1973 年 12 月获得了 UNIX 系统。

在这种情势下,UNIX 在世界范围内快速发展。到 1975 年底,它已经传播到西至耶路撒冷的 Hebrew 大学,南至澳大利亚的新南威尔士大学,北至加拿大多伦多大学的世界各地。UNIX 的第一个移植是在 InTerdata 机上完成的,由 Wollongong 大学在 1976 年独立完成。1977 年在 BTL 的 Ritchie 和 Steve Johnson 也完成了同样的移植工作。

1979 年 1 月发行的 UNIX 版本是一个真正可移植的 UNIX 系统,它对其后的 UNIX 发展有着深远的影响。该版本最初是运行在 PDP-11 和 Interdata 8/32 上的,该系统更加健壮,而且提供了比版本 6 更强大的功能,但其相当慢。许多领取了许可证的用户针对这一情况从不同方面对系统的性能进行了改进;AT&T 又将这些改进中的许多加入了其以后的 UNIX 发行中。这种管理员与用户之间的合作精神上是 UNIX 快速成长和不断发展的关键因素(不幸的是,当 UNIX 在商业上取得很大成功之后,这种精神就荡然无存了)。

很快 UNIX 被移植到其他几种体系结构中。微软公司与 Santa Cruz Operation(SCO)合作将 UNIX 移植到 Intel 8086 上,也就是 XENIX 系统,它是最早的 UNIX 商业变体之一。1978 年 Digital 推出了 32 位的 VAX 11。在遭到 Ritchie,Thompson 和 Johnson 等人的拒绝后.Digital 找到 BTL 在新泽西州 Holmdel 的一个分部,由它的一个小组将 UNIX 移植到 VAX 上。这是首次在 32 位机上移植 UNIX,该版本称为 UNIX/32V。该版本后来寄给了加州大学伯克利分校,1979 年它又发展为 3BSD。

1.1.4 BSD

1974 年 12 月,加州大学伯克利分校获得了 UNIX 许可证。在其后的几年里,包括 Bill Joy 和 Chuck Haley 在内的一组研究生为 UNIX 开发了几个实用工具,其中包括 ex 编辑器(后来发展为 vi)和一个 Pascal 编译器。他们将这些软件绑定成一个称为“伯克利软件销售(BSD)”的软件包,并在 1978 年春以每个许可证 50 美元出售。最初的 BSD 发行(1978 年末发行了版本 2)仅包括应用程序和实用工具,并没有对操作系统进行修改和再发行。Joy 早期的贡献之一就是 C hell[Joy 86],它具有 Bourne shell 不支持的若干功能,如作业控制、命令历史等。

1978 年伯克利得到一台 VAX-11/780 以及由新泽西州 Holmdel 的 BTL 小组移植的 UNIX/32V。VAX-11 采用 32 位体系结构，可以有 4GB 的地址空间，而物理内存仅为 2MB。与此同时，Ozalp Babaoglu 为 VAX 设计了一个分页式的虚存系统，并将其加入到 UNIX 系统中。这就是 1979 年末发行的 3BSD，是由伯克利发行的第一个操作系统。

伯克利在虚存方面的工作促使美国国防部高级研究项目署(DARPA)决定投资支持伯克利的 UNIX 系统开发。DARPA 项目的目的之一就是将传输控制协议/网际协议(TCP/IP)网络协议族集成到 UNIX 系统中。在 DARPA 的资助下，伯克利又推出了几个 BSD 发行，统称为 4BSD：1980 年的 4.0BSD，1981 年的 4.1BSD，1983 年的 4.2BSD，1986 年的 4.3BSD，以及 1993 年的 4.4BSD。

伯克利小组在技术方面为 UNIX 系统做出了许多贡献。除了虚存和 TCP/IP 的集成外，BSD UNIX 还引入了快速文件系统(FFS)，可靠的信号机制，以及 socket 等先进技术。4.4BSD 用崭新的基于 Mach(见 1.1.7 节)的虚存设计代替了原有的设计，并加入了其他改进，如日志结构文件系统等。

伯克利对 UNIX 的开发工作是由计算机科学研究小组(CSRG)完成的。在 4.4BSD 时，CSRG 决定停工，终止 UNIX 的开发。下面是从[Bost 93]上抄录的原因：

- 缺少资金和支持。
- 许多商业系统都有 BSD 系统的特征。
- 系统已太庞大，太复杂，难于由一个小组构建和维护。

这之后，成立了一家称为伯克利软件设计公司(BSDI)的商业公司，该公司完成 4.4BSD 的商业化，并出售其产品。由于他们已经用伯克利开发的新的源代码替代了原来的 UNIX 源代码，BSDI 公司声称其 BSD/386 中的源代码不受 AT&T 的许可证制约。UNIX 系统实验室和 AT&T 的一家负责 UNIX 开发的子公司，上诉 BSDI 公司和加州伯克利董事会，状告其侵犯版权、违约和盗用商业机密[Gerb 92]。而这次诉讼的导火线则是 BSDI 公司使用电话 1-800-ITS-UNIX 出售 UNIX 源代码。加州大学反诉，结果双方的争执耽搁了源代码的发行。1994 年 2 月 4 日，双方都撤回了诉讼，庭外解决。BSDI 宣布推出 4.4BSD-Lite，加上不受制约的源代码，它的售价为 \$ 1000。

1.1.5 System V

现在再回到 AT&T。其与美国司法部的法律大战终于在 1982 年达到终点，一部里程碑式的法令出台了。根据该法令：西电公司被迫解散，地区性运营公司都从 AT&T 中分离出来，成立了一系列的“贝尔宝宝”，贝尔电话实验室也独立出来，改名为 AT&T 贝尔实验室。相应地，AT&T 也被允诺可以进入计算机市场。

在 BTL 研究小组继续进行 UNIX 工作的同时，对外部发行 UNIX 的责任落到了 UNIX 支持小组身上，然后又是 UNIX 系统开发小组。接着是 AT&T 信息系统。这些小组先后于 1982 年发行了 System III，1983 年 System V，1984 年 System V Release 2(SVR2)，1987 年 Release 3(SVR3)。AT&T 不断地开拓 UNIX 的市场，许多商业 UNIX 实现都是基于它的系统实现的。

System V UNIX 也引入了许多新特征，新设施。它的被称为“区”结构的虚存实现与 BSD 截然不同。SVR3 增加了进程间通信设施(包括共享内存，信号量和消息队列)，远程文件共享，共享库，以及用于设备驱动程序和网络协议的 STREAMS 框架。最新的 System V 版本是 Release 4(SVR4)。我们将在 1.1.10 节中介绍。

1.1.6 商业化

UNIX 的不断发展吸引了许多计算机公司。它们急急忙忙地开始将 UNIX 商业化，出售自己的 UNIX 版本。每一家都是以 AT&T 或伯克利的 UNIX 发行的版本为蓝本，将它们移植到自己的硬件上，并加上一些自己的“增值”功能。1977 年，Interactive System 公司率先成为 UNIX 厂商。它们的第一个发行称为 IS/1，并运行在 PDP-11 上。

1982 年, Bill Joy 离开了伯克利, 与人合资成立了 Sun Microsystems 公司, 发行基于 4.2BSD 的 SunOS(后来又基于 SVR4 发行 Solaris)。微软与 SCO 合作发行 XENIX。此后, SCO 将 SVR3 移植到 386 上, 并将其命名为 SCO UNIX。80 年代出现了许多商业 UNIX, 包括 IBM 的 AIX, 惠普公司的 HP-UX, Digital 公司的 ULTRIX(后来是 DEC OSF/1, 之后又改名为 Digital UNIX)。

各种商业变体都增加了许多新特征, 这些功能中的许多后来又被加入到主线系统中。SunOS 开发了网络文件系统(NFS), 支持多种文件系统的 vnode/vfs 接口, 以及后来由 SVR4 采纳的新型虚存体系结构。AIX 是第一个提供日志文件系统的 UNIX 系统。ULTRIX 是首批支持多处理机系统的 UNIX 系统之一。

1.1.7 Mach

UNIX 系统广泛流行的一个重要原因就是它的内核小巧、简单, 同时又提供丰富而且实用的功能。随着系统不断地添加功能, 内核变得越来越庞大、复杂和笨拙。许多人开始感觉到 UNIX 正在偏离最初使其趋于完美和成功的那些准则。

80 年代中期, 在宾夕法尼亚匹兹堡的卡内基·梅隆大学, 有一群研究者开始了一个称为 Mach 的全新的操作系统的研发工作[Acce 86]。他们的目标是开发一个微内核, 内核仅提供必要功能的最小集, 同时提供一个在用户级实现其他操作系统功能的框架。Mach 系统结构将支持 UNIX 编程接口, 可以在单处理机或多处理机系统上运行, 适用于分布式环境。一切都是从零开始, 他们希望能够避免那些困扰 UNIX 的问题。

基本的方案是让微内核对外开放几个简单的抽象概念, 然后用一组称为服务器的用户级任务提供大部分操作系统功能。Mach 还有另外一个优势- 它不受 AT&T 许可证限制, 这吸引了不少厂商。Mach 2.5 是最流行的发行, OSF/1、Nextstep 等商业系统都是基于它开发的。早期版本的 Mach 都是单体式内核, 并在上层提供 4BSD UNIX 的接口。Mach 3.0 是第一个实现微内核的系统。

1.1.8 标准

UNIX 变体的泛滥造成了若干不兼容问题。尽管所有的变体从某种程度“看起来都像 UNIX”, 但它们在许多重要的方面却千差万别。最初, 业界根据 AT&T 的 System V 发行正式版 UNIX)和伯克利的 BSD 发行的不同而一分为二。随着商业变体的不断涌现, 情况愈来愈糟。System V 与 4BSD 有许多不同- 它们的物理文件系统, 网络结构和虚存体系不尽相同, 而且不兼容。某些不同仅限下内核的设计与实现, 而某些则影响到用户编程。此时, 你已不可能编写一个复杂的应用, 而不加修改就可以在 System V 和 BSD 系统上运行。

每个商业变体都以 System V 或 BSD 为基础加上自己的“增值”特征, 而这些额外的特征通常是不可移植的。结果, 应用程序员被搞得晕头转向, 花费大量的精力来保证它们的程序可以运行于各种不同的 UNIX。

在这种局面下, 人们强烈要求出台一组标准接口, 许多小组开始了这方面的工作。可结果却是标准也如 UNIX 变体一样浩如烟海, 变幻莫测。最终, 大部分厂商在少数标准上达成协议。这些标准包括 AT&T 的 System V 接口定义(SVID), IEEE POSIX 规范, 以及 X/Open 财团的 X/Open 可移植导引。

每部标准都仅仅关注于程序员与操作系统间的接口, 并不规定系统该如何实现这些接口。它定义这些函数及它们的语义细节。依从标准的系统必须满足这些规范, 但可以在不同的层次实现这些函数、比如在内核或用户级库中。

标准只涉及大多数 UNIX 系统所具有的功能的一个子集。理论上说; 如果程序员只使用该子集的函数, 他们的应用程序就可以移植到任何依队标准的系统上。但如果程序员不牺牲可移植性的话, 他就不可能利用硬件或操作系统的特性对应用程序进行优化, 也不可能利用某些厂商提供的特殊功能。

SVID 是 System V 编程接口的详细规范。AT&T 业已发行了三个版本——SVID, SVID2, SVID3[AT&T 89], 分别对应 SVR2, SVR3, SVR4。只有厂商的系统依从了 SVID 规范。它的系统才可以称为是“System V”的。AT&T 同时提供了 System 验证套件(SVVS)用于检验系统是否依从 SVID 规范。

1986 年, IEEE 委派它的一个委员会制定一套操作系统环境的标准。他们使用的名称是 POSIX(基于 UNIX 的可移植操作系统), 该标准基本上是 SVR3 和 4.3BSD UNIX 主要部分的混合。1990 年, IEEE 公布了 POSIX.1 1003.1 标准, 一般称为 POSIX.1[IEEE 90]。由于它并不偏爱于任何一个 UNIX 变体, 该标准已受到广泛的承认。

X/Open 是一个 1984 年成立的国际性计算机厂商组成的财团。它并不制定新标准, 而是基于已有的事实标准开发一个开放式的通用应用环境(CAE)。它已出版了 7 卷册的 X/Open 可移植性导引(XPG), 其最新版本是 1993 年的版本 4[XPG4 93]。XPG 是依照 POSIX.1 标准草案完成, 同时增加了许多 POSIX 不涉及的领域, 如应用软件的国际化, 窗口界面, 以及数据管理等。

1.1.9 OSF 和 UI

1987 年, 面对大众对其许可证制度的不满, AT&T 宣布购买 Sun Microsystems 公司 20% 的股份。AT&T 和 Sun 计划合作开发 AT&T 的 System V UNIX 的下一个发行即 SVR4。AT&T 向 Sun 保证它可以获得特惠待遇, 而 Sun 则一改它的基于 4BSD 的 Sun OS, 宣布其下一代操作系统将基于 SVR4。

UNIX 其他厂商对此反应强烈, 他们害怕这会给 Sun 公司带来不公平的竞争优势。于是包括 Digital、IBM、HP、Apollo 在内的厂商联手于 1988 年成立了开放软件基金会(OSF)。OSF 由其组建公司资助, 主要从事操作系统, 用户环境和分布式计算环境的开发。所有这些都不再受 AT&T 的许可证制约。它接收来自其成员的技术申请(RFT), 然后经过一套中立于厂商的过程从中选出最佳方案。

作为报复 AT&T 与 Sun 连同其他基于 System V 的厂商随即成立了一个称为 UNIX 国际(UI)的组织。UI 主要出售 SVR4, 并规划 UNIX System V 的未来发展方向。1990 年, UI 发行 UNIX System V 的前景规划, 勾画了未来 UNIX 的发展方向。

1989 年, OSF 出台了一套称为 Motif 的图形用户界面, 深受欢迎。之后不久, 它又推出其操作系统的最初版本, OSF/1。OSF/1 的一个发行是基于 Mach 2.5 的, 同时兼容 4.3BSD, 还附带了 IBM AIX 操作系统的一些特征。它还带有许多 SVR4 没有的先进功能, 如完整的多处理机支持, 动态加载, 以及逻辑卷管理。按照 OSF 的计划, 其组建成员将基于 OSF/1 开发它们的商业操作系统。

OSF 和 UI 这对竞争对手很快就不得不面对共同的来自外界的威胁了。90 年代初期经济低迷. 再加上微软的 Windows 系统蓬勃发展, 这一切都威胁着 UNIX 的发展, 乃至生存。1993 年 UI 停止商业运作。同时 OSF 也放弃了许多雄心勃勃的计划(如分布式管理环境)。由 Digital 在 1993 年发行的 DEC OSF/1 是唯一一个基于 OSF/1 的主要商业操作系统。此后 Digital 从其操作系统中删减了许多与 OSF/1 相关的部分。1995 年则干脆将其改名为 Digital UNIX。

1.1.10 SVR4 及其之后

AT&T 和 Sun 联合开发 System V Release 4(SVR4), 并于 1989 年首次发行。SVR4 集成了 SVR3, 4BSD, SunOS, 以及 XENIX 的一些特征, 还增加了一些新功能, 如实时调度, Korn shell, 以及对 STREAMS 子系统的改进。次年, AT&T 成立了一家称为 UNIX 实验室(USL)的软件公司, 开发和出售 UNIX。

1991 年, 生产 Netware PC 机网络操作系统的 Novell 公司收购了部分 USL, 并联合投资创建了 Univel。Univel 主要开发 SVR4 的桌面版, 集成 Netware。这个称为 [UnixWare 的系统于 1992 年末发行。这之后, 又出现了新的 SVR4 发行, 其中最新的是 SVR 4.2/ES/MP,

它支持多处理机并增强了系统安全性。

1993 年, AT&T 将 USL 的剩余股份一并卖给了 Novell。该年末, Novell 向 X/Open 发行了 UNIX 商标和授权书。1994 年 Sun Microsystem 向 Novell 购买了 SVR4 代码使用权, 从此不必再受授权书的限制。Sun 公司基于 SVR4 的发行称为 Solaris, 最新的版本是 Solaris 2.5。Solaris 提供了许多先进功能, 包括全抢占多线程内核, 对多处理机的全面支持等等。

1.2 演变的动力

在过去的 25 年里, UNIX 已经发生了相当大的变化。最初, 它仅是个由实验室小组使用的简单操作环境。如今, 它已成长为一个主流操作系统, 许多厂商都以各种版本出售 UNIX。它广泛地应用于各种系统, 从小巧的嵌入式控制器到巨型机和大规模并行系统。它还广泛地应用于各种应用领域, 包括办公室的桌面系统, 金融业的大型数据库管理, 或量子物理实验室的高速数字计算。

为了适应新的挑战, UNIX 系统已发生了相当大的变化。虽然现在它已成为一个 J 戈熟的操作系统, 但仍免不了会进行进一步的修改、这种不断的改进并不意味着最初的设计不够精湛。相反, 能够将新技术源源不断地融入 UNIX 系统正是归功于它的原始设计。UNIX 的原创者们并没有顽固地对操作系统的目的、形式和功能抱有当时的成见。最初, 他们只给出了一个简单的可扩充的框架。然后由业界, 学术界, 以及热心的用户逐渐为其添砖加瓦。

我们需要回过头来看看是什么因素促成了一个操作系统的成长和变化, 这是件非常有益的事。本节我们将回顾影响 UNIX 系统成长的主要因素, 并预测 UNIX 未来发展的方向。

1.2.1 功能

为系统增加新功能这种变化的最大的源动力。起初, 系统主要是通过用户级实用工具增添新功能。随着系统不断成熟, 开发者们慢慢开始为 UNIX 内核本身增加功能。

许多新功能主要用来支持复杂的应用程序。最显著的一个例子就是 System V 的进程间通信 (IPC) 软件包, 它包括共享内存, 信号量, 以及消息队列。这些设施可以使协作进程共享数据, 交换消息, 同步彼此的动作。许多现代 UNIX 系统还在不同层次上提供多线程支持。

IPC 和线程都用于辅助复杂应用程序的开发上。如基于 Client/Server 模型的应用。在这些程序中, 服务器大体上是一个循环结构, 时刻等待客户请求。当一个请求到达时, 服务器对其处理并继续等待下一个请求。由于服务器可能要同时为若干个客户提供服务, 它应该能一并发地处理多个请求。服务器可以为每个请求派生一个进程, 利用 IPC 机制共享数据和完成彼此间的同步、多线程系统则允许将服务器实现为多个共享同一地址空间的线程组成的单个进程。

整个操作系统变化最显著的部分可能就是文件系统了。UNIX 文件系统也加入不少新特征。这些特征包括先入先出 (FIFO) 文件, 符号链接, 以及跨越多个分区的文件等等。现代 UNIX 系统还支持文件锁以及对文件某一区域加锁的功能。此外还有访问控制表和用户配额功能等等。

1.2.2 网络

内核中变化最大的部分要算网络子系统了。早期的 UNIX 系统都是单机运行的, 彼此间不能通信。随着计算机网络的普及, UNIX 支持网络迫在眉睫。承担网络工作的主要部门是伯克利, DARPA 资助它将 TCP/IP 协议族集成到 4BSD 系统中。今天 UNIX 系统支持许多网络接口 (如以太网, FDDI 和 ATM)、协议 (如 TCP/IP, UDP/IP 和 SNA) 和组织框架 (如 socket 和 STREAMS)。

可互连的机器间在许多方面给系统造成了重大影响。用户需要在互连的机器上共享文件, 运行远程节点上的程序等等。为了满足这些需求, UNIX 从 3 个方面进行解决:

- 开发了各种分布式文件系统, 可以透明地访问远程节点上的文件。它们中最成功的是

Sun Microsystem 公司的网络文件系统(NFS)，卡内基·梅隆大学的 Andrew 文件系统(AFS) 和 Transar 公司的分布式文件系统(DFS)。

- 同时还开发许多分布式服务，可以在网络上共享信息。一般它们都实现为基于 client/server 模型的用户程序，使用 RPC 调用其他节点上的操作。这样的系统有 Sun Microsystems 公司的网络信息服务(NIS)和开放软件基金会的分布式计算环境(DCE)。

- 此外，还有分布式操作系统，如 Mac ,Chorus ,Sprite 等等。它们都不同程度地与 UNIX 兼容，是开拓未来分布式 UNIX 市场的基础技术。

1.2.3 性能

提高系统性能是造成这种变化的一个恒久不变的源动力。相互竞争的 UNIX 厂商全力展示自己的系统性能比其他竞争对手要好。因为内核子系统的性能问题，它们几乎都进行过较大的改动。

20 世纪 80 年代初，伯克利推出快速文件系统，利用智能化磁盘块分配策略提高性能。紧随快速文件系统，又出现了区间化分配技术和日志技术等。在许多开发领域都要求对性能进行改进，诸如进程间通信，内存体系结构，以及多线程进程等。一个处理机已不能再满足许多应用的需要了，于是许多厂商随即开发了多处理机的 UNIX 系统，有的系统甚至支持数百个 CPU。

1.2.4 硬件变化

UNIX 系统必须紧随计算机硬件技术的发展。这通常意味要将操作系统移植到更新更快的处理机上。由于 UNIX 大部分是用 C 语言写的，移植工作相对简单。最近几年，开发者已花了很大的精力将硬件相关代码分离出来，形成一个独立的模块。在移植到一个新机器时只需改写这些模块。这些模块通常都用于中断处理，虚地址转换，上下文切换，以及设备驱动程序等。

在某些情况下，操作系统需要进行大的修改才能运行到新硬件上。最明显的一个例子就是多处理机系统、传统的 UNIX 内核是为单处理机系统设计的，缺少对多个处理机并发访问共享数据的保护。现在已有不少厂商开发了多处理机 UNIX 系统，许多都是使用原来的 UNIX 内核，然后使用锁来保护共享数据结构。这一过程称为并行化。也有些系统自底向上重新构建内核，然后再利用新原语将已有的子系统集成到系统中。

再进一步分析可以看出，不同硬件技术的发展速度是不平衡的，这也对操作系统的设计有着较大的影响。自首次在 PDP-7 上实现 UNIX 系统起，运行 UNIX 机器的 CPU 速度已增长了约 100 倍，每个用户的内存容量和磁盘容量都提高了 20 多倍。而另一方面，内存和磁盘速度增长了还不到 1 倍。

20 世纪 70 年代，UNIX 的性能主要受处理速度和内存大小的影响。这主要是因为内核普遍采用了交换和换页技术，在较小的内存中运行许多进程。随着时间推移，内存和 CPU 的速度都不再是什么问题了。系统性能主要受 I/O 影响。内核耗费大量的时间在主存和磁盘间移动页面。为了减弱磁盘的瓶颈效应，人们开始对文件系统，存储设备，以及虚存体系结构进行了大量的研究。结果，人们发明了廉价磁盘冗余阵列(RAID)，日志结构文件系统也日益流行。

1.2.5 改进质量

如果系统总是出错，即便有了好的功能和速度也无济于事。为了使系统更加健壮并且使现有的软件运转得更好更可靠，人们对系统进行了大量的修改。

最初的信号机制不可靠而且在许多方面都是不完善的。首先是伯克利然后是 AT & T 修订了该实现，提供了可靠的信号设施，称为可靠信号。

System V 和 BSD 的文件系统在系统崩溃时都是十分脆弱的。UNIX 系统将数据缓存在内存中而不是将修改同步地写到磁盘上。结果，在系统崩溃时可能丢失某些数据而使文件系统

进入一种不一致状态。传统情况下，UNIX 提供一个称为 fsck(8)的使用工具，用它来检查和修复已毁坏的文件系统。这是一个非常耗时的操作，在有許多磁盘的大型服务器上需要数十分钟。许多现代 UNIX 系统都使用一种带有日志技术的文件系统，不再使用 fsck，大大提高厂系统的可用性和健壮性。

1.2.6 模式变化

最近的 30 年用户使用计算机的方式发生了极大的变化。20 世纪 70 年代的典型系统是大型的、集中式的计算机。这些计算机要占整个房子大小，许多用户通过由串口线相连的终端来使用系统。系统一般为分时系统 - 计算机将 CPU 时间片均匀地分给所有用户、用户的终端一般为“哑”终端，它们只是一个简单的文本显示系统。

20 世纪 80 年代出现了工作站、工作站一般带有高速位图显示器，可以将其分割为若干个窗口，每一个都运行一个 UNIX shell。这是交互式操作所需的理想设施。同时工作站也拥有足够的计算能力，可以运行一般的用户应用程序。尽管工作站可以支持多用户，但一般同一时间只由一个用户使用。工作站用高速网络相连，可以互相通信，也可以和其他计算机通信。

很快一种新的计算模型出现了，这就是客户-服务器计算。在这种计算模型中，一个或多个功能强、集中式的机器成为服务器，它向称为客户机的工作站提供各种服务。许多系统都使用文件服务器作为用户文件的一个公用存储设备，用户可以通过许多种协议访问它们。计算服务器是有一个或多个功能强大的处理机的机器，用户可以向它提交含有复杂计算的批处理作业(如科学计算应用)。数据库服务器运行数据库引擎，处理客户提交的查询和事务请求。服务器都是带有高速处理器和大量内存和磁盘的高端机器，而客户机工作站相对计算功能较弱，内存和磁盘较少，但有很好的显示和交互特性。

随着工作站变得越来越强大，客户机与服务器间的差别越来越模糊。此外，将重要的服务集中于少数几台服务器会造成网络拥塞和服务器过载。结果又带来了一次模式变化，这回是分布式计算。在这种模型中，许多机器协同提供基于网络的服务。例如，每个节点可以有一个本地文件系统，而且同时可以向其他节点提供文件服务。因此，每个节点既是文件服务器，允许其他节点使用自己的本地文件系统，又是其他节点的客户。从而避免了网络拥塞和单点失效。

UNIX 已采纳了多种不同的计算模型。例如，早期的 UNIX 发行只有一个本地文件系统。在支持了网络协议之后，又开发了分布式文件系统。这些分布式文件系统的某些，比如 AFS 的早期版本，需要集中式的、专用的服务器。迟早这些系统都会演化为分布式文件系统，每一台机器既是客户又是服务器。

1.2.7 其他应用领域

UNIX 系统最初是设计用于研究实验室和大学这样的简单分时环境，允许有多个用户运行简单的诸如文字处理、编辑和数字计算之类的程序。随着 UNIX 的成功，人们开始将它应用到更为广泛的应用领域。20 世纪 90 年代初期，UNIX 已应用于量子物理和空间实验室，运行音频和视频应用程序的多媒体工作站，以及用于任务关键系统的嵌入式控制器。

每种应用领域都对 UNIX 系统提出了不同类型的需求，而 UNIX 则不得不将加入相应的修改满足这些要求。多媒体和嵌入式应用需要保证资源的可用性和有限的响应时间。科学应用则需要同时使用许多个处理机。现代 UNIX 系统为这些需求增加了若干实时特性，诸如固定优先级进程，处理机的集群调度，以及在内存中锁定数据的功能。

1.2.8 简洁就是美

最早，UNIX 系统的一个最大的优点是小巧和简单，并且有一组基本抽象概念。基本的模式就是系统提供简单的工具，并且用户可以灵活地将它们以管道组合起来。传统 UNIX 内核是

单体的，不易扩充。随着越来越多的功能加入，它变得越来越大，越来越复杂，从最初的不到 10 万字节增长到几兆字节。由于计算机的内存不断增长，厂商和用户起初都忽略了这一问题。但这在低端系统却是一个不小的问题，UNIX 不适合于小的个人计算机系统和膝上型计算机系统。

许多人开始感觉到并不是所有的修改都会带来好处，而且系统已经变得庞大、混乱、毫无组织。为此，许多人投入大量精力重写系统，或者说是按照最初的 UNIX 思想写一个新的，更易扩充，更模块化的操作系统。这些系统中最成功的系统就是 Mach，许多像 OSF/1 和 NextStep 之类的商业操作系统都是基于它实现的。Mach 最终发展为微内核体系(见 1.1.7 节)，有一个很小的内核为运行应用程序提供框架结构，并通过用户级任务提供其他操作系统的功能。

控制内核大小的努力只能算是基本上成功了。由于消息传递的代价，微内核无法提供与传统单体内核相提并论的性能。而其他一些并不是那么雄心勃勃的努力反而更成功，如模块化(见 1.2.9 节)，可换页内核，以及可以按照需要将某个部件加载到内核或从内核中卸载的动态加载机制。

1.2.9 灵活性

20 世纪 70 年代和 80 年代初的 UNIX 内核功能并不十分丰富。它们只支持一种类型的文件系统，一种调度策略和一种可执行文件格式(图 1-1)。唯一有一定灵活性的地方就是块设备和字符设备的函数转换表，通过它可以用一个统一的接口访问各种不同类型的设备。80 年代中期开发的分布式文件系统要求 UNIX 支持本地和远程两种文件系统。类似地还有不同类型的可执行程序格式所需的共享库。UNIX 不得不加入一些新的可执行文件格式，同时为了兼容性还必须支持传统的 a.out 格式。在支持一般的交互式程序并存的同时，UNIX 还要支持多媒体应用和实时应用，这要求调度器支持不同类型的应用程序。

图 1-1 传统的 UNIX 内核

总之，UNIX 系统的广泛使用需要人们构建一个更加灵活的操作系统，可以用不同的方法完成同一个任务。这需要开发各种灵活的框架结构，如 vnode/vfs 接口，exec 开关表，调度类，以及基于段的内存体系结构。现代 UNIX 内核与图 1-2 所示的系统非常相似，每一个外围的圆代表一个可以用不同方法实现的接口。

图 1-2 现代 UNIX 内核

1.3 回顾与展望

在 UNIX 系统出现之后，它已经有了长足的发展。尽管它最初很简陋，现在 UNIX 系统仍是非常流行。在 20 世纪 80 年代中期到 80 年代末，UNIX 已成为商业、大学和研究机构等各种不同环境的首选操作系统。一般地，只有比较正式的购买者才决定到底买那种类型的 UNIX。后来，这种显赫的地位受到微软的 Windows 和 Windows/NT 操作系统的挑战。在低端、桌面市场上，UNIX 显然已经是败下阵来。在本节中，我们来看看 UNIX 系统如此成功，如此流行的原因，同时也来看看是什么因素导致它失去了计算机世界的霸主地位。

1.3.1 UNIX 好在哪里

UNIX 如此的成功，这大大超出了它的创造者最初的设想。UNIX 成功的一个主要原因是它最初的发行方式。由于法令(consent decree)的限制，AT & T 不得不以极少的费用出售 UNIX 许可证和源代码。世界各地的人们都可以索取和运行该系统。由于附带源代码，用户可以分析它，更改它，告诉其他人自己所做的修改。然后 AT & T 再将开多修改加入到新发行中。在伯克利的工作更带动了这一趋势。总而言之，UNIX 在一种极为开放的过程(或者说没有过程)中不断发展。各国、各大洲的学术界、厂商、以及疯狂的黑客们都为 UNIX 的发展做出了贡献。即便在 UNIX 商业化之后，许多厂商也认识到开放系统的价值，将它们的成果公布于众，制定

了许多像 NFS 这样的开放规范。

最初的 UNIX 设计非常精湛，并为后来版本和分支奠定了成功的基础。它的一个最大的长处就是始终贯彻“越小越完美”的思想[Allm87]，用一个小的内核提供一组必要功能的最小集。系统有许多小的实用工具完成简单的数据处理。用户可以通过管道机制以及可编程的 shell 等工具将这些实用工具以各种方式组合起来，形成功能更强大的工具。

UNIX 的文件系统是这种小巧，简单方式的一个典型代表。与当时使用诸如索引顺序访问方式 (ISAM)，层次顺序访问方式 (HSAM) 等复杂文件访问方式的操作系统不同，UNIX 将文件看成是一个字节流。应用它可以在文件上建立任何数据结构，实用自己的访问方式访问文件，不需要文件系统的干预。

大多数系统应用都使用文本数据。例如，比较重要的系统数据库 /etc/passwd，/etc/fstab，以及 /etc/ttys 都是普通的文本文件。尽管使用二进制的有结构的方法组织信息会更有效，使用一般的文本可以让用户不必使用特殊的工具就可以访问和处理这些文件。文本是一件友好的，通用的，高度可移植的数据形式，可以用许多实用工具进行处理。

UNIX 另一个很突出的特点就是它简单而统一的 I/O 设备接口。UNIX 将所有的设备都用文件表示，用户可以使用与处理文件相同的命令和系统调用集访问设备。开发者写程序完成 I/O 操作时不必考虑这个操作是对文件的，还是对用户终端，打印机，或其他设备进行的。这一特性与 shell 中的 I/O 重定向一起提供了一个简单而强大的 I/O 接口。

促使 UNIX 成功和流行的一个关键因素是它的可移植性。内核的大部分都是用 C 语言写的。这就使其在移植到新机器时相对容易了许多。它先是用于 PDP-11，后来移植到 VAX-11 及其他许多机器、许多厂商都是开发出自己的新机器，然后简单将 UNIX 移植到它的上面，而不必重写一个新的操作系统。

1.3.2 UNIX 的误区在哪儿

事物总有相反的两个方面。在赞誉 UNIX 优点的同时，也不能忽略了它的缺点。对 UNIX 进行如此客观回顾的人中就有这样一个人，他不是别人，正是 Dennis Ritchie 本人。在 1987 年 1 月 USENIX 会议中的一次 UNIX 回顾会晤中，Ritchie 分析了许多 UNIX 的缺点[Rit87]，下面我们对其进行简述。

尽管 UNIX 最初是一个简单的系统。它已不再是这样了。例如，AT&T 为了高效地对数据进行缓冲和将程序移植到非 UNIX 系统上，为 UNIX 系统增加了标准 I/O 库。这个库相对其底层的系统调用接口已变得越来越复杂了。例如，read 和 write 系统调用是原子性的(不可分的)操作，而 I/O 库中的缓冲破坏了这种原子性。

虽然 UNIX 是一个优秀的操作系统，但大多数用户并不需要一个操作系统，他们只要有一个可以完成特定任务的系统就行了。这些用户并不关心底层文件系统结构和进程结构是否优美。他们只需要以最少的代价运行一个特定的应用(如编辑器，商务软件包，绘图程序)、早期的 UNIX 缺少一个简单统一的(最好是图形的)用户界面，这也是阻碍 UNIX 被大众接受的一个主要因素。用 Ritchie 的话来说就是，“UNIX 简单而一致，但只有天才(至少是程序员)才能赏识它的简洁。”

以积木方式构造上工具利弊参半。尽管它优美而且从审美角度来说非常优美，但用户需要具有丰富的创造力和想像力才能高效地使用这些工具。许多用户更喜欢使用个人计算机中有的那些已集成好的，一步到位的程序。

从某件程度上说，UNIX 也是自己成功发展的受害者。它简单的许可证条款以及可移植性造成它不可控制的发展和扩散。人们都对 UNIX 系统进行修补，每个小组都以不同的方式进行修改，结果是他们的系统往往不兼容。起初只有两大势力——AT&T 和 BSD。它们都有自己的文件系统，内存结构，以及自己的信号和终端处理框架。很快许多厂商都发行了自己的变体，在一定程度上保持与 AT&T 版本和 BSD 版本的兼容性。后来，这种局面变得更加混乱了，

许多应用开发者不得个花很大的力气才能将它们的程序移植到各种不同的 UNIX 上。

由于参加标准化的人们意见不一，标准化并没有完全成功。这主要是每个厂商都想加入一些特有的特性来标榜自己的“产品差异”，表明自己的产品与众不同，或者优于它们的竞争对手。

Mach 系统的主要开发人员 Recharh Rashid 更进一步地探讨了 UNIX 的失败的原因。在 Mach 系列讲座的开场白中[Rash 89]，他解释了他们是如何通过对 UNIX 发展的调查产生了开发 Mach 系统的灵感。UNIX 有一种最小化的，积木式的工具构造方法。通过将小巧简单的工具组合在一起构造大型的复杂的工具。但这一方法并没有用于内核的构造。

传统的 UNIX 内核不够灵活，不具备很好的可扩充性，也很少代码复用的设施。随着 UNIX 的发展，开发者只简单把代码加入内核，结果内核成了新功能的“倾销地”。很快内核就膨胀起来。缺少模块化，而且越来越复杂。Mach 试图用一种新方法来解决它，它使用一组精简的抽象概念从底向上重新编写操作系统。现代 UNIX 采用不同的方式解决这一问题，为若干子系统增加灵活的框架结构，见 1.2.9 节所述。

1.4 本书的范围

本书讲述现代 UNIX 系统。但为了保证完整性和给用户一个历史线索，本书也简要介绍一些以往 UNIX 版本的特性。现在市场上有许多 UNIX 系统，每一种都与众不同。我们将这些系统分为两类——主线系统和商业变体。主线系统包括 System V，4BSD 和 Mach。变体都是从这些系统之一演比而来的，同时又加入了一些厂商的增值特性和改进。这些系统包括 Sun Microsystems 的 Sun OS 和 Solaris 2.x，IBM 的 AIX，惠普的 HP-UX，以及 Digital 的 UTRIX 和 Digital UNIX。

本书并不专门讲述 UNIX 的某个特定的版本或变体。相反，它考察许多 UNIX 的重要实现，并比较它们的结构和处理某些重要问题的方法。我们将重点讲述 SVR4，同时还包括 4.3BSD，4.4BSD 以及 Mach 系统。在商业变体中，本书将用很大的篇幅讲解 SunOS 和 Solaris 2.x。这不仅是因为其在 UNIX 市场上的重要性，而且还因为 Sun Microsystems 开创了许多项先进技术，这些技术后来又融入了主线系统中。此外，它们还发表了许多关于其系统的论文。

一般情况下，书中都要引述传统 UNIX 或现代 UNIX。这里传统 UNIX 是指 SVR3，4.3BSD 以及更早的版本。通常我们只谈论传统系统的特性或特征(比如，“传统 UNIX 系统只有一种类型的文件系统”)。虽然 SVR3 与 4.3BSD 的每个子系统都在许多方面差别很大，它们还是有一些共同点的，一般情况下我们只讨论它们的共性。在谈到现代 UNIX 时，我们是指 SVR4，4.4BSD，Mach，以及其他自它们派生的系统。“现代 UNIX 系统提供某种类型的日志文件系统”，像这样的一般性评述是指许多现代系统都有这种特性，并非是每一种都有。

1.5 参考文献

它后来又重新修订和打印为[Rite 78]。

西电公司曾是 AT&T 公司的一个子公司，后来 AT&T 公司分解了。贝尔实验室是由 AT&T 与西电公司联合成立的。

4.1BSD 共有 3 个独立的发行版本-4.1a，4.1b 和 4.1c。

用户数据报协议/网际协议。

IBM 的系统网络体系结构。