普通高等教育“十一五”国家级规划教材

第3版

UNIX操作系统教程

张红光 李福才 编著
南开大学

机械工业出版社
China Machine Press

普通高等教育“十一五”国家级规划教材

第3版

UNIX操作系统教程

张红光 李福才 编著
南开大学

机械工业出版社
China Machine Press

本教材第3版修订中删减了第2版中一些过时的UNIX技术和描述，增加了针对Linux系统设计技术和使用技术的阐述，增加了对UNIX环境编程实用技术的描述。其中第1章是有关UNIX技术的概述，第2～4章描述UNIX应用技术，第5～6章重点说明UNIX环境的编程技术，第7～11章阐述UNIX内核实现技术，第12～14章描述UNIX扩展知识。本书讲解细致，由浅入深，全面论述了UNIX系统的核心技术与操作系统的基本原理。

新版教材更加实用，非常适合作为高等院校计算机及相关专业的操作系统等课程的教材，同时也适合对UNIX系统技术感兴趣的广大读者自学。

图书在版编目（CIP）数据

UNIX操作系统教程/张红光，李福才编著. —3版. —北京：机械工业出版社，2009.11
（普通高等教育“十一五”国家级规划教材）

ISBN 978-7-111-28374-4

Ⅰ. U…　Ⅱ. ①张…　②李…　Ⅲ. UNIX操作系统－高等学校－教材　Ⅳ. TP316.81

中国版本图书馆CIP数据核字（2009）第172305号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：刘立卿
北京瑞德印刷有限公司印刷
2010年1月第3版第1次印刷
184mm × 260mm · 22.25印张
标准书号：ISBN 978-7-111-28374-4
定价：38.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

第3版前言

《UNIX操作系统教程》自2003年出版以来，得到许多高校师生的认同和支持，尤其是本书第2版自2006年1月第1次印刷以来，到目前已重印了9次，并被多所高校确定为授课教材。有很多读者通过多种渠道与我们联系，讨论本书涉及的教学和各种技术问题，并提出很多中肯的批评和建议。这些积极热情的反馈建议说明了读者对我们的支持和厚爱，我们在表示由衷感谢的同时也感到责任的重大。为回报大家的关爱，在本书出版近4年之际，我们决定进行第三次修订。本次修订是在前两版的基础上对教程内容进行的又一次重要调整，主要完成的工作包括：

1）删减了第2版中一些过时的UNIX技术描述；

2）增加了针对Linux系统的设计和使用方法的阐述；

3）将书中所有的举例程序都重新在Linux 2.6系统上进行了运行验证；

4）增加了对UNIX环境的编程实用技术描述。

修订后的教程将更便于读者自学，并可以做到边学习边实践。教程中给出的程序实例都在Linux 2.6环境中进行了调试验证，包括头文件和语法格式都进行了调整。这样做的意义是使大多数读者将学习重点放在理解UNIX相关概念和消化UNIX实用设计技术上，避免过多地关注高级语言的语法细节。另外，在设计书中的程序举例时，我们充分考虑了读者进行扩展练习的需要，一般都是给出一个最基本的格式，在这个格式下读者可以进行多种扩展，以完成一个典型的应用实例。

本书的目标是以介绍UNIX应用技术为起点，进而描述UNIX核心技术和实现策略，修订后的教程应该更适合本科教学和读者自学，因为无论是在系统性方面，还是在由浅入深的技术阐述方面，教程都有了进一步的完善和改进。修订后的教程共包括14章，其中第1章是有关UNIX技术的概述，第2～4章是UNIX应用技术的描述，第5～6章重点说明UNIX环境的编程技术，第7～11章是UNIX内核实现技术的阐述，第12～14章是有关UNIX扩展知识的描述。这些章节的内容，构成了我们为读者设计的学习UNIX技术的不同阶段，即学习UNIX基本知识的第一个阶段，掌握UNIX实用技术的第二个阶段，理解UNIX核心实现技术的第三个阶段以及了解和掌握UNIX扩展知识的第四个阶段。经历过这些阶段的学习和实践，希望能使读者对UNIX操作系统技术有一个比较全面的理解和认识，同时还能具备一定的UNIX环境编程开发能力。由于UNIX系统的核心实现技术是操作系统原理的很好应用和延展，因此在学习过程中最适宜的做法是采用理论与实践相结合的方式，循序渐进地掌握操作系统理论。希望这种学习方式可以提升读者对操作系统技术的学习兴趣，能够适当地减轻传统操作系统教学中理论枯燥、概念抽象等给人带来的困扰。

本书适合用做高等院校计算机及相关专业的UNIX系统技术和操作系统课程的教学和辅导教材，同时也适合对UNIX系统技术感兴趣的软件开发人员自学。学习本书内容，需要读者具备一定的计算机体系结构和C语言程序设计的知识（当然，这些知识的掌握也可以同步进行）。在进行教学时，课程实验环境建议建立在Linux 2.6系统中，基本的编程语言采用C语言。学习中除了本教程中给出的例题外，读者还可以参考与本教程第2版配套的实验教材。需要指出的是，书中的这些内容对读者只是起到抛砖引玉的作用，读者应该在基本知识建立的基础上充分发挥自己的主观能动性，多做实验，做好实验，力求将UNIX系统知识和操作系统理论融会贯通，真正领会其中的技术真谛。

本书的第1、2、11章由李福才修订，其余章节由张红光修订，张红光对全书进行了统稿。作者

在本次修订过程中参阅了大量的相关著作和资料，并就修订的相关专题与有关专家和学者进行了多方面的探讨，我们还非常重视南开大学历届使用本教材的本科生阅读学习的意见，同学们在使用本书中曾提出了许多非常宝贵的反馈意见和修改建议，这些我们都力图在本次修订中加以体现。

另外非常感谢对本书的编写做出无私贡献的所有人员，对直接参与并为完成本教材修订做了大量工作的潘岳、张勇、兰旭泽等人表示衷心的感谢，另外还要感谢机械工业出版社华章分社的编辑们，正是他们卓有成效的努力和细致用心的工作才使本书得以顺利出版。

尽管在本书修订和编写过程中，我们力图认真工作、细心整理，但由于水平所限，书中难免存在谬误和不足之处，我们诚恳地期待读者的批评指正。联系的电子邮箱是：zhanghg@nankai.edu.cn。

作者

2009年8月于南开园

目　录

第1章 绪 论

UNIX系统自20世纪70年代诞生以来，经历了30多年的风风雨雨，从一个最初简单的文件管理软件，变成了今天各类计算机系统中一个重要的操作系统。与20年前相比，人们已不再怀疑UNIX系统是否会被市场和用户所认可，是否能在业内流行并生存发展下去，也不再会为“是否需要继续进行UNIX系统研究”等论题而展开无休止的争辩。UNIX在当今的世界上已经拥有了几千万个用户，并且这个数字还在不断地增长。今天，UNIX系统运行在世界上各种不同类型的计算机系统中，特别是在工作站、小型机及中型以上的计算机系统中，UNIX往往是首选操作系统。UNIX以它日臻完善的系统管理、调度技术，友好的人机交互界面，完善而又实用的各种编程工具、命令，及系统运行的可靠性、稳定性给每个使用者留下深刻的印象。纵观UNIX系统从实验室走向市场，走向世界的过程，我们可以深切地感受到：UNIX能够在巨大而又残酷无情的市场中顽强地生存下来，并一步步发展光大，完全源于其完美的技术内涵和对操作系统所要解决问题的正确理解，还有对市场和用户需求的准确把握。同时我们也清楚地看到，在UNIX的发展过程中，正是由于UNIX的设计开发者们顺应了市场的需求，及时满足了各种用户不同的应用需要，不断地调整自己的发展策略，不断扩大自身的功能，开发出无数优秀的应用模块，才取得了今天在计算机领域不可替代的地位及不断推广壮大的业绩。

1.1 操作系统概述

在了解UNIX系统之前我们需要对操作系统的基本概念做一些描述，以利于理解UNIX系统的相关概念和所涉及的设计问题。

从基本概念上看，各种操作系统完成的功能是相似的。操作系统在计算机系统体系结构中所处的位置很特别，它的一边面对的是应用程序和用户，另一边面对的是计算机系统的硬件。操作系统的职责是协调计算机内部所有的活动，为用户和应用程序构造一个开发和运行的虚拟环境，这个虚拟环境比计算机的实际环境具有更加友好、更加便利、更加有效等特性。

1.1.1 建立操作系统的目标

最初的计算机中是没有操作系统的，但随着计算机技术的推广应用，人们深切感受到，计算机硬件无论怎样强大也不能真正解决用户的应用问题。在计算机应用中需要由操作系统来控制应用程序的执行过程，并充当用户、应用程序与计算机硬件之间的接口，因此建立操作系统有以下三个主要目标：

- 方便计算机的使用：有操作系统做支持，计算机的操作会变得更加友好和方便。
- 有效地使用计算机资源：在操作系统的管理下，计算机的系统资源可以得到有效的利用。
- 便于计算机技术的发展：计算机技术的发展可以不断地提供新的应用功能和产品，将这些功能和产品有效地表现和应用在系统中需要靠操作系统体系结构的扩展能力。

1.1.2 操作系统是用户与计算机的接口

随着计算机的普及，使用计算机系统的用户会涉及各个层面和各个领域。使用者往往不关心系统结构和内部实现技术细节，他们关注的问题是计算机系统将如何在实际中得到应用，将如何解决他们实际中碰到的应用问题。

操作系统在实际中承担起了用户与计算机系统的接口作用。比如，操作系统在执行I/O操作、与用户程序通信和控制用户程序时使用了大量的中断。中断机制使操作系统可以分配优先级、从一个用户程序转换到另一个用户程序、实现安全和保护特性、协调各种I/O的活动。但是如果将这些中断机制完全交给用户使用，将会给用户带来极大的不便，也对用户提出了不切实际的高标准要求。通常采用的方式是由操作系统用中断服务程序将所有与计算机相连的设备组织在一起，应用程序本身并不执行实际的I/O操作。当应用程序需要进行I/O操作时，它首先指明需要传输的数据，然后请求操作系统执行相应的操作；这时由操作系统控制暂停执行该程序转去执行请求的I/O操作。当操作结束后，再将控制权传回给应用程序。这些过程应用程序和用户并不需要完全知道，由操作系统完成管理和切换，使用户的工作与系统解决的问题分开处理。操作系统用这种方式为用户和应用程序提供各种的服务，这些服务构成了用户与计算机系统的接口，形成了计算机的运行与执行环境。

操作系统中通常包含以下服务内容：

- 程序设计开发环境：提供各种软件设计服务，如编辑器、编译器、调试器等，用于协助程序员完成软件的设计与开发工作。这些服务通常以实用程序的方式提供，严格地讲它们并不属于操作系统核心中的内容，而是由操作系统提供的开发工具。
- 程序并发运行环境：程序完成后交给计算机系统执行，需要在操作系统支持和管理下才能正确运行。比如需要将指令和数据加载到系统主存中、要对使用的I/O设备进行初始化、要准备使用的系统资源。在多道环境中还要考虑多道程序并发执行的问题，这是操作系统设计中需要解决的主要技术问题。
- I/O设备使用功能：每个I/O设备都有其独特的指令集和控制信号组，通过操作系统的管理，隐藏每个设备控制过程的细节，为用户提供统一的接口和使用模式。
- 文件访问与控制管理：用户对文件的认识比较抽象，而操作系统不仅需要了解存储文件的I/O设备性能，还要确定文件在存储介质中的数据结构，以及多个用户使用文件时如何进行共享管理提供保护机制等问题。
- 系统核心资源的访问与保护：在计算机系统中有一些核心模块和硬件资源，为了保证这些资源为不同的用户提供服务，需要设定访问权限和保护措施，避免系统软件被恶意或无意地破坏，硬件资源被强行占用。
- 系统错误检测管理：计算机系统运行过程中会出现各种错误和问题，当出现这些问题时，操作系统要能够检测到并有相应的应对措施。在错误处理与系统恢复时要尽量保证对运行的程序影响最小。
- 统计与记账管理：对程序运行中使用的资源进行跟踪和统计，用于今后的记账管理。

1.1.3 操作系统是资源管理器

计算机系统的运行需要软件和硬件的协同工作，将系统中的各个硬件功能部件和软件功能模块看成系统资源，操作系统就是这些资源的管理器。如何有效合理地使用各种资源、为用户提供最佳的服务是操作系统的主要工作内容。在资源管理中将包括以下工作内容：

1）跟踪记录资源使用情况：在多道任务系统中，系统资源要满足多道任务的需要，但一个任务对资源的需要可能并不是连续的，这就需要对资源使用情况进行跟踪和记录，了解当前资源状态或剩余情况，以满足任务对资源的请求。

2）分配或回收资源：在条件满足的情况下将资源分配给请求的任务，分配后要记录当前资源剩余情况和状态，以备下一次的分配；根据任务完成的情况适时地回收系统资源，保证新进程的请求。

3）提高资源的利用率：在操作系统的管理下力图使系统资源得到合理、高效的使用。

4）协调多个进程对资源请求的冲突：当少量资源为多个请求服务时，会产生资源使用冲突，这时操作系统需要分析请求进程的特性，对资源使用做出决策，协调各进程合理地使用资源。

1.2 UNIX系统的主要特性

UNIX系统实现技术中有很多优秀的技术特点，在操作系统的发展历程中，它们一直占据着技术上的制高点。很长一个阶段中，UNIX是许多其他操作系统学习模仿的样板。UNIX系统的特点和优势很多，此处我们仅列出几个主要的特征，便于大家对UNIX系统有一个初步的了解。

1. 用简单的设计技术和方法去完成较复杂、较全面的功能

在UNIX系统的设计中，所采用的最基本的设计思想是将复杂的问题进行分解，用最简单、最基本的功能模块做堆积、连接、组合来解决复杂问题。这样在设计上不但可以保证每个基本模块功能单一、易于实现、设计结构清晰，而且组合使用的效果也会比较理想。也正是因为这种设计思想的出现，才引发了软件规范化模块化设计、软件构件可重用理论与方法的研究，这也是当今软件工程设计的重要思想。随着基本设计模块的不断积累，设计库的不断丰富壮大，UNIX系统的延展性、可移植性得到了充分的发挥，并大大缩减了设计的工作量和工程实现时间。这也是UNIX系统不断被新的计算机系统所接受的重要原因之一。

2. 支持多用户、多任务的运行环境

由于UNIX系统内部采用分时多任务调度管理策略，它不但可以支持某一用户在某一时刻和某一地点上的多种请求，而且能够同时满足多个用户的相同或不同的请求。采用多用户分时多任务调度管理策略，计算机可为多个用户的一般性请求提供服务。比如，用户可以在进行数据处理的同时向另一个用户发送电子邮件；在边播放音乐的同时浏览相关网站的信息。这种方式与我们在日常生活中做事的情况很相似。一般微机上的UNIX系统就可以支持多个用户的同时请求，在大型机中运行的UNIX系统，更是可以支持几百个用户同时进行工作。由于UNIX具有良好的多用户分时多任务调度管理特点，这些共同使用UNIX系统的用户并不会感觉到所使用的计算机资源被分割、被抢占，而是感觉自己在独占计算机资源。

3. 文件系统可随意装卸

由于UNIX系统采用模块化的结构进行设计，为了便于系统构造和用户使用，其文件系统是可裁剪的。用户使用文件系统时，可根据需要构建独特的文件系统并将它对应于某个指定的硬件存储设备。使用时可以加载，用完后卸载。这样做可以最大限度地保证用户使用数据时的方便性和安全性；而对UNIX系统设计来讲，这样的措施可以保证系统的简洁性。

4. 良好的开放性和可移植性

随着计算机技术的发展，各种类型的计算机产品在不断地更新和发展。不同的计算机系统其内部硬件结构可能会有很大的差异。因此，任何一种操作系统的固定模式都很难做到可以适应所有的硬件平台。如果一个操作系统的适应性太差，将其应用在一种新型的计算机系统中，就可能有大量的原有系统软件需要重新设计，这对快速更新的计算机类型来讲是非常不利的。从应用层面上看，用户常会碰到这样的情况：已熟悉了一种应用软件的使用方式、数据格式，因为硬件环境或操作系统的变更而使得该软件无法使用，不得不重新学习一种新的同类软件。这将给用户带来很多不便并造成一定的资源浪费。从系统设计角度看，因为操作系统软件的开发是一个庞大的软件系统工程，它不同于一般软件的设计，众所周知，它的设计难度大、工期长、资金耗费大。若每一次硬件的改变都需要重新设计操作系统，不仅会使用户许多珍贵的应用数据和应用软件模块遭受破坏，给用户带来重大的损失，而且还会使操作系统设计工作出现大量的重复劳动，同时还无法保证系统的可靠性和安全性。由此看来，操作系统的适应性差是系统设计者和用户都不能接受的事情。因此，操作系统的开放性和可移植性是衡量其优劣的一项重要指标。UNIX系统由于其内核设计的许多特征，

尤其是UNIX系统内核的大部分是用C语言实现的，这就使得操作系统的移植工作变得比较容易；同时也是UNIX系统拥有众多用户群以及不断有新用户加入的重要原因之一。

5. 强大的命令功能

UNIX系统的命令功能非常强大，用一个简单的命令就可完成其他操作系统需要花费许多时间去做大量编程设计才能实现的功能。随着用户对系统和命令使用的不断熟悉，他们可以灵活应用、巧妙结合UNIX命令和系统内部提供的服务，用一条复合性命令完成其他操作系统需要花费几条到几十条命令才能完成的动作或功能。UNIX在命令使用上为用户提供了极大的便利，考虑的因素比较周全，正是由于这些，许多UNIX老用户提起UNIX系统的性能总有一种如数家珍的感觉。

6. 完善的安全机制

在UNIX系统中，由于它的开发是基于多用户的环境进行的，因此在安全机制上考虑得比较严谨，其中包括了对用户的管理、对系统结构的保护及对文件使用权限的管理等诸多因素。许多业内的专家认为，与其他系统相比，基于UNIX系统平台构筑的信息系统及用户安全管理机制是比较完善比较可靠的，可以为应用提供良好的基础平台。当今，在我国和世界各地许多关键性行业建立的信息化管理系统不乏采用UNIX系统平台的例子，它们都表现出良好的运行性能。

7. 具有网络特性

在UNIX系统中，由于支持多用户的需要，强调了其内部通信机制及对外部设备的易接入性，并使其对当今网络环境的支持非常自然顺畅。在许多新版UNIX系统中，更是增加了对TCP/IP协议的支持，使UNIX系统的网络连接变得更加容易和便捷。优良的内部通信机制，方便的网络接入方式，快速的网络信息处理方法，使UNIX系统成为构造良好网络环境的首选操作系统。

1.3 UNIX系统的发展史

对UNIX系统的研究可追溯到20世纪60年代末期，开始是由AT&T Bell实验室的研究人员Ken Thompson和Dennis Ritchie首先进行的。在今天看来，UNIX系统所展现和产生的商业价值是无比巨大的。但在当时，UNIX的构思与设计，既不是由Bell实验室的管理层提出的指令性计划，也没有某种商业利益的驱使，而是完全由于像Ken Thompson和Dennis Ritchie这样一些有独创思想的人，为了个人工作的需要而萌发出的创造的冲动。尽管在以后的30多年中，有众多的大公司和学校及科研单位的技术精英为UNIX的不断发展、壮大做出了巨大的贡献，但今天我们谈起UNIX来还是要感谢Ken Thompson和Dennis Ritchie，毕竟是他们创建了UNIX，使后来人受益匪浅。

纵观UNIX的发展史，尽管有强大的Bell实验室做后盾，但它还是经历了各种波折和风雨。UNIX的发展过程有以下几个阶段：

1. 20世纪60年代到70年代，完成了系统内核的雏形

AT&T Bell实验室的K. Thompson及Dennis Ritchie都曾参加过Multics（多路存取计算机系统）系统的研制工作。在20世纪60年代开发的Multics中蕴含着许多现代计算机的新思想和新技术，但由于始创者的原因，Multics的设计过于复杂和粗糙，以至于远离了实际应用的需要，未能得到推广。但正是由于有了开发Multics的过程和经历,才培育了一批具有创造力的天才。也正是由于有了设计Multics的经验，设计者们提出了多用户、分时系统的想法，这些都在后来UNIX系统的研制中起着非常重要的作用。

退出Multics的研究后，Bell实验室的工作人员为了改善自己的工作环境，也为了在今后的工作中能够用到现代操作系统的技术和方法，他们决定创造一个自己的操作系统。很快，K. Thompson和Dennis Ritchie等人根据研制Multics的经验创造了新系统，被命名为GECOS，其主要功能是一个文件管理系统。后来他们在一台废弃的PDP-7机上编制了“Space Travel”（太空旅行）的程序，由于这一程序在当时的计算机上实现过于复杂，他们不得不充分地利用文件管理系统的

优势对“太空旅行”程序进行设计和管理。后来他们在PDP-7上重写了GECOS的代码，并且完成了从PDP-7到GECOS的交叉汇编功能。这样设计者们在PDP-7上构造了一个新系统，由于它包涵着Multics的技术精华，又比Multics简单实用，因此开发者为其取名UNIX，这就是UNIX系统的雏形。

2. 1978年，UNIX版本7面世

从UNIX系统雏形建立后，设计者们对它进行了多次的版本更新和功能扩展。除了完善文件系统的设计外，还增加了调度处理、系统界面设计、命令行结构管理等功能。最初的UNIX系统和当时的其他操作系统一样是用汇编语言完成的。在1971年，Bell实验室的开发人员研制出了一种高级语言C，C语言当时主要是为操作系统设计而研制的。1973年，UNIX系统的设计者们将UNIX系统的内核用C进行了重新编写。用高级语言编写操作系统的绝大部分内核代码，在当时也是一种创举。显然相对于用汇编语言编写的系统来说，它的代码量有所增加，运行速度有所降低。但它的易读性、易改性、易移植性得到了显著提高，并且降低和改善了代码与硬件之间的依赖关系，大大提高了工作质量与工作效率，使得操作系统的开发设计者们获益匪浅。因此，这一重大变革也可以说是今天UNIX系统能够得到迅速推广和发展的一个重要转折点。

1978年UNIX的第7版问世，当时UNIX系统已具有了相当的规模，系统功能已经比较完善。其中UNIX核心部分是由一万条指令构成的，而90%是用C完成的。AT&T当时向世界的各所大学发布和拷贝该系统，掀起了一场UNIX系统研究开发的巨浪。正如有人所说：20世纪70年代，整整一代计算机科学家是在UNIX系统中成长起来的。

3. 20世纪80年代初，UNIX商业化阶段

从20世纪70年代末到80年代初，UNIX系统的发展势头使商家从中看到了商机。许多商家开始了商业运营，当时形成了多个UNIX系统研究开发的组织机构。如Bell实验室派，他们完成了UNIX新版本的开发，继第7版后又研制出了UNIX System III、UNIX V.1、V.2等；以美国加州大学伯克利分校为主的大学派，在UNIX第7版的基础上研制了BSD 3、BSD 4.0、BSD 4.1、BSD 4.3等；以微软为主的软件开发商派研制了微机版本UNIX系统，XUNIX系列；以IBM和HP公司为首的计算机制造商也相继研制出了相应的UNIX版本，如AIX、HPUNIX等。在那段时间里，UNIX系统可谓是遍地开花，研制的产品也五花八门、各有所长。

此时，许多商业版本的UNIX系统虽然都采用了UNIX第7版作为系统的核心，但在应用功能和应用界面上已经有了很大的改变，在一些优秀的版本中已出现了许多现代操作系统的特征。因此，这一时期的UNIX系统发展为当今商业版的UNIX系统的开发成功积累了丰富的经验。

4. 1988年后，UNIX的标准化阶段

商业集团的参与使UNIX的各种版本开发速度加快，UNIX系统的设计与开发朝着系统功能越来越齐全、软件模块越来越丰富、系统规模越来越庞大、用户群越来越多的方向发展。在这个阶段，用户手中会有多个变异的UNIX系统。虽然各个开发商都称自己的系统与某某UNIX系统标准版兼容，但用户在使用中却实实在在地感到了它们中存在的越来越大的差异。用户常常因为需要使用一种软件工具而必须安装支持这一软件的操作系统，而使用另一种软件时还要安装支持它的另一种操作系统，结果使用户机器中经常安装了很多种操作系统，给工作带来了极大的麻烦。若任由这种局面发展下去，势必会产生无法计数的变异UNIX系统的混乱局面，也必将会给用户带来更大的不便。久而久之，UNIX很可能会因此而失去大量的用户群，从而断送它来之不易的发展时机。鉴于这种情况，几个最具实力的开发商向业界提出了强化UNIX系统设计开发，制订UNIX系统标准化等问题的倡议。

1988年春，AT&T与SUN公司结盟，宣布开发UNIX System V.4，在其中包容了AT&T的System V、Sun OS、XENIX的各种特性，并要将其作为一种UNIX操作系统的技术标准进行发布。

1988年中期，一些系统销售商为了占有市场份额，也组织了一个联合体（最初是由IBM，HP，

DEC发起的），称为OSF（开放软件基金会），致力于开发独立版本的UNIX 系统。现在看来，当时这些世界上最具实力的厂商和集团公司的做法，虽然带有浓重的商业色彩，但在实际中却起到了促使UNIX系统的设计与开发走向标准化的作用。这一时期制定了许多优秀的研制开发工具和开发应用接口的标准，经权威部门认证后，将其正式作为UNIX系统的标准，并要求今后的开发者们必须遵照其执行。如著名的Motif用户界面标准就是这一时期的产物。毋庸置疑，这种由国际权威机构制定标准，并由许多业界著名大公司、大集团参加的竞争机制对UNIX系统的发展起到了积极的作用，从而使UNIX系统的开发工作进入了一种良性的竞争发展轨道。

5. 20世纪90年代后，并行处理及分布式网络系统的发展

进入20世纪90年代后，计算机技术最显著的发展是并行多处理技术和分布式网络处理技术的实现。计算机系统中出现的多处理机并行处理、多用户系统中分布处理的请求以及网络资源的共享等都为UNIX系统提供了新的发展空间。这个时期的UNIX系统主要发展了并行处理功能、分布式处理功能和网络服务能力，如RS/6000上的新版AIX系统，Sun服务器上的Solaris系列版本等就是这类系统的典范。

1.4 开源软件与UNIX的推广发展

众所周知，一个优秀的软件是要靠使用群体和修改群体支撑的，使用群体不断地反馈应用中的实际需要，修改群体不断地将软件中存在的错误进行修正，从而将软件设计得更加适应实际应用的需要。而开源软件的发展方式恰恰可以满足这种需要，正是由于这一点UNIX在其发展过程中与开源软件建立了密不可分的关系。

1.4.1 开源软件

严格意义上的开源软件是指开放源码（Open Source）的软件，它们是在一些非营利的软件组织（如美国的Open Source Initiative协会）进行了注册，并对软件进行了正式的定义、可以被公众使用的软件。通常这种软件的使用、修改和发行将不受许可证的限制。

建立开源软件的目的是为了打破商业软件在市场上占垄断地位的状况，使得软件的发展可以沿着一条更有生命力、更加健康的道路发展。开源软件的发展思想是由美国的Richard Stallman提出的，早在1971年他就在麻省理工学院开始对此项工作进行了前期准备，1984年在一些企业的支持下Richard Stallman以自由软件基金会（FSF）主席的身份启动了“自由软件联盟规划”，开始了这项工程浩大的工作。

开源软件的倡导者们对于开放源码软件进行了定义，其中包含了四种自由，它们是：

1）无论出于何种目的，都可以自由运用该软件。这是第0种自由，也是最基本的一种自由。

2）可以自由学习该软件的工作过程，并使之适应使用者的需求。这是第1种自由，这种自由包含着一种含义，就是可以自由地读取源程序。

3）可以自由重新分发拷贝，以便帮助使用者的朋友。这是第2种自由。

4）可以自由改善该程序，并发布给公众，让整个社会得利。这是第3种自由。

开源软件的发展是一种技术进步的象征，正如一位著名的开源软件倡导者指出的那样：“Open Source is a process, Open Source is not a product（开源软件是一个过程，而不是一种产品）。”由于开源软件发展方式非常适合于实现软件快速、分布式的开发，在软件开发中可以聚集最佳的人才和最好的设计蓝图，并且可以实行一种适合软件设计的最有效的人员管理模式，因此一经提出，就得到了各方极大的支持和响应，同时也使这种设计思想和方式得到了快速推广并取得了一些实效。

开放源码的精髓是使用者可以使用、复制、散布、研究、改进软件，分析开源软件的思想可以发现，开源软件同时涉及源码本身和开发过程中的许多问题。主要涵盖了三个方面的意义，即免费分发源代码、建立模块化的体系结构以及集市式的开发方式。这里所谓集市式的开发方式是

指，任何地方的任何人都可以参与到最终软件产品的制造中去，这是一种在网络环境下可以实现的理想化软件设计模式。开源软件中包含的三个方面有着密切相关性，因为源代码分发可以使开发者之间共享设计成果，模块化体系结构可以使分布式开发具备良好的基础，而集市式的开发过程会给开源软件提供强大的改错能力——因为这种机制将程序中的错误公开给了数量巨大的用户，而这些用户一方面是使用者，同时又都是潜在的改错者。另一方面，由于任何人都可以复用和发行开源软件代码，这事实上又支持了公众利益，从而使创新的观念被整个集市所共享。

美国一些进步的评论家指出，在网络这样的虚拟环境中，驱动软件系统的底层代码，尤其是广为人知的那些应用程序之间的通信协议，在某种意义上很像现实社会中的法规。换句话说，这些代码对网上的行为给出了一些规范，它鼓励某些行为，而限制其他一些行为，就像现实社会的法律一样。因此从这个意义上讲，开放源码带来了一个更民主的软件开发方式，在这种方式下，好的主意将被集体分享，而不是作为智力资本被某个人私藏起来。

最典型的开源软件是在1991～1992年间诞生的，芬兰的Linus Torvald制造了第一版的Linux操作系统，然后在一群热心的程序员的努力下，把Linux操作系统以及外围的应用程序逐一打造成具有实用价值的产品。由于Linux是一个UNIX的仿照系统，因此该系统的普及和应用也给UNIX系统技术本身带来了新的春天。由于内核代码公开，极大地激发了人们对这一技术的研究兴趣，人们对UNIX技术本身更加关注，使用和了解它的人群越来越多，这样也促进了UNIX技术的不断创新和发展。近年来，除了日趋成熟的Linux操作系统外，还有Apache网页服务器、Perl程序语言、MySQL数据库、Mozilla 浏览器、OpenOffice等开源产品不断面市。可以预料开源这股旋风将会掀起一场新的软件革命，会给软件设计这项技术带来新的活力。

1.4.2 促进UNIX发展的重要组织机构

在UNIX的发展中，的确有一些组织机构发挥了重要的促进作用，正是由于他们的努力，才使UNIX系统和相关的技术得到了发展和创新。

（1）联合体组织

早期支持UNIX发展的一些联合体包括USG（Unix Support Group, UNIX支持小组）、USDL（Unix System Development Laboratory，UNIX开发实验室）和USL（Unix System Laboratory，UNIX系统实验室），这三个实体后来经过商业运作派生出了AT&T贝尔实验室，该组织在后来UNIX的标准化过程中起到了非常重要的作用。

（2）大学组织

由于早期的UNIX系统对大学是免费使用的，这促进了大学研究开发UNIX系统的热潮，做得最好的是加州大学伯克利分校，他们推出了BSD（Berkeley Software Distributions）UNIX版本。

（3）开放式自由软件组织

开放式自由软件组织是在20世纪90年代发展起来的，由于该组织的宗旨符合软件设计的规律，因此在组织成长和软件成果上都卓有成效。Linux系统在这个丰沃的土壤中得以茁壮成长，它的发展融入了以上两个组织发布的UNIX系统的特点。

1.4.3 各种UNIX系统分支

应该指出的是，在UNIX发展的过程中产生了许多基于UNIX System V（简称UNIX SV）的变种版本。这些UNIX系统在命令使用上和系统表现特征上都与UNIX的SVR4（UNIX System V第4版）很相似，例如：

1）AIX：这是一个由IBM公司主持研究的UNIX操作系统版本，它与SVR4兼容。主要是针对IBM的计算机硬件环境对UNIX系统进行了优化和增强。

2）HP-UX：是HP公司的UNIX系统版本，该系统是基于UNIX System V第2版开发的。它主要