

精通Linux丛书

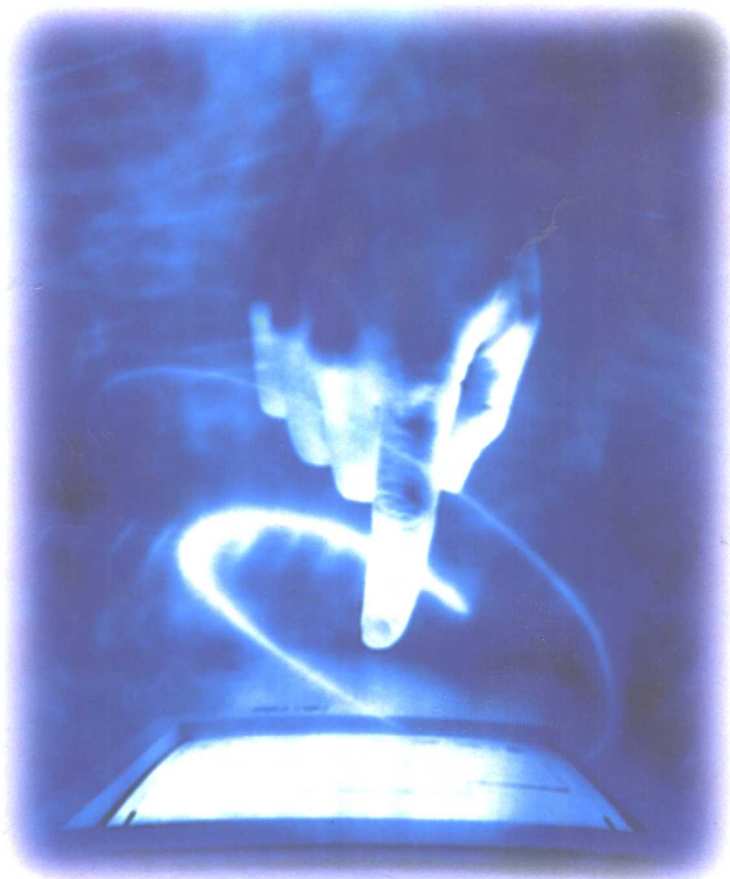
# Linux

徐严明 等 编著

编

程

指南



科学出版社

TDS17 80



精通 Linux 丛书

# Linux 编程指南

徐严明 等 编著

科学出版社

## 内 容 简 介

本书主要介绍 Linux 系统环境下的开发工具、编辑集成环境、系统调用、内核剖析、编程技巧等内容。

本书深入浅出,通俗易懂,对于每个编程技巧都有具体的实例,力求使读者做到概念清楚,迅速掌握编程技巧。

本书适合 Linux 的中、高级用户和系统管理员、Linux 下应用程序开发人员及广大 Linux 爱好者和喜欢 hacking 的用户;也可以作为计算机专业学生学习操作系统、数据结构课程的辅助教材。

### 图书在版编目(CIP)数据

Linux 编程指南 / 徐严明等编著. —北京: 科学出版社, 2000

(精通 Linux 丛书)

ISBN 7-03-007949-3

I. L... I. 严... III. Linux 操作系统-程序设计-指南  
N. TP316.89-62

中国版本图书馆 CIP 数据核字 (2000) 第 09410 号

科学出版社 出版

北京东黄城根北街 16 号  
邮政编码: 100717

北京双青印刷厂 印刷

科学出版社发行 各地新华书店经销

\*

2000 年 6 月第 一 版 开本: 787×1092 1/16

2000 年 6 月第一次印刷 印张: 20

印数: 1—5 000 字数: 471 000

定价: 26.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

## 序

Linux 是目前最流行的计算机操作系统之一。Linux 的最初版本是芬兰的 Linus Torvalds 于 1991 年独立开发的。由于它免费提供源代码和可执行文件，并且公布在因特网上，因此从一开始就吸引了世界各地的 UNIX 行家为其编写了大量的驱动程序和应用软件，在短短几年时间里，Linux 就迅速地发展成为一个相当完善的操作系统。现在国外使用 Linux 作为各种网络服务器的站点已经超过了 60%，而国内由于接触 Linux 的时间并不长，还没有这方面的统计数字，估计占有率不会超过 10%。相信随着 Linux 的普及，会有越来越多的站点使用 Linux。

Linux 可以认为是 UNIX 在微机上的完整实现。众所周知，作为网络服务器及高性能工作站的主流操作系统，UNIX 经过了三十几年的发展完善，已经相当可靠和稳定；但遗憾的是，UNIX 大多运行在昂贵的工作站上，普通人难得一见。现在有了 Linux，任何人都可以在微机上学习和使用 UNIX 了。

Linux 的主要特点有：

(1) Linux 系统与 UNIX 系统非常相似，具有 UNIX 操作系统的优点，比如多用户、多任务等；但是 Linux 是在最常见的微机上运行，方便用户学习和使用。

(2) 系统模块化。Linux 的设计思想是模块化设计思想，用户可以自由选择安装模块，同时用户可以根据实际需求，开发有特定功能的微内核操作系统。

(3) 网络功能强大。作为网络服务器操作系统，相对于目前流行的 Windows NT，Linux 更加稳定可靠，并且网络处理速度更快。

(4) 相对于 Windows 操作系统，计算机硬件资源的使用效率更高。

(5) 支持多种文件系统。

(6) 可移植性好。Linux 完全支持 POSIX（可移植操作系统接口）规范，UNIX 下的许多应用程序可以很容易地移植到 Linux 下，反之亦然。

Linux 强大的功能、开放式模块化系统及其免费自由等特点，使得用户可以在 Linux 基础上充分发挥想象力和创造力，开发出丰富多采的应用程序和系统。比如，目前流行的“商务通”，它是用 Windows CE 开发的；但是完全可以采用 Linux 系统，开发一个微内核的操作系统，替代 Windows CE，同时其功能更加稳定可靠，开发成本更低。

为了便于读者学习和了解 Linux 操作系统，能够在 Linux 环境下开发应用程序，本书选取了以下内容：

• 《TurboLinux（中文版）使用指南》 主要介绍 TurboLinux 的安装和基本使用，适合于初级读者。

• 《Red Hat 6.0 使用指南》 主要介绍 Red Hat Linux 的安装和基本使用，适合于初级读者。

• 《X Window 使用指南》 主要介绍 Linux 环境下 X Window 的安装、配置、使用和编程，适合于初、中级读者。

- 《Linux 系统管理》 主要介绍 Linux 系统管理, 帮助读者成为系统管理员, 适合于初、中级读者。
- 《Linux 编程指南》 主要介绍 Linux 环境下的编程技术, 适合于初、中级读者。
- 《Linux 网络技术》 主要介绍 Linux 各种网络技术, 比如 WWW 服务器、FTP 服务器等, 适合于初、中级读者。
- 《Linux 网络编程》 主要介绍 Linux 环境下的网络应用程序开发技术, 适合于初、中级读者。

编 者

1999 年 12 月于清华大学

# 前 言

Linux 充分体现了操作系统的发展趋势,即开放、稳定、标准。Linux 是类 UNIX 操作系统,它继承了 UNIX 的特点并有所发展。作为 UNIX 类操作系统,它具有下列基本特征:

- 采用页式存储管理。
- 支持动态链接库。
- 提供具有内置安全措施的分层的文件系统。
- 提供 shell 命令解释程序和编程语言。
- 提供强大的管理功能,包括远程管理功能。
- 具有内核的编程接口。
- 具有图形用户接口。
- 大量的高级程序设计语言已移植到 Linux 系统上,因而它是理想的应用软件开发平台。

另外, Linux 还有许多独到之处:

- Linux 许多组成部分的源代码是开放的。
- 可以运行在许多硬件平台上。从 Linux 2.0 开始,不仅支持单处理器的机器,还能支持对称多处理器(SMP)的机器。
- 支持大量的外部设备。
- 支持的文件系统多达 32 种。
- 不仅可以运行许多自由发布的应用软件,还可以运行许多商品化的应用软件。
- 通过各种仿真软件, Linux 系统还能运行许多其他操作系统的应用软件,如 DOS, Windows, Windows NT 等,都有相应的仿真软件来运行它们各自的应用软件。

要在 Linux 环境下开发应用程序,就必须了解 Linux 系统的开发环境、系统组成的必要知识和程序开发技巧。本书主要包括:

- (1) Linux 下的系统开发工具,如 gcc, g++, gdb 等。
- (2) 编辑器和集成环境 vi 和 Emacs。
- (3) Linux 的内核剖析,包括内存管理、进程管理、文件系统、设备驱动程序、网络、进程通信、内核机制、源码结构等。
- (4) 系统编程,包括 Linux 核心与系统调用、Linux 文件系统及内部结构、文件与控制终端的系统调用、进程管理、进程通信、存储管理、网络编程等。这部分内容分别用 C 和 C++ 实现。

本书深入浅出,通俗易懂,适合 Linux 的中、高级用户和系统管理员、Linux 下应用程序开发人员、广大 Linux 爱好者及喜欢 hacking 的用户;也可以作为计算机专业学生学习操作系统、数据结构课程的辅助教材。

本书由傅宇旭、陈嘉繁、叶青等策划,由严民军、徐严明、吴加、傅宇旭、李勇、任志勇、陈力、邓涛、王宇、许钢、李策、薛政宇、郑小平等编写和录入。由于作者水平有限,书中错漏和不当之处在所难免,恳请读者批评指正。

编 者

# 目 录

序	
前 言	
第一章 Linux 的历史和功能简介	(1)
1.1 什么是 Linux	(1)
1.2 Linux 的历史	(2)
1.3 Linux 的特点	(3)
1.4 Linux 与自由软件及 GNU 的关系	(4)
1.5 Linux 的组成	(6)
1.5.1 存储管理	(6)
1.5.2 进程管理	(6)
1.5.3 文件系统	(7)
1.5.4 进程间通信	(7)
1.6 Linux 的用途	(7)
1.6.1 个人 UNIX 工作站	(7)
1.6.2 X 终端客户	(8)
1.6.3 X 应用服务器	(8)
1.6.4 UNIX 开发平台	(8)
1.6.5 网络服务器	(8)
1.6.6 Internet 服务器	(8)
1.6.7 终端服务器、传真服务器、Modem 服务器	(9)
1.7 Linux 的发行版本	(9)
1.8 红帽子 6.0 的新特性	(10)
1.9 Linux 的不足之处	(10)
第二章 基础知识	(12)
2.1 硬件基础	(12)
2.1.1 CPU	(12)
2.1.2 存储器	(13)
2.1.3 总线	(14)
2.1.4 控制器和外围设备	(14)
2.1.5 地址空间	(14)
2.1.6 定时器	(15)
2.2 软件基础	(15)
2.2.1 计算机语言	(15)
2.2.2 操作系统	(17)
2.2.3 内核数据结构	(20)
2.3 小结	(21)
第三章 Linux 程序开发工具简介	(22)
3.1 C 语言的编译器和调试器	(22)

3.1.1	GCC 的基本用法 .....	(23)
3.1.2	一些常用的选项 .....	(23)
3.1.3	优化选项 .....	(24)
3.1.4	用 gdb 调试程序 .....	(24)
3.1.5	图形的调试工具 xxgdb .....	(27)
3.1.6	cproto .....	(28)
3.1.7	grpof .....	(29)
3.1.8	其他工具 .....	(29)
3.2	C++ 语言简介及其编译器与调试器 .....	(29)
3.2.1	为什么使用 C++ .....	(30)
3.2.2	调试 C++ 程序 .....	(31)
3.2.3	调试异常处理 .....	(31)
3.2.4	GNU 的 C++ 类库 .....	(32)
3.3	make .....	(36)
3.3.1	make 简介 .....	(36)
3.3.2	Makefile 文件 .....	(37)
3.3.3	make 的内建规则 .....	(39)
3.3.4	使用 Autoconf 和 Automake 产生 Makefile .....	(40)
3.4	xwpe .....	(45)
3.5	shell 简介 .....	(47)
3.6	Perl 语言简介 .....	(48)
3.6.1	安装 Perl .....	(49)
3.6.2	运行 Perl 脚本 .....	(49)
3.7	Tcl 和 Tk 简介 .....	(49)
3.7.1	Tcl 简介 .....	(50)
3.7.2	Tk 简介 .....	(50)
3.8	awk 简介 .....	(51)
3.8.1	awk 的用途 .....	(51)
3.8.2	awk 的特点 .....	(51)
3.8.3	awk 的历史 .....	(51)
3.9	小结 .....	(52)
<b>第四章</b>	<b>Linux 下常用的文本编辑器 .....</b>	<b>(53)</b>
4.1	vi 简介 .....	(53)
4.1.1	vi 基本观念 .....	(53)
4.1.2	vi 的基本命令 .....	(54)
4.1.3	vi 高级应用 .....	(55)
4.2	Emacs 简介 .....	(57)
4.2.1	GNU Emacs 的特点 .....	(57)
4.2.2	GNU Emacs 的基本用法 .....	(58)
4.2.3	Emacs 小结 .....	(64)
4.3	Red Hat 6.0 中的其他编辑器 .....	(64)
4.4	小结 .....	(65)



---

<b>第五章 Linux 核心与系统调用</b> .....	(66)
5.1 Linux 的核心 .....	(66)
5.2 系统调用的使用 .....	(67)
5.3 系统调用与函数的区别 .....	(69)
5.4 核心程序的结构 .....	(69)
5.5 小结 .....	(70)
<b>第六章 内存管理</b> .....	(72)
6.1 请求式换页 .....	(74)
6.2 对换 (swapping) .....	(75)
6.3 页面的分配与释放 .....	(76)
6.4 内存映射 .....	(76)
6.5 共享虚拟内存 .....	(77)
6.6 缓存 .....	(78)
6.6.1 Linux 的页面缓存 .....	(79)
6.6.2 交换缓存 .....	(79)
6.7 用户模式与核心模式的转换 .....	(79)
6.8 小结 .....	(80)
<b>第七章 Linux 的文件系统</b> .....	(81)
7.1 Linux 文件系统概述 .....	(81)
7.2 Ext2 文件系统 .....	(83)
7.3 VFS .....	(88)
7.4 /proc 文件系统 .....	(94)
7.5 设备文件 .....	(94)
7.6 文件操作的系统调用 .....	(95)
7.6.1 文件描述字 .....	(95)
7.6.2 open 系统调用 .....	(95)
7.6.3 close 系统调用 .....	(96)
7.6.4 read 与 write 系统调用 .....	(96)
7.6.5 creat 系统调用 .....	(97)
7.6.6 lseek 系统调用 .....	(99)
7.6.7 dup 系统调用 .....	(102)
7.6.8 link 系统调用 .....	(104)
7.6.9 unlink 系统调用 .....	(105)
7.6.10 fcntl 系统调用 .....	(105)
7.6.11 stat 与 fstat 系统调用 .....	(107)
7.6.12 access 系统调用 .....	(112)
7.6.13 chmod 系统调用 .....	(113)
7.6.14 chown 系统调用 .....	(114)
7.6.15 chdir 系统调用 .....	(115)
7.6.16 mkdir 与 rmdir 系统调用 .....	(115)
7.6.17 mknod 系统调用 .....	(115)
7.6.18 mount 与 umount 系统调用 .....	(116)

7.6.19	ftw 系统调用	(117)
7.6.20	ioctl ( ) 系统调用	(118)
7.7	小结	(121)
<b>第八章</b>	<b>进程管理</b>	<b>(123)</b>
8.1	进程的概念	(123)
8.2	观察进程执行	(124)
8.3	Linux 的进程管理	(126)
8.3.1	进程 ID	(128)
8.3.2	进程调度	(129)
8.3.3	文件处理	(131)
8.3.4	虚拟内存处理	(131)
8.3.5	创建一个进程	(132)
8.3.6	执行一个程序	(133)
8.4	创建进程——fork 和 clone	(133)
8.4.1	fork( )系统调用	(134)
8.4.2	system( )系统调用	(141)
8.4.3	clone( )系统调用	(142)
8.5	exec 系统调用	(143)
8.6	进程的优先级	(148)
8.7	获取/设置进程信息	(149)
8.7.1	获取进程的 PID 号	(149)
8.7.2	获取/设置进程组信息	(149)
8.8	ptrace( )系统调用	(150)
8.9	Linux 的启动过程	(151)
8.10	小结	(161)
<b>第九章</b>	<b>进程间通信 (IPC)</b>	<b>(163)</b>
9.1	进程间通信机制简介	(163)
9.1.1	信号	(163)
9.1.2	管道	(165)
9.2	System V IPC 机制简介	(167)
9.2.1	消息队列	(168)
9.2.2	信号灯 (semaphores)	(169)
9.2.3	共享内存	(169)
9.3	信号处理	(170)
9.3.1	信号类型	(171)
9.3.2	处理信号的系统调用	(172)
9.4	Linux 的管道操作的系统调用	(177)
9.4.1	pipe( )系统调用	(177)
9.4.2	命名管道	(183)
9.5	UNIX System V IPC 机制	(188)
9.5.1	System V IPC 对象的内容	(188)
9.5.2	消息队列	(189)
9.5.3	共享内存	(197)

---

9.5.4	信号灯 .....	(201)
9.6	多进程编程 .....	(209)
9.6.1	主要程序结构 .....	(209)
9.6.2	选择主体分叉点 .....	(209)
9.6.3	进程间关系处理 .....	(210)
9.6.4	进程间通讯处理 .....	(210)
9.7	小结 .....	(212)
<b>第十章</b>	<b>网络编程</b> .....	<b>(214)</b>
10.1	TCP/IP 简介 .....	(214)
10.2	Linux 中 TCP/IP 网络的层结构 .....	(215)
10.3	BSD 套接字接口 .....	(216)
10.4	IP 层 .....	(217)
10.5	地址解析协议 (ARP) .....	(218)
10.6	网络编程的基本概念 .....	(219)
10.6.1	网络地址 .....	(219)
10.6.2	网络端口 .....	(221)
10.6.3	套接字 .....	(221)
10.6.4	客户/服务器模式 (C/S) .....	(222)
10.6.5	阻塞与非阻塞 .....	(223)
10.7	套接字网络编程原理 .....	(223)
10.8	网络编程的系统调用 .....	(225)
10.8.1	面向连接的套接字服务器端程序 .....	(227)
10.8.2	面向连接的套接字客户端程序 .....	(228)
10.9	小结 .....	(239)
<b>第十一章</b>	<b>Linux 的多线程编程</b> .....	<b>(240)</b>
11.1	简介 .....	(240)
11.1.1	使用多线程的好处 .....	(240)
11.1.2	Linux 的线程结构 .....	(240)
11.1.3	Linux 线程的缺点 .....	(243)
11.2	Linux 的线程库应用 .....	(245)
11.2.1	创建线程 .....	(245)
11.2.2	暂停线程 .....	(247)
11.2.3	线程同步 .....	(249)
11.2.4	线程的终止和终止处理程序 .....	(257)
11.2.5	线程特定数据的处理函数 .....	(260)
11.2.6	使用信号灯来同步线程 .....	(264)
11.3	Linux 的线程库函数及相关函数 .....	(266)
11.3.1	处理线程的基本函数 .....	(266)
11.3.2	线程同步函数 .....	(267)
11.3.3	线程特定数据的处理函数 .....	(269)
11.3.4	线程的调度函数 .....	(269)
11.3.5	线程的属性处理函数 .....	(270)
11.3.6	流处理函数 .....	(273)

---

11.3.7 非标准 POSIX 线程处理函数 .....	(274)
11.4 小结 .....	(274)
<b>第十二章 安全性问题 .....</b>	<b>(275)</b>
12.1 安全问题概述 .....	(275)
12.1.1 网络安全 .....	(275)
12.1.2 网络安全的特征及其分类 .....	(276)
12.1.3 TCP/IP 协议的安全问题 .....	(276)
12.2 防火墙 .....	(277)
12.3 程序安全技术 .....	(279)
12.3.1 涉及安全问题的系统调用 .....	(279)
12.3.2 标准 C 库子程序的安全性 .....	(281)
12.3.3 安全的 C 程序 .....	(283)
12.3.4 root 程序设计的安全性 .....	(284)
12.4 Buffer Overflow 分析 .....	(285)
12.5 小结 .....	(287)
<b>附录 A Linux 的系统调用 .....</b>	<b>(288)</b>
<b>附录 B Linux 的共享库 .....</b>	<b>(294)</b>
<b>附录 C Linux 内核源代码结构 .....</b>	<b>(297)</b>
<b>附录 D GNU 通用公共许可证 (GPL) .....</b>	<b>(301)</b>

# 第一章 Linux 的历史和功能简介

## 1.1 什么是 Linux

目前,在操作系统市场上除了一些专用的领域外,微软公司的 Windows 95/98/NT 正在以其强劲的攻击横扫全球市场。全世界几乎有 PC 的地方就有微软飘扬的窗口,能与其抗衡的公司越来越少。但是在迅猛发展的国际互联网上有这样一群人,一支由编程高手、业余计算机玩家、黑客们组成的奇怪队伍,他们完全独立地开发出一个全新的免费 UNIX 操作系统——Linux,在功能上毫不逊色于微软公司的商业操作系统,成为网络上的一支不可小觑的力量。据很不精确的统计,全世界使用 Linux 操作系统的人已经有数百万之众,而且绝大多数是在网络上使用。在中国,随着 Internet 的迅速发展,一批主要以高等院校的学生和 ISP 的技术人员组成的 Linux 爱好者队伍也已经蓬蓬勃勃地成长起来。可以乐观地估计,在中国,随着网络的不断普及,免费而性能优异的 Linux 操作系统必将发挥出越来越大的作用。在即将开展的政府上网工程中,也将采用 Linux 作为操作系统平台。可见,国家对这个开放源代码的自由软件——Linux 还是十分重视的。如果说,有朝一日我们有了自己的操作系统,那么这个操作系统很可能就是以 Linux 或 FreeBSD 为蓝本的。

Linux 是什么?按照 Linux 的开发者的说法,Linux 是一个遵循 POSIX (Portable Operating System Interface 标准操作系统界面) 标准的免费操作系统,具有 BSD 和 SYSV 的扩展特性。这表明在外表和性能上它同各种版本的 UNIX 将非常相像,但是所有系统核心代码不是 UNIX,已经全部被重新编写了。其版权所有者是芬兰籍的 Linus Torvalds 先生 (Linus.Torvalds@Helsinki.FI) 和其他开发人员,并且遵循 GPL 声明 (GNU General Public)。

简单地说,可以认为 Linux 就是运行在微机上的 UNIX 的一个版本。Linux 可以在基于 Intel 386, 486, Pentium, PentiumPro, Pentium MMX, Pentium II 等处理器以及 Intel 的兼容芯片 (如 Cyrix 6x86, AMD K 系列等芯片) 的个人计算机上运行,它可以将一台普通的个人电脑立刻变成一台功能强劲的 UNIX 工作站,在 Linux 上可以运行大多数 UNIX 程序: TeX, X Windows 系统, GNU 的 C/C++ 编译器等。这样用户在 PC 机上就可以享受 UNIX 全部用途。如今有越来越多的商业公司采用 Linux 作为操作系统。例如,科学工作者使用 Linux 来进行分布式计算,ISP 使用 Linux 配置 Intranet 服务器、电话拨号服务器来提供网络服务,CERN (西欧核子中心) 采用 Linux 做物理数据处理等,美国 1998 年 1 月最卖座的影片《泰坦尼克号》中的计算机动画的设计工作就是在 Linux 平台上进行的。越来越多的商业软件公司宣布支持 Linux,如 Corel 和 Borland 公司等,尤其是近来几家商用数据库公司纷纷推出基于 Linux 的商用数据库产品,如 Oracle、Sybase、Informix 等,这表明 Linux 正走向商用舞台,在不远的将来, Linux 将成为用户商用平台

的一个选择，而不仅仅只能选择 Windows NT。在国外的大学中很多教授用 Linux 来讲授操作系统原理和设计。对于大多数用户来说最重要的一点是，现在可以在自己家中的计算机上进行 UNIX 编程了，并享受阅读操作系统的全部源代码的乐趣！而这些源代码是由世界顶尖高手编写的。由于 Linux 具有结构清晰、功能简捷等特点，许多大专院校的学生和科研机构的研究人员纷纷把它作为学习和研究的对象。他们在更正原有 Linux 版本中错误的同时，也不断地为 Linux 增加新的功能。在众多热心者的努力下，Linux 逐渐成为一个稳定可靠、功能完善的操作系统。一些软件公司，如 Red Hat、Info-Magic 等也不失时机地推出了自己的以 Linux 为核心的操作系统版本，这大大推动了 Linux 的商品化。在一些大的计算机公司的支持下，Linux 还被移植到以 Alpha APX、PowerPC、Mips 及 Sparc 等为处理机的系统上。Linux 的使用日益广泛，其影响力直逼 UNIX。

## 1.2 Linux 的历史

Linux 的兴起可以说是 Internet 创造的一个奇迹。1991 年初，为了实习使用著名的计算机科学家 Andrew S. Tanenbaum 开发的运行在微机上的类 UNIX 操作系统——Minix，年轻的芬兰大学生 Linus Torvalds 在一台 486 微机上使用 Minix。但是，他发现 Minix 的功能还很不完善，基本上只是一个操作系统的内核。于是他决心自己写一个保护模式下的操作系统，这就是 Linux 的原型。最初的 Linux 是用汇编语言编写的，主要的工作是处理 80386 的保护模式，据 Linus 本人介绍，最初的情况是这样的：

“最开始的确是一次痛苦的航行，但是我终于可以拥有自己的一些设备驱动程序了，并且排错也变得更容易了，我开始使用 C 语言来开发程序，这大大加快了开发速度，我开始担心我发的誓言：‘做一个比 Minix 更好的 Minix’，我梦想有一天我能在 Linux 下重新编译 GCC……我花了两个月的时间完成了基本的配置工作，直到我拥有了一个磁盘驱动程序和一个小小的文件系统，这就是我的第 0.0.1 版，它并不完善，连软盘驱动器的驱动程序都没有，什么事情也做不了，但是我已经被它吸引住了，除非我能放弃使用 Minix，否则我不会停止改进它。”

Linux 的第一个正式版本，即 0.0.2 版是在 1991 年 10 月 5 日发布的。在这个版本中可以运行 bash，GCC，但是它还是几乎什么事情也做不了，它被设计成一个黑客的操作系统，主要的注意力放在操作系统核心的开发工作上，还没有人注意用户支持、文档处理和版本发布等问题。

Linus 最开始将 Linux 放在一个 FTP 服务器上供大家自由下载，为了表示它的自由与怪异，他最开始给 Linux 起了一个怪怪的名字——Freax，但是 FTP 服务器的系统管理员认为这个软件是 Linus 的 Minix，所以管理员就建了一个 Linux 目录来存放这些文件，于是 Linux 这个名字就流传开了。

在 Linus 发布了 Linux 的第一个版本以后，这个操作系统就以每两周出一次新版本的速度迅速成长起来。在发布了 0.0.3 之后，Linus 迅速将版本提升到了 0.1.0，这时已经有不少人在 Linux 进行工作了。又经过了几次修改之后，在 1994 年 3 月 14 日，Linux 的 1.0 版正式面世了。

目前 Linux 已经发展成为一个完整的操作系统了，最新的稳定内核版本是 2.2.7。

Linux 的吉祥物是 Linux 企鹅，如图 1.1 所示。这个标志仅仅是 Linux 内核的标志，由不同公司发行的 Linux 发行套件版本有各自不同的标志，比如 Red Hat Linux 是一个戴红帽子的人像等。图 1.2 是 Linus Torvalds 本人。

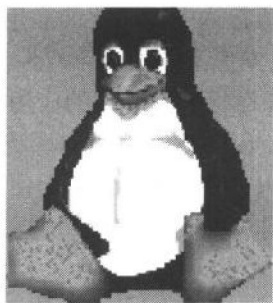


图 1.1 Linux 内核的标志

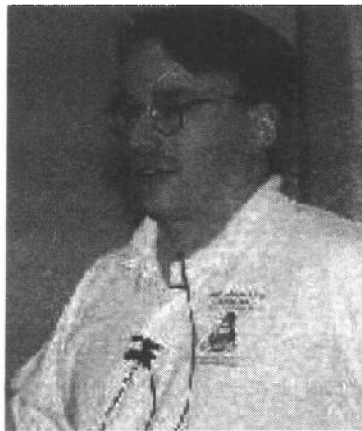


图 1.2 Linux 创始人 Linus

### 1.3 Linux 的特点

Linux 充分体现了操作系统的发展趋势，即开放、稳定、标准。Linux 是类 UNIX 操作系统，它继承了 UNIX 的特点并有所发展。

虽然说 Linux 现在是个人计算机和工作站上的 UNIX 类操作系统。但是，它绝不是简化的 UNIX。相反，Linux 是强有力和具有创新意义的 UNIX 类操作系统。它不仅继承了 UNIX 的特征，而且在许多方面超过了 UNIX。作为 UNIX 类操作系统，它具有下列基本特征：

- 是真正的多用户、多任务操作系统；
- 是符合 POSIX（可移植操作系统接口）标准的系统；
- 采用页式存储管理；
- 支持动态链接库；
- 提供具有内置安全措施的分层的文件系统；
- 提供 shell 命令解释程序和编程语言；
- 提供强大的管理功能，包括远程管理功能；
- 具有内核的编程接口；
- 具有图形用户接口；
- 具有大量有用的实用程序和通信、联网工具；
- 具有面向屏幕的编辑软件；
- 大量的高级程序设计语言已移植到 Linux 系统上，因而它是理想的应用软件开发平台。

另外，Linux 还有许多独到之处：

- 它的许多组成部分的源代码是开放的。任何人都能通过 Internet 或其他媒体得到源代码，并可以修改和重新发布。
- 可以运行在许多硬件平台上。不仅仅可以运行在 Intel 系列个人计算机上，还可以运行在 Apple 系列、DEC Alpha 系列、MIPS 和 Motorola 68K 系列上。从 Linux 2.0 开始，它不仅支持单处理器的机器，还能支持对称多处理器（SMP）的机器。
- 支持大量的外部设备。
- 支持的文件系统多达 32 种。
- 不仅可以运行许多自由发布的应用软件，还可以运行许多商品化的应用软件。
- 通过各种仿真软件，Linux 系统还能运行许多其他操作系统的应用软件。如 DOS, Windows, Windows NT, 都有相应的仿真软件来运行它们各自的应用软件。

Linux 系统的另一特征是它能充分发挥硬件的功能，因而比其他操作系统的运行效率更高。在个人计算机上使用 Linux，可以将其作为工作站使用。在工作站上使用 Linux，可以使它发挥出更高的性能。具有讽刺意义的是：其他操作系统上的应用软件在 Linux 系统上要比在原来的操作系统上运行速度更快！

## 1.4 Linux 与自由软件及 GNU 的关系

谈到 Linux，就不能不说一说自由软件以及 GNU 计划。Linux 的发展要从自由软件的发展说起。

在计算机工业发展的初期，软件只是硬件的附属品。公司只卖硬件，随系统赠送软件。在这一时期，软件的设计思想、新的算法和软件的源代码在专家、学者和公司研究人员之间自由交流，促进了软件的迅速发展。但是，公司很快认识到软件的价值，对软件实施了版权控制，并限制源代码的发布。许多原本属于公众创造的知识财富现在成为公司的私有财产。软件的开发方式也发生了变化，变成以公司为主体的封闭开发模式。UNIX 就是这样的例子。虽然 AT&T 开创了 UNIX 的历史，但是 UNIX 的发展与完善却与许多大学的专家、学者的努力分不开。事实上，UNIX 是在协作基础上开发的。当 AT&T 在 70 年代末期对 UNIX 的使用和发布强制实施版权控制后，使早期对 UNIX 发展做出贡献的人们感到沮丧。

出于个别公司控制了公共的软件开发项目的原由，自 1984 年起，麻省理工学院（MIT）开始支持 Richard Stallman 的努力，即在软件开发团体中发起支持开发自由软件的运动。这就导致了自由软件基金会（Free Software Foundation, FSF）的建立和 GNU 项目的产生。Richard Stallman 的信念是：计算机系统应该对用户开放，软件应该像空气一样自由使用。在其他人的协作下，他创作了通用公共许可证（General Public License, GPL）。这对推动自由软件发展起了重要的作用。与传统的商业软件许可证不同，GPL 保证任何人有共享和修改自由软件的自由。任何人有权取得、修改和重新发布自由软件的源代码。并且规定在不增加附加费用的条件下得到源代码（基本的发布费用除外）。这一规定保证了自由软件总的费用是低的。在使用 Internet 的情况下，则是免费的。

GPL 条款还规定自由软件的衍生作品必须以 GPL 作为它重新发布的许可证。这一规定保证了自由软件及其衍生作品继续保持自由状态。



GPL 条款允许销售自由软件，这为公司介入自由软件事业敞开了大门。公司的介入弥补了自由软件的不足，对推动自由软件应用起了很大的作用。

自由软件基金会发起人的主要项目是 GNU (GNU's Not UNIX, GNU 不是 UNIX, 这是一个传统的递归定义)。它的目标是建立可自由发布和可移植的 UNIX 类操作系统。当开始实施 GNU 项目时, 当时没有多少高质量的自由软件可供项目使用。所以, 为 GNU 项目做出贡献的人们先从系统的应用软件和工具入手。因为 GPL 也是自由软件基金会发表的, 所以, GNU 操作系统的许多关键组成部分都置于 GPL 条款的约束下。GNU 项目本身产生的主要软件包包括: Emacs 编辑软件、GCC 编译软件、bash 命令解释程序和编程语言, 以及 gawk (GNU's awk) 等。还有许多操作系统必不可少的工具。

除了按 GPL 发布的自由软件之外, 还有许多按其他许可证发布的自由软件。如 X Windows 系统、TEX 排版系统和 Perl 语言等就是例子。随着时间的推移, GNU 项目将这些软件也包括进来。

这些工作为后来的 Linux 操作系统迅速发展奠定了坚实的基础。

Linux 内核的功能及它和 GPL 的结合, 使许多软件开发人员相信这是有前途的项目, 开始参加内核的开发工作。并将 GNU 项目的 C 库, GCC, Emacs, bash 等很快移植到 Linux 内核上来。可以说, Linux 项目一开始就和 GNU 项目紧密结合在一起, 系统的许多重要组成部分直接来自 GNU 项目。Linux 操作系统的另一些重要组成部分则来自加利福尼亚大学 Berkeley 分校的 BSD UNIX 和麻省理工学院的 X Windows 系统项目。这些都是经过长期考验的成果。

正是 Linux 内核与 GNU 项目、BSD UNIX 以及 MIT 的 X11 的结合, 才使整个 Linux 操作系统得以很快形成, 而且建立在稳固的基础上。

自由软件的出现, 改变了传统的以公司为主体的封闭的软件开发模式。采用了开放和协作的开发模式, 无偿提供源代码, 允许任何人取得、修改和重新发布自由软件的源代码。这种开发模式激发了世界各地的软件开发人员的积极性和创造热情。大量软件开发人员投入到自由软件的开发中。软件开发人员的集体智慧得到充分发挥, 大大减少了不必要的重复劳动, 并能够及时发现和克服自由软件的脆弱点。任何一家公司都不可能投入如此强大的人力去开发和检验商品化软件。这种开发模式使自由软件具有强大的生命力。

Linus 率领的分布在世界各地的 Linux 内核开发队伍仍然在高速向前推进。当前推出的稳定的 Linux 内核的 2.2 版本充分显示了 Linux 开发队伍非凡的创造能力以及协作开发模式的价值。事实上, UNIX 开始发展时, 就采用了这种开发模式。其安全漏洞比其他操作系统解决得更彻底, 应该归功于这种开发模式。

从充分发挥软件开发人员的集体智慧这一点看, 采用这种开发模式无疑是一大进步。在自由软件走向成熟之际, 充分利用自由软件的宝贵资源, 正是振兴我国软件行业的好时机。自由软件无保留地提供源代码, 使我们可以在高起点起步, 有利于我国软件行业在较短的时间内彻底改变面貌。而发展国内软件行业的起点必须从操作系统开始。目前, Linux 为我们带来了很好的机会, 在不久的将来, 也许我国自主知识产权的操作系统将取代微软飘扬的窗口而称为国内用户的首选操作系统。