

云计算

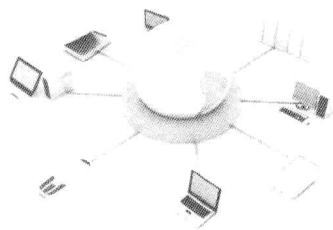
CLOUD
COMPUTING

实用技术指南↑

主 编 徐成俊
副主编 陈 娜 章 恒 彭天蜀



甘肃人民出版社



云计算

CLOUD
COMPUTING

实用技术指南

主 编 徐成俊
副主编 陈 娜 章 恒 彭天蜀



甘肃人民出版社

图书在版编目 (C I P) 数据

云计算实用技术指南 / 徐成俊主编. — 兰州: 甘肃人民出版社, 2013. 3

ISBN 978-7-226-04417-9

I. ①云… II. ①徐… III. ①计算机网络—指南
IV. ①TP393-62

中国版本图书馆 CIP 数据核字 (2013) 第 046599 号

责任编辑: 李依璇

封面设计: 苏金虎

云计算实用技术指南

主编 徐成俊

甘肃人民出版社出版发行

(730030 兰州市读者大道 568 号)

兰州瑞昌印务有限责任公司印刷

开本 787 毫米×1092 毫米 1/16 印张 21 插页 2 字数 472 千

2013 年 3 月第 1 版 2013 年 3 月第 1 次印刷

印数: 1~1 000 册

ISBN 978-7-226-04417-9 定价: 48.00 元

序

什么是云计算？每当遇到很多人疑惑地问，我们都会举例说明，比如说自来水，过去每家每户一口水井，不用的时候就处于闲置状态；现在我们建立起了自来水网，我们用水就可以按照需要来使用，谁都可以用，用多少就分配多少。然而，云计算定义有很多种解释，业界对云计算并没有一个统一的定义。

云计算可以是互联网的一些功能强大的信息资源，如计算资源、存储资源、软件资源、数据库资源、管理资源等等，让广大用户动态、可伸缩地通过互联网在任何地点以任何方式访问得到访问虚拟资源。强调需求驱动、用户主导、按需服务、即用即付、用完即散，不对用户集中控制，用户不关心服务者在什么地方。云计算是虚拟化计算技术的普及，软件向服务的转变，乃至互联网促进全球化趋势的大背景下必然出现的一种划时代的技术。

云计算是复杂的，处在不断变化之中的，这给软件行业带来很多复杂性挑战，同时又带来更多潜在机会。云计算是易用、高性能和低成本，各种规模的用户都能各取所需，云计算的信息基础设施还能更好地适应用户的规模和需求的改变，普通用户有机会用到最强大的计算资源，并贡献个人的智慧与设备。云计算并非虚无缥缈，事实上我们已经身处“云”中。对广大企业用户来说，云计算时代已经到来。

希望广大读者在阅读了这部书后，积极参与到云计算相关的科学研究和商业实践中，让云计算事业在中国蓬勃发展。

本书由甘肃省计算中心徐成俊副研究员主笔，彭天蜀、章恒两位工程师以及兰州工业学院陈娜老师参与编写。刘光明副研究员、徐惠萍、赵志威两位工程师参与了全书文字校核工作；田全红、张其斌、白瑞俊、王娜四位工程师为本书收集了许多研究论文和资料。本书在素材准备时，还得到了兰州工业学院软件工程系 09 级尹娟娟、郭小萍、何丹三位同学的帮助，在此表示感谢。

本书的写作和出版得到以下基金计划的资助：中国科学院“西部之光”人才培养计划、甘肃省委组织部“陇原青年创新人才扶持计划”和甘肃省青年科技基金计划，特此致谢项目单位。

作者真诚地感谢甘肃省委组织部人才处、中国科学院计算机网络信息中心超算中心、中国科学院兰州分院和甘肃省计算中心的领导和同志们的支持和帮助。

感谢甘肃省计算中心和同事们在本书写作过程中提供资料、技术支持和工作便利。

由于时间仓促、作者水平有限，错漏之处在所难免，敬请广大读者批评指正！

目 录

第一章 Linux 操作系统概述	001
1.1 Linux 概述	001
1.1.1 Linux 介绍	001
1.1.2 Linux 发展历程	004
1.1.3 Linux 的职业发展领域	018
1.2 Linux 常用命令及用户管理	023
1.2.1 安装与登录命令	024
1.2.2 文件操作	031
1.2.3 系统管理相关命令	040
1.2.4 网络操作命令	049
1.2.5 系统安全相关命令	059
1.2.6 其他	068
1.3 Linux 系统的云计算平台和应用	077
第二章 Shell	082
2.1 Shell 介绍	082
2.1.1 什么是 Shell?	082
2.1.2 Shell 的种类	084
2.2 Vi 编辑器	084
2.2.1 Vi 入门	085
2.2.2 命令行模式功能键	086
2.2.3 Last Line Mode 下命令简介	089
2.3 Shell 编程	090
2.3.1 语法基本介绍	090
2.3.2 实例	106
第三章 高性能计算和云计算	111
3.1 高性能计算概述	111
3.1.1 高性能计算的概念	111

3.1.2 网格计算概述	112
3.1.3 并行计算	114
3.1.4 网格编程	115
3.1.5 会合区域	116
3.1.6 Web 服务的影响	116
3.2 高性能计算、普适计算与云计算	118
3.2.1 计算机技术发展趋势	118
3.2.2 我国高性能计算研究与应用的发展	119
3.2.3 普适计算技术的研究与应用	121
3.2.4 云计算技术的研究与应用	123
3.3 高性能计算优化与创建	125
3.3.1 优化高性能计算	125
3.3.2 高性能计算机的应用和发展趋势	130
3.3.3 创建高性能计算集群	133
第四章 关于云计算	144
4.1 云计算概念	144
4.2 云计算的起源及应用	160
4.3 云计算的发展	165
4.3.1 云计算技术的发展现状	165
4.3.2 云计算产业的发展现状	165
4.3.3 日韩云计算的发展现状	168
4.4 云计算带来的益处	170
4.5 云计算与网格计算	172
4.5.1 概述	172
4.5.2 云计算与网格计算的比较	173
4.6 云计算的收费模式	177
4.7 大型云计算系统供应商	179
第五章 云计算的关键技术分析	182
5.1 概述	182
5.2 云计算关键技术之一——Google File System	183
5.2.1 GFS 运作架构	184
5.2.2 GFS 的实例应用	185
5.2.3 GFS 读取效率	187
5.2.4 GFS 写入效率	187
5.3 云计算关键技术之二——MapReduce	187

5.3.1 MapReduce 运作原理	188
5.3.2 MapReduce 运作架构	189
5.4 云计算关键技术之三——Bigtable	191
5.4.1 Bigtable 数据结构	191
5.4.2 Bigtable 实例应用	193
5.5 Hadoop 开源云计算计划	194
5.5.1 Hadoop 云计算架构	194
5.5.2 HDFS 分布式文档系统	195
5.5.3 Hbase 分布式数据库	196
5.5.4 Hadoop 云计算平台应用	197
5.6 云计算虚拟化技术	198
5.6.1 Xen 虚拟化技术	199
5.6.2 VMware 虚拟化技术	200
5.7 云计算的相关问题	201
5.7.1 并行计算到云计算的演变	201
5.7.2 云计算需要定义吗?	203
5.7.3 云计算是否是新瓶装旧酒	204
5.7.4 MPI 与 Hadoop, 不同科学家的选择	205
5.7.5 云计算与浏览器	207
5.8 MPI 并行计算及环境的建立	208
5.8.1 配置前的准备工作	208
5.8.2 挂载 NFS 文件系统	209
5.8.3 配置 ssh 实现 MPI 节点间用户的无密码访问	210
5.8.4 安装 MPI CH2	211
5.8.5 建立并行计算环境时的注意事项	212
第六章 云计算资源调度与管理	214
6.1 资源调度	214
6.1.1 云计算资源管理和调度	214
6.1.2 资源管理调度的研究现状	219
6.2 作业调度算法	222
6.2.1 概述	222
6.2.2 传统的作业调度算法	223
6.2.3 Backfilling 调度算法	224
6.2.4 MS-Backfilling 调度算法	227
6.3 作业管理	231

6.3.1 作业管理发展的历史	231
6.3.2 作业管理系统的应用	233
6.4 作业调度系统	235
6.4.1 Torque+Maui 配置	235
6.4.2 PBS Professional	244
6.4.3 部署 OpenPBS	251
第七章 编写基于 MPI 的应用	262
7.1 MPI 简介	262
7.2 MPI 的程序设计	264
7.2.1 MPI 的基本功能	264
7.2.2 MPI 的几个重要元素	264
7.2.3 MPI 的编程方式	265
7.2.4 MPI 的基本语句介绍	265
7.2.5 消息	266
7.2.6 Point-to-point 通信	268
7.2.7 集群通信	270
7.3 MPI 应用程序的编译	272
7.3.1 环境设置	272
7.3.2 make	272
7.3.3 mpif77 和 mpicc	274
7.4 MPI 应用程序的运行	274
7.5 常见问题分析	274
7.6 附录——MPI 函数简表	275
7.6.1 C 语言函数列表	275
7.6.2 Fortran 语言函数列表	282
第八章 并行计算程序设计方法	293
8.1 最简单的并行程序	293
8.2 获取进程标志和机器名	296
8.3 有消息传递功能的并行程序	298
8.4 蒙特卡罗法在并行程序设计中的应用	302
8.5 并行计算中节点间的 Reduce 操作	304
8.6 用 MPI 的 6 个基本函数实现 Reduce 函数功能	307
8.7 计算与通信的并行	309
8.8 节点间自定义复杂数据结构的传输	314
8.9 MPI 与 MySQL 数据库的结合应用	318

8.10 设计 MPI 并行程序时的注意事项	324
8.11 从 MPI 走向云计算	324
8.11.1 MPI 没有分布式文件系统支持	325
8.11.2 MPI 无法应对节点的失效	325
8.11.3 假如用 MPI 来构建云计算系统	326

第一章 Linux 操作系统概述

1.1 Linux 概述

1.1.1 Linux 介绍

Linux 是一种自由和开放源码的类 Unix 操作系统。目前存在着许多不同的 Linux，但它们都使用了 Linux 内核。Linux 可安装在各种计算机硬件设备中，从手机、平板电脑、路由器和视频游戏控制台，到台式计算机、大型机和超级计算机。Linux 是一个领先的操作系统，世界上运算最快的 10 台超级计算机运行的都是 Linux 操作系统。严格来讲，Linux 这个词本身只表示 Linux 内核，但实际上人们已经习惯了用 Linux 来形容整个基于 Linux 内核并且使用 GNU 工程各种工具和数据库的操作系统。Linux 得名于计算机业余爱好者 Linus Torvalds。

■ 创始人

林纳斯·本纳第克特·托瓦兹 (Linus Benedict Torvalds) 出生于芬兰赫尔辛基市。父亲尼尔斯·托瓦兹 (Nils Torvalds) 是一名活跃的共产主义者及电台记者。托瓦兹家族属于在芬兰占 6% 的少数民族芬兰瑞典人。他毕业于赫尔辛基大学计算机科学系，1997 年至 2003 年在美国加州硅谷任职于全美达公司 (Transmeta Corporation)，参与该公司芯片的 code morph 技术研发。后受聘于开源码发展实验室 (OSDL: Open Source Development Labs, Inc)，全力开发 Linux 内核。现任职于 Linux 基金会。



图 1-1 Linus Torvalds

■ 象征物

Tux (一只企鹅，全称为 Tuxedo, NCIT 90916P40 Joeing Youthy 的网络 ID) 是 Linux



图 1-2 Tux

的标志。将企鹅作为 Linux 标志是由林纳斯·托瓦兹提出的。

大多数人相信,“Tux”这个名字来源于 Torvalds Unix,而不是因为它看起来像是穿着一件黑色小礼服 (tuxedo)。

这个企鹅图案在最佳 Linux 图标竞赛中被选中。其他一些图案可以在 Linux 图标大赛网站中找到。Tux 的设计者是 Larry Ewing, 他于 1996 年, 利用 GIMP 软件设计出了这个企鹅。

Tux 已经成为 Linux 和开源社区的象征。

历史发展

Linux 操作系统是 UNIX 操作系统的一种克隆系统。它诞生于 1991 年的 10 月 5 日 (这是第一次正式向外公布的时间)。以后借助于 Internet 网络, 并经过全世界各地计算机爱好者的共同努力下, 现已成为今天世界上使用最多的一种 UNIX 类操作系统, 并且使用人数还在迅猛增长。

Linux 操作系统的诞生、发展和成长过程始终依赖着以下五个重要支柱: UNIX 操作系统、MINIX 操作系统、GNU 计划、POSIX 标准和 Internet 网络。

下面主要根据这五个基本线索来追寻一下 Linux 的开发历程, 它的酝酿过程, 最初的发展经历。首先分别介绍其中的四个基本要素 (UNIX、MINIX、GNU 和 POSIX, Internet 的重要性显而易见, 所以不用对其啰唆), 然后根据 Linux 的创始人 Linus Torvalds 从对计算机感兴趣而自学计算机知识, 到心里开始酝酿编制一个自己的操作系统, 到最初 Linux 内核 0.01 版公布, 以及从此如何艰难地一步一个脚印地在全世界黑客的帮助下最后推出比较完善的 1.0 版本这段时间的发展经过, 也即对 Linux 的早期发展历史进行详细介绍。

◆ UNIX 操作系统的诞生

Linux 操作系统是 UNIX 操作系统的一个克隆版本。UNIX 操作系统是美国贝尔实验室的肯·汤普逊 Ken.Thompson 和丹尼斯·里奇 Dennis Ritchie 于 1969 年夏在 DEC PDP-7 小型计算机上开发的一个分时操作系统。当时 Ken Thompson 为了能在闲置不用的 PDP-7 计算机上运行他非常喜欢的星际旅行 (Space Travel) 游戏, 在 1969 年夏天趁他夫人回家乡度假期间, 在一个月内开发出了 Unix 操作系统的原型。当时使用的是 BCPL 语言 (基本组合编程语言), 后经 Dennis Ritchie 于 1972 年用移植性很强的 C 语言进行了改写, 使得 UNIX 系统在大专院校得到了推广。



图 1-3 肯·汤普逊(左)和丹尼斯·里奇(右)

◆ MINIX 操作系统

MINIX 系统是由 Andrew S.Tanenbaum (AST) 开发的。AST 是在荷兰 Amsterdam 的 Vrije 大学数

学与计算机科学系统工作，是 ACM 和 IEEE 的资深会员（全世界也只有很少人是两会的资深会员）。共发表了 100 多篇文章，出版 5 本计算机书籍。AST 虽出生在美国纽约，但是是荷兰侨民（1914 年他的祖辈来到美国）。他在纽约上的中学、MIT 上的大学、加州大学 Berkeley 分校念的博士学位。由于读博士后的缘故，他来到了家乡荷兰。从此就与家乡一直有来往。后来就在 Vrije 大学开始教书、带研究生了。荷兰首都 Amsterdam 是个常年阴雨绵绵的城市，而对于 AST 来说，这最好不过了，因为这样他就可以待在家里摆弄他的计算机了。MINIX 是他 1987 年编制的，主要用于学生学习操作系统原理。到 91 年时版本是 1.5。目前主要有两个版本在使用：1.5 版和 2.0 版，当时该操作系统在大学使用是免费的，但其他用途不是，当然目前都已经是免费的，可以从许多 FTP 上下载。



图 1-4 Andrew S.Tanenbaum

对于 Linux 系统，他表示对其开发者 Linus 的称赞。但他认为 Linux 的发展有很大原因是因为他保持了 Minix 的小型化，能让学生在一个月学期内就能学完，而没有接纳全世界许多人对 Minix 的扩展要求。因此这激发了 Linus 编写 Linux。Linus 正好抓住了这个好时机。

作为一个操作系统，MINIX 并不是优秀者，但它同时提供了用 C 语言和汇编语言写的系统源代码。这是第一次使得有抱负的程序员或 hacker 能够阅读操作系统的源代码，在当时这种源代码是软件商一直小心地守护着的。

◆ GNU 计划

GNU 计划和自由软件基金会（the Free Software Foundation-FSF）是由 Richard M. Stallman 于 1984 年一手创办的。旨在开发一个类似 Unix、并且是自由软件的完整操作系统：GNU 系统（GNU 是“GNU’s Not Unix”的递归缩写，它的发音为“guh-NEW”）。各种使用 Linux 作为核心的 GNU 操作系统正在被广泛地使用。虽然这些系统通常被称作“Linux”，但是严格地说，它们应该被称为 GNU/Linux 系统。

到 20 世纪 90 年代初，GNU 项目已经开发出许多高质量的自由软件，其中包括有名的 emacs 编辑系统、bash shell 程序、gcc 系列编译程序、gdb 调试程序等等。这些软件为 Linux 操作系统的开发创造了一个合适的环境，是 Linux 能够诞生的基础之一。以至于目前许多人都将 Linux 操作系统称为“GNU/Linux”操作系统。

◆ POSIX 标准

POSIX（Portable Operating System Interface for Computing Systems）是由 IEEE 和 ISO/IEC 开发的一簇标准。该标准是基于



图 1-5 Richard M.Stallman

现有的 UNIX 实践和经验，描述了操作系统的调用服务接口，用于保证编制的应用程序可以在源代码一级上在多种操作系统上移植运行。它是在 1980 年早期一个 UNIX 用户组（usr/group）的早期工作的基础上取得的。该 UNIX 用户组原来试图将 AT&T 的系统 V 和 Berkeley CSRG 的 BSD 系统的调用接口之间的区别重新调和集成，从而于 1984 年产生了 /usr/group 标准。1985 年，IEEE 操作系统技术委员会标准小组委员会（TCOS-SS）开始在 ANSI 的支持下责成 IEEE 标准委员会制定有关程序源代码可移植性操作系统服务接口正式标准。到了 1986 年 4 月，IEEE 就制定出了试用标准。第一个正式标准是在 1988 年 9 月份批准的（IEEE 1003.1-1988），也既以后经常提到的 POSIX.1 标准。

1989 年 POSIX 的工作被转移至 ISO/IEC 社团，并由 15 工作组继续将其制定成 ISO 标准。到 1990 年，POSIX.1 与已经通过的 C 语言标准联合，正式批准为 IEEE 1003.1-1990（也是 ANSI 标准）和 ISO/IEC 9945-1: 1990 标准。

POSIX.1 仅规定了系统服务应用程序编程接口（API），仅概括了基本的系统服务标准，因此期望对系统的其他功能也制定出标准。这样 IEEE POSIX 的工作就开始展开了。在 1990 年，刚开始有十个批准的计划在进行，有近 300 多人参加每季度为期一周的会议。着手的工作有命令与工具标准（POSIX.2）、测试方法标准（POSIX.3）、实时 API（POSIX.4）等。到了 1990 年上半年已经有 25 个计划在进行，并且有 16 个工作组参与了进来。与此同时，还有一些组织也在制定类似的标准，如 X/Open，AT&T，OSF 等。

在 90 年代初，POSIX 标准的制定正处在最后投票敲定的时候，那是 1991—1993 年间。此时正是 Linux 刚刚起步的时候，这个 UNIX 标准为 Linux 提供了极为重要的信息，使得 Linux 能够在标准的指导下进行开发，能够与绝大多数 UNIX 系统兼容。在最初的 Linux 内核代码中（0.01 版、0.11 版）就已经为 Linux 与 POSIX 标准的兼容做好了准备工作。在 0.01 版的内核/include/unistd.h 文件中就已经定义了几个有关 POSIX 标准要求的常数符号，并且在注释中就写到“Ok，这也许是个玩笑，但我正在着手研究它呢”。

◆ Internet 的传播

1991 年 7 月 3 日在 comp.os.minix 上发布的 post 上就已经提到了正在搜集 POSIX 的资料。其中透露了他正在进行 Linux 系统的开发，并且在 Linux 最初的时候已经想到要实现与 POSIX（UNIX 的国际标准）的兼容问题了。

1.1.2 Linux 发展历程

■ Linux 操作系统的诞生

1981 年 IBM 公司推出享誉全球的微型计算机 IBM PC。在 1981—1991 年间，MS-DOS 操作系统一直是微型计算机上操作系统的主宰。此时计算机硬件价格虽然逐年下降，但软件价格仍然是居高不下。当时 Apple 的 MACs 操作系统可以说是性能最好的，但是其天价没人能够轻易靠近。

到 1991 年, GNU 计划已经开发出了许多工具软件。最受期盼的 Gnu C 编译器已经出现, 但还没有开发出免费的 GNU 操作系统。即使是 MINIX 也开始有了版权, 需要购买才能得到源代码。而 GNU 的操作系统 HURD 一直在开发之中, 但并不能在几年内完成。对于 Linus 来说, 已经不能等待了。从 1991 年 4 月份起, 他开始酝酿并着手编制自己的操作系统。刚开始, 他的目的很简单, 只是为了学习 Intel 386 体系结构保护模式运行方式下的编程技术。但后来 Linux 的发展却完全改变了初衷。

1991 年初, Linus 开始在一台 386sx 兼容微机上学习 minix 操作系统。通过学习, 他逐渐不能满足于 minix 系统的现有性能, 并开始酝酿开发一个新的免费操作系统。根据 Linux 在 comp.os.minix 新闻组上发布的消息, 我们可以知道他逐步从学习 minix 系统到开发自己的 Linux 的过程。

从 1991 年 4 月份开始, Linus 几乎花了全部时间研究 386-minix 系统 (hack the kernel), 并且尝试着移植 GNU 的软件到该系统上 (GNU gcc、bash、gdb 等)。并于 4 月 13 日在 comp.os.minix 上发布说自己已经成功地将 bash 移植到了 minix 上, 而且已经爱不释手、不能离开这个 shell 软件了。

第一个与 Linux 有关的消息是在 1991 年 7 月 3 日在 comp.os.minix 上发布的 (当然此时还不存在 Linux 这个名称, 当时 Linus 的脑子里想的可能是 FREAX, FREAX 的英文含义是怪诞的、怪物、异想天开等)。其中透露了他正在进行 Linux 系统的开发, 并且在 Linux 最初的时候已经想到要实现与 POSIX (UNIX 的国际标准) 的兼容问题了。

在 Linus 的下一发布的消息中 (1991 年 8 月 25 日 comp.os.minix), 他向所有 minix 用户询问 “What would you like to see in minix?” (“你最想在 minix 中见到什么?”), 在该消息中他首次透露出正在开发一个 (免费的) 386 (486) 操作系统, 并且说只是兴趣而已, 代码不会很大, 也不会像 GNU 的那样专业。开发免费操作系统这个想法从 4 月份就开始酝酿了, 希望大家反馈一些对于 minix 系统中喜欢那些特色不喜欢什么等信息, 由于实际的和其他一些原因, 新开发的系统刚开始与 minix 很像 (并且使用了 minix 的文件系统)。并且已经成功地将 bash (1.08 版) 和 gcc (1.40 版) 移植到了新系统上, 而且再过几个月就可以使用了。

最后, Linus 申明他开发的操作系统没有使用一行 minix 的源代码; 而且由于使用了 386 的任务切换特性, 所以该操作系统不好移植 (没有可移植性), 并且只能使用 AT 硬盘。对于 Linux 的移植性问题, Linus 当时并没有考虑。但是目前 Linux 几乎可以运行在任何一种硬件体系结构上。

到了 1991 年的 10 月 5 日, Linus 在 comp.os.minix 新闻组上发布消息, 正式向外宣布 Linux 内核系统的诞生 (Free minix-like kernel sources for 386-AT)。这段消息可以称为 Linux 的诞生宣言, 并且一直广为流传。因此 10 月 5 日对 Linux 社区来说是一个特殊的日子, 许多后来 Linux 的新版本发布时都选择了这个日子。所以 RedHat 公司选择这个日子发布它的新系统也不是偶然的。

■ Linux 的命名

Linux 操作系统刚开始时并没有被称作 Linux，Linus 给他的操作系统取名为 FREAX，其英文含义是怪诞的、怪物、异想天开等意思。在他将新的操作系统上载到 ftp.funet.fi 服务器上时，管理员 Ari Lemke 很不喜欢这个名称。他认为既然是 Linus 的操作系统就取其谐音 Linux 作为该操作系统的目录吧，于是 Linux 这个名称就开始流传下来。

在 Linus 的自传《Just for Fun》一书中，Linus 解释说：“坦白地说，我从来没有想到过要用 Linux 这个名称发布这个操作系统，因为这个名字有些太自负了。而我为最终发布版准备的是什么呢？Freax。实际上，内核代码中某些早期的 Makefile——用于描述如何编译源代码的文件——文件中就已经包含有‘Freax’这个名字了，大约存在了半年左右。但其实这也没什么关系，在当时还不需要一个名字，因为我还没有向任何人发布过内核代码。而 Ari Lemke，他坚持要用自己的方式将内核代码放到 ftp 站点上，并且非常不喜欢 Freax 这个名字。他坚持要用现在这个名字（Linux），我承认当时我并没有跟他多争论。但这都是他取的名字。所以我可以光明正大地说我并不自负，或者部分坦白地说我并没有本位主义思想。但我想好吧，这也是个好名字，而且以后为这事我总能说服别人，就像我现在做的这样。”——Linus Torvalds 《Just for fun》第 84—88 页。

■ Linux 的读法

Linux 发音五花八门版本颇多，见到和听到的不下 10 种。根据 Linux 的创始人 LinusTorvalds 的说法，Linux 的发音和“Minix”是押韵的。“Li”中“i”的发音类似于“Minix”中“i”的发音，而“nux”中“u”的发音类似于英文单词“profess”中“o”的发音。依照国际音标应该是 /ˈlinəks/——类似于“里讷克斯”。但是，由于 Linus Torvalds 本人是芬兰人，所以他的 Linux 读音不是 /ˈlinəks/，而是 /ˈliniks/。

有人综合网上和 linux 自己的读音，概括出几个自认为最合适也最通用的读法：/liˈnʌks/（“里那克斯”）或 /liː nəks/（“里讷克斯”）或 /liˈnjuː ks/（“里纽克斯”）。这几个应该都是谁都能听得懂的。至于哪个比较正宗，当然是 linux 的原音。但事实上使用 linux 哪种读法的人似乎都不在少数。

■ 基本思想

Linux 的基本思想有两点：第一，一切都是文件；第二，每个软件都有确定的用途。其中第一条详细来讲就是系统中的所有都归结为一个文件，包括命令、硬件和软件设备、操作系统、进程等等对于操作系统内核而言，都被视为拥有各自特性或类型的文件。至于说 Linux 是基于 Unix 的，很大程度上也是因为这两者的基本思想十分相近。

■ 版本命名

Linux 内核有三个不同的命名方案。

早期版本：

第一个版本的内核是 0.01。其次是 0.02, 0.03, 0.10, 0.11, 0.12 (第一 GPL 版本), 0.95, 0.96, 0.97, 0.98, 0.99 及 1.0。

从 0.95 版有许多的补丁发布于主要版本之间。

旧计划 (1.0 和 2.6 版之间), 版本的格式为 A.B.C, 其中 A, B, C 代表:

A 大幅度转变的内核。这是很少发生变化, 只有当发生重大变化的代码和核心发生才会发生。在历史上曾改变两次的内核: 1994 年的 1.0 及 1996 年的 2.0。

B 是指一些重大修改的内核。

内核使用了传统的奇数次要版本号码的软件号码系统 (用偶数的次要版本号码来表示稳定版本)。

C 是指轻微修订的内核。这个数字当有安全补丁、bug 修复、新的功能或驱动程序, 内核便会有变化。

第三次, 自 2.6.0 (2003 年 12 月) 发布后, 人们认识到, 更短的发布周期将是有益的。自那时起, 版本的格式为 A.B.C.D, 其中 A, B, C, D 代表:

A 和 B 是无关紧要的

C 是内核的版本

D 是安全补丁

■ Linux 内核

Linux 是最受欢迎的自由电脑操作系统内核。它是一个用 C 语言和汇编语言写成, 符合 POSIX 标准的类 Unix 操作系统。Linux 最早是由芬兰黑客林纳斯·托瓦兹 (Linus B.Torvalds) 为尝试在英特尔 x86 架构上提供自由免费的类 Unix 操作系统而开发的。该计划开始于 1991 年, 林纳斯·托瓦兹当时在 Usenet 新闻组 comp.os.minix 登载帖子, 这份著名的帖子标示着 Linux 计划的正式开始。

在计划的早期有一些 Minix 黑客提供了协助, 而今天全球无数程序员正在为该计划无偿提供帮助。

技术上说 Linux 是一个内核。“内核”指的是一个提供硬件抽象层、磁盘及文件系统控制、多任务等功能的系统软件。一个内核不是一套完整的操作系统。一套基于 Linux 内核的完整操作系统叫做 Linux 操作系统, 或是 GNU/Linux。



图 1-6 《深入理解 LINUX 内核》

■ 桌面环境

在图形计算中，一个桌面环境（Desktop Environment，有时称为桌面管理器）为计算机提供一个图形用户界面（GUI）。这个名称来自桌面比拟，对应于早期的文字命令行界面（CLI）。一个典型的桌面环境提供图标，视窗，工具栏，文件夹，壁纸以及像拖放这样的能力。整体而言，桌面环境在设计 and 功能上的特性，赋予了它与众不同的外观和感觉。

现今主流的桌面环境有 KDE，gnome，Xfce，LXDE 等，除此之外还有 Ambient，EDE，IRIX Interactive Desktop，Mezzo，Sugar，CDE 等。

◆ KDE

KDE（Kool Desktop Environment）项目始建于 1996 年 10 月，相对于 GNOME 还要早一些。KDE 项目是由图形排版工具 Lyx 的开发者、一位名为 Matthias Ettrich 的德国人发起的，目的是为满足普通用户也能够通过简单易用的桌面来管理 Unix 工作站上的各种应用软件以及完成各种任务。

◆ GNOME

GNOME 即 GNU 网络对象模型环境（The GNU Network Object Model Environment），GNU 计划的一部分，开放源码运动的一个重要组成部分。是一种让使用者容易操作和设定电脑环境的工具。

目标是基于自由软件，为 Unix 或者类 Unix 操作系统构造一个功能完善、操作简单以及界面友好的桌面环境，他是 GNU 计划的正式桌面。

◆ XFCE

Xfce（XForms Common Environment）创建于 2007 年 7 月，类似于商业图形环境 CDE，是一个运行在各类 Unix 下的轻量级桌面环境。原作者 Olivier Fourdan 最先设计 XFce 是基于 XForms 三维图形库。Xfce 设计目的是用来提高系统的效率，在节省系统资源的同时，能够快速加载和执行应用程序。

■ Linux 发行版

Linux 发行版指的就是我们通常所说的“Linux 操作系统”，它可能是由一个组织，公司或者个人发行的。Linux 主要作为 Linux 发行版（通常被称为“distro”）的一部分而使用。通常来讲，一个 Linux 发行版包括 Linux 内核，将整个软件安装到电脑上的一套安装工具，各种 GNU 软件，其他的一些自由软件，在一些特定的 Linux 发行版中也有些专有软件。发行版为许多不同的目的而制作，包括对不同计算机结构的支持，对一个具体区域或语言的本地化，实时应用和嵌入式系统。目前，超过三百个发行版被积极开发，最普遍被使用的发行版有大约十二个。

一个典型的 Linux 发行版包括：Linux 核心，一些 GNU 库和工具，命令行 shell，图形界面的 X 窗口系统和相应的桌面环境，如 KDE 或 GNOME，并包含数千种从办公包、编