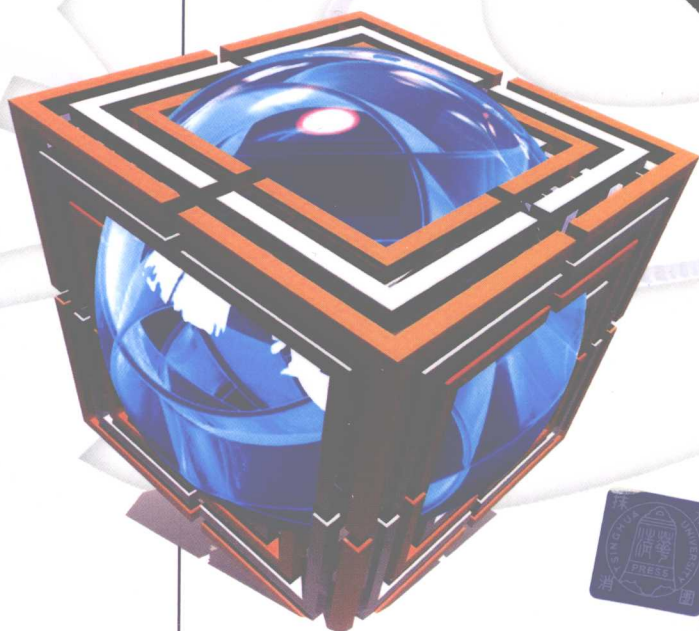


Linux 嵌入式 Qtopia 智能移动终端开发指南

Qtopia

编程之道

苗忠良 宛斌 编著



附光盘



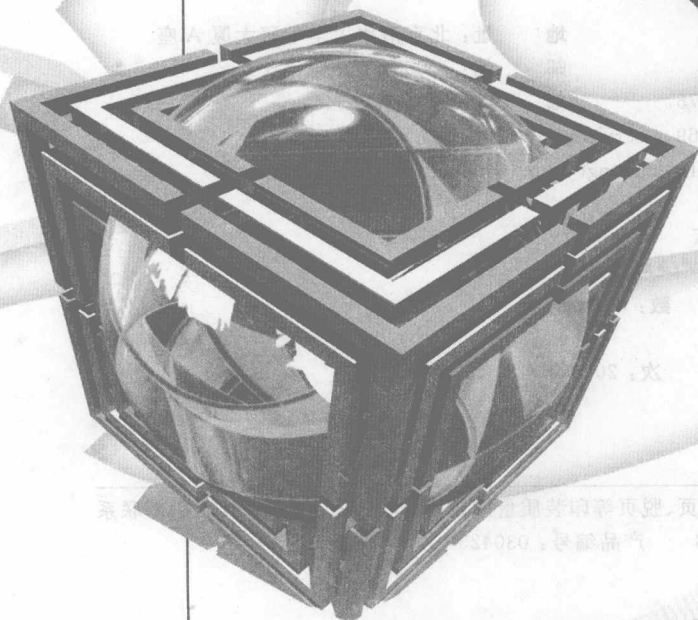
清华大学出版社

Linux 嵌入式 Qtopia 智能移动终端开发指南

Qtopia

编程之道

苗忠良 宛斌 编著



附光盘

清华大学出版社
北京

内 容 简 介

本书是一本针对 Linux 环境下利用 C++ 函数库 Qtopia 进行嵌入式研发的软件研发人员的入门书。本书融合了笔者在长期的软件研发过程中总结的大量实战经验,以智能移动终端的研发为背景,详细介绍了嵌入式 Linux 的研发环境和 Qtopia 的设计架构。本书适用的读者主要为从事嵌入式软件、C/C++ 等领域的初、中级软件工程师和在校本科生、研究生等。

Trolltech®、Trolltech、Qt®、Qt、Qtopia®、Qtopia、SXE™ 等为 Trolltech 公司的注册商标,本书涉及的所有代码遵循 GPL v2 许可。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Qtopia 编程之道 / 苗忠良, 宛斌编著. —北京: 清华大学出版社, 2009.1
ISBN 978-7-302-18700-4

I. Q… II. ①苗… ②宛… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 154733 号

责任编辑: 夏兆彦 王冰飞

责任校对: 徐俊伟

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京鑫海金澳胶印有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 24.5 字 数: 612 千字

附光盘 1 张

版 次: 2009 年 1 月第 1 版

印 次: 2009 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 48.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 030425-01

当前,嵌入式技术发展迅猛,已深入到各个领域,应用极为广泛。嵌入式系统是多学科领域知识的综合应用,已深入到各个行业的具体应用中。嵌入式系统具有软硬件紧密结合的特点,这使其设计与计算机领域的软件设计及电子工程领域的硬件设计都不同,它是计算机与电子工程设计两者的结合,是软硬件的综合设计。

嵌入式 Linux 操作系统不仅继承了 Linux 的源码开放、内核稳定性强、软件丰富等特点,而且还支持几乎所有的主流处理器和硬件平台。目前, Linux 在嵌入式系统中的应用已经非常普遍,它能给嵌入式开发人员提供很大的自由度,使其能掌控自己的设计。Linux 因其开源、免费等诸多优点,在嵌入式操作系统市场上扮演着越来越重要的角色,已经成为主流的嵌入式技术之一。

嵌入式硬件系统和 Linux 系统的有机结合,成为后 PC 时代计算机最普遍的应用形式。嵌入式 Linux 技术在中国有巨大的发展潜力和市场需求。有数据显示,未来两年里,在计算机、消费电子、通信、汽车电子、工业控制和军事国防这六大主要应用领域,嵌入式 Linux 产品将达到 80 亿美元的市场规模,可见这个行业的前景是非常乐观的。当然, Linux 嵌入式操作系统本身也有一定的局限性,就是开发难度过高,对于企业需要很高的技术实力。这就要求 Linux 系统用户不光要利用 Linux,更要掌握 Linux,尤其是掌握涉及 Linux 的具体的应用开发工具和平台。

Qtopia 是一种为采用嵌入式 Linux 操作系统的嵌入式电子设备而开发的知名软件平台。目前,国内能进行嵌入式 Linux 软件设计,尤其是采用 Qtopia 平台进行嵌入式产品开发的人才还很缺乏,学习也较为困难。本书的作者凭借数年的软件开发经验,总结归纳出了一本面向 Qtopia 开发技术人员以及对嵌入式 Linux 操作系统和 Qtopia 平台感兴趣的本科生、研究生的入门书。该书不同于一般的 Linux 指导性书籍,书中的每个章节都结合作者的亲身经历进行详细分析和讲解。

本书能对有志于从事嵌入式系统设计的学生和工程技术人员,在嵌入式 Linux 操作系统和 Qtopia 平台开发方面,提供较有价值的借鉴和指导作用,更快地提高嵌入式系统的软件设计能力及研发水平。因此,我向大家推荐本书。

吴介一
2008 年 8 月

人格心智是一个人能否幸福、快乐、成功的基础，在人类生活的交往过程中有重要的意义，当今社会无情的竞争给人带来的压力越来越大，以一种什么样的人格心态和思维方式来迎接这样的挑战，人格心智是否健康显得尤为重要。

心智决定视野，视野决定格局，格局决定命运，命运决定未来。

以上是管理学大师 Peter F. Drucker 在《卓有成效的管理者》中写的一段话，对于软件研发而言，笔者相信也适用之，并尝试以此作为本书的主旨来阐述相关的内容，与读者一起探讨软件研发的真谛。

作为一位从事嵌入式软件研发的研究人员，一直以来，都有一种强烈的愿望，希望自己构建出的代码足够完美。但数年的程序设计经验告诉我，编出完美的代码并不是一件很容易的事，受国内环境所限，众多的软件工程师在软件上的素养一直停留在编程语言的语法阶段，认为懂得语法、能写出可运行的代码就是编程的一切，能够通过调用系统提供的函数来实现所需要的功能就是所要完成的任务，很少用心去体会代码内在的设计原则和其中蕴含的哲理，很少去考虑蕴含其间的性能、健壮性、可读性、可扩展性等问题，更不用说去思考自己的设计是否已经是最优了。正如用砖头、水泥和钢筋去建造建筑，总以为自己能够将手头的材料组合成貌似需要的模样并能够大致地实现所需的功能就认为自己已经完成了项目的开发工作。其实不然，建造的建筑是否具有美感暂且不说，它是否足够牢固，是否经得起地震、飓风，是否为以后的住户的自我调整提供方便，事实上这些都是每个建筑师应该考虑的问题。作为软件，同样的问题一直存在，写出来的代码是否足够简洁，是否足够强壮，是否有足够的可扩展性？这些概念很多人都懂，但很遗憾的是却很少有人在自己从事研发的时候去考虑，或是迫于工作进度的压力，而往往把迅速地完成任务作为自己考虑的第一要素，即使最终时间还宽裕，也往往因为潜意识中任务已经完成而不再尝试去进一步的优化。

如何才能成为一个好的研发人员？笔者自己也一直在苦苦探索，综合前人的经验，笔者认为大致有以下几点：第一，要有扎实的基础，不积跬步，无以至千里，不积小流，无以成江海，扎实的基本功是攀登高峰的基础；第二，要有开阔的视野和思维，大鹏展翅九万里，才能翻动扶摇羊角，只有拥有开阔的视野，对自己、对周围的同事和朋友、对业界的发展、对技术的走向、对整个社会的发展具有深刻的认知和洞察力，才能更好地把握前进的方向，才能有效地察觉自己的优点和不足，才能在每日三省其身之中获得不断的进步；第三，要乐于探索，坚持不懈，成功往往是百分之九十九的汗水加百分之一的灵感，成功之花需要你的辛勤付出才能开的更艳，只有不畏艰难，勇于追求真理的人们才能在科学的道路上走的更高更远；第四，要有良好的品德和团队合作素养，随着需求不断增加，现代软件的复杂度也不断上升，软件开发早已不是一个人两个人所能完

成的工作，现代的软件开发更强调团队合作，如何保证你的团队高效、有序运转？除了一套完善的制度外，还需要团队成员具备优秀的个人素养、良好的道德品质、公平公开公正的做事风范，这样才能赢得同事们的高度信任，彼此间相互保留甚至猜疑的团队是没有生命力的。另外，细心的读者应该早有体会，世间的万物并不是孤立存在的，彼此间存在着千丝万缕的联系，最重大的发明和发现所蕴含的哲理往往在普通的事物上已经有所体现，事物千变万化的背后往往体现了最朴素的哲学，以软件为例，设计模式的众多模式背后隐藏的原则不过寥寥数语。因此笔者建议，闲暇时读读工作外的书籍，看看历史，品品哲学，你将走的更远。

在本书的撰写过程中，笔者力求在书中体现自己参悟到的一些哲理，尝试从设计背后隐藏的原则着手，为软件研发引入更多的思考元素，尝试说明软件从来就不是单纯的代码编写，优雅的代码背后往往蕴含着深刻的设计原则。授人以鱼不如授人以渔，慢慢看来，也许你会发现本书的内在价值。当然笔者功力有限，所想所写可能存在着这样那样的不足，吾生也有涯，而知也无涯，希望与同道者共勉！

适用读者

本书适合的读者范围包括：从事嵌入式软件、C/C++等领域的初、中级软件工程师和在校本科生、研究生等。

需要说明的是本书主要侧重于研发环境的搭建和 Qt 的内在设计与架构，而非一本简单的实例教程，关于如何利用 Qt 进行编程的实例，请参考 Qt 自带的教程和帮助文档。

预备知识

本书假设读者已具备如下方面的知识：

- 面向对象程序设计语言（C++）的基础知识。
- 面向对象技术的基本概念如类、对象、封装、继承等。
- Linux 操作系统和嵌入式软件研发语言 Qtopia 等的基础知识。

本书的组织

本书共四部分 13 章，其中第 1 章和第 2 章为第一部分，介绍了 Linux 的基础知识和移动终端的基础知识；第 3~10 章为第二部分，介绍了 Qtopia 的内在设计与架构；第 11~12 章为第三部分，介绍了人机交互和软件设计的一些重要原则；第 13 章为第四部分，介绍了一些从事嵌入式软件研发的经典书籍，便于扩大读者的视野。

致谢

首先，非常感谢我们的爸爸妈妈，是他们的辛勤培养造就了我们今天的成就，没有他们，就没有我们今天的一切，衷心的祝福他们永远快乐、健康长寿！

其次，非常感谢我们在单位的领导张孝林博士，谢谢他给了我们一个宽松的研发环境；同事李艳平、崔恒利、宣彩平、陈敬诚、张红涛、周阳霖等，实习生杨鹏飞、何伟伟、韩振中、赵志岳等，朋友叶峰、孔德涛、管艳丽等在本书的撰写过程中帮助校对了本书初稿中存在的一些不足并提出了很多很有价值的建议，他们都是业界一流的软件工程师，很高兴能够和他们一起交流经验、探讨学术话题，在此对他们深表感谢！

接着，要感谢的是我们在东南大学求学时期的导师吴介一教授、张飒兵高工和实验室的同门师兄（姐）们，和他们相处的美好时光，笔者终身难忘，他（她）们渊博的学识和良好的道德修养，让笔者受益匪浅。其中吴老师在百忙中还特地抽出宝贵的时间为本书作序，在此笔者深表谢意！

另外，需要感谢的是在开源领域持续奋斗的人们，九层之台，起于垒土，没有他们的辛勤努力，本书也就失去可以撰写的基础。

最后，笔者在撰写本书过程中，参考了网上一些朋友的见解，在这里也一并表示感谢！

学海无涯，在学习这条道路上，笔者本人也仅是刚刚迈上大道！撰写本书的主要目的在于和大家一起分享一些笔者的研发经验，在交流中一起提高在软件方面的素养。

衷心希望本书能够为在无边黑暗中苦苦探索的人们带来些许光明。

笔者

2008年8月 于张江

元为希

约 定

本书的最终完成，是两位作者共同呕心沥血的结果，为了叙述的方便，在本书中以“笔者”来指代两位作者。

1. 作者 为了说明的方便，对于本书中采用的一些符号和名词术语做如下的约定：

本书的最终完成，是两位作者共同呕心沥血的结果，为了叙述的方便，在本书中以“笔者”来指代两位作者。

2. 注释 在本书中在为代码或者命令选项进行注释时，无论是 Linux 平台还是 Windows 平台，单行注释采用“//”的方式进行，多行注释采用“/* ... */”的方式进行。

3. 终端提示符 在本书中，常要用到 Linux 终端命令。为了方便起见，终端提示符统一用“#”表示。

4. Qt 与 Qtopia 为了说明的方便，在本书中，默认情况下所说的 Qt 是指 Qt 和 Qtopia 的公共部分，而 Qtopia 则是指 Qtopia 中特有的部分。

5. 省略 出于缩减篇幅的考虑，本书对涉及的代码进行了有选择性的删减，其中“...”表示此处省略了部分代码。完整的代码请参考随书光盘。

第 1 章 Linux 基础	1
1.1 Linux 的发展历程	1
1.1.1 开源运动	3
1.1.2 Linux 内核	6
1.1.3 桌面环境	9
1.1.4 发行版	11
1.2 常用工具	13
1.3 编译与调试	24
1.3.1 编译配置	24
1.3.2 交叉编译	32
1.3.3 GDB 调试	40
1.3.4 内存调试	44
1.4 文件系统	46
1.4.1 NFS	46
1.4.2 EXT2/EXT3	47
1.4.3 ReiserFS	48
1.5 安装方法	49
1.5.1 虚拟机方式	49
1.5.2 双系统方式	53
1.5.3 Samba 服务	54

第 2 章 移动终端基础	57
2.1 蜂窝通信	57
2.2 操作系统	59
2.2.1 Symbian	59
2.2.2 Linux	61
2.2.3 Windows Mobile	62
2.3 研发语言	64
2.3.1 J2ME	64
2.3.2 BREW	65
2.3.3 Qtopia	66
2.4 展望	67

第 3 章 Qtopia 基础	71
3.1 Qtopia 编译	72

3.1.1 编译 Qtopia	72
3.1.2 环境变量	72
3.1.3 启动模拟器	75
3.2 研发环境	82
3.2.1 UML 工具	82
3.2.2 集成开发环境	84
3.2.3 源代码分析	88
3.2.4 差异比较	92
3.3 Qt 工具	95
3.3.1 Qt Designer	95
3.3.2 Qt Assistant	96
3.3.3 Qt Linguist	96
3.3.4 qmake	97
第 4 章 核心特性	99
4.1 父子化	103
4.2 元对象系统	106
4.2.1 信号与槽	107
4.2.2 动态属性	116
4.2.3 软件本地化	121
4.3 布局管理	124
4.4 插件系统	131
4.4.1 系统扩展	131
4.4.2 应用扩展	135
4.5 拖放	136
4.6 多线程	139
4.7 文件管理	149
第 5 章 通信机制	154
5.1 事件处理	154
5.1.1 产生事件	155
5.1.2 分发事件	161
5.1.3 事件循环	165
5.1.4 事件队列	168
5.1.5 事件过滤器	173

5.1.6	目标窗体	174
5.2	套接字通信	179
5.2.1	创建信道	179
5.2.2	注册信道	181
5.2.3	发送消息	183
5.2.4	接收消息	184
5.2.5	系统信道	185
第 6 章	渲染机制	189
6.1	着色系统	189
6.1.1	着色	191
6.1.2	填充	194
6.1.3	着色设备	196
6.1.4	读写图像	199
6.1.5	风格定制	201
6.2	图形引擎	201
6.2.1	图形加速	204
6.2.2	帧缓冲	212
6.2.3	屏幕显示	213
第 7 章	配置与编译	216
7.1	编译系统	216
7.1.1	环境变量	216
7.1.2	编译模式	217
7.1.3	调试方法	218
7.1.4	编译过程	219
7.2	应用程序配置	220
7.3	配置文件	225
7.3.1	Pro 文件	225
7.3.2	Pri 文件	230
7.3.3	Desktop 文件	236
7.3.4	Conf 文件	236
7.4	模拟器配置	238
7.5	分布式编译	239
7.6	链接编辑	243
第 8 章	启动过程	247
8.1	C/S 模型	247
8.2	服务器	249
8.2.1	初始化过程	249
8.2.2	启动过程	254
8.3	应用程序	256

第 9 章	风格与主题	262
9.1	风格	262
9.2	主题	265
第 10 章	集成与安装	277
10.1	文件系统	277
10.1.1	ROMFS	279
10.1.2	RAMFS/TMPFS	280
10.1.3	JFSS2	281
10.1.4	CRAMFS	282
10.1.5	YAFFS2	284
10.1.6	UNIONFS	285
10.2	下载与安装	286
10.2.1	通信接口	286
10.2.2	服务配置	288
10.3	触摸屏适配	294
10.3.1	校准原理	297
10.3.2	滤波算法	300
10.3.3	去抖算法	302
10.3.4	坐标变换	304
第 11 章	设计之道	306
11.1	人机交互	308
11.1.1	用户第一	309
11.1.2	概念模型	310
11.1.3	可视性	311
11.1.4	匹配原则	311
11.1.5	反馈原则	313
11.2	软件设计	314
11.2.1	研究目的	314
11.2.2	命名原则	315
11.2.3	面向对象	316
第 12 章	Qtopia 设计	323
12.1	模块设计	323
12.1.1	组合模式	324
12.1.2	单子模式	330
12.1.3	观察者模式	332
12.1.4	工厂模式	337
12.1.5	Qt 模式	341
12.2	架构设计	343

12.2.1 Qtopia Core	343	13.5 软件工程.....	368
12.2.2 Model/View	345	13.6 通用程序设计.....	372
第 13 章 书籍简介	350	13.7 嵌入式.....	373
13.1 Qt.....	350	13.8 无线通信.....	375
13.2 C/C++.....	351	13.9 计算机网络.....	376
13.3 Linux.....	361	13.10 其他.....	377
13.4 设计模式.....	366	参考文献	379

第1章 Linux 基础

知之者不如好之者，好之者不如乐之者。

《论语·雍也》

从1991年Linus Torvalds在新闻组comp.os.minix发布了大约有一万行代码的Linux v0.01版本起，Linux已经走过了波澜壮阔的16年历程。随着Linux的蓬勃发展和开源运动的越演越烈，从最简单的嵌入式设备到最复杂的超级计算机，随处可见Linux的身影，从最初的只具备最基本功能的丑小鸭，到今天不断吞噬其他操作系统市场份额的白天鹅，Linux已经开始展翅翱翔。

1.1 Linux 的发展历程

“Hello everybody out there using minix—I’m doing a (free) operating system”在1991年的8月，网络上出现了一篇以此为开篇话语的帖子，这是当时一个芬兰的名为Linus Torvalds的大学生在为自己开始编写的一个类似Minix，并可运行在CPU为Intel 386的操作系统寻找志同道合的合作伙伴，从此揭开了Linux的序幕。

1991年10月5日，Linus Torvalds在新闻组comp.os.minix发布了大约有一万行代码的Linux v0.01版本。

到了1992年，大约有1000人在使用Linux，值得一提的是，他们基本上都属于真正意义上的骇客(hacker)。

1993年，大约有100余名程序员参与到Linux内核代码的编写、修改工作中，其中核心组由5人组成，此时的最新版本Linux 0.99的代码已经达到大约十万行，用户达到大约10万左右。

1994年3月，Linux 1.0发布，代码量达到17万行，当时是按照完全自由免费的协议发布，随后正式采用GPL v2许可。至此，Linux的代码开发进入良性循环。很多系统管理员开始在自己的系统环境中尝试Linux，并将修改的代码提交给核心小组。由于拥有了丰富的系统平台，因而Linux的代码中也充实了对不同硬件系统的支持，大大地提高了跨平台移植性。

1995年，此时的Linux已经可以在Intel、Digital及Sun SPARC处理器上运行，用户量超过了50万，介绍Linux的Linux Journal杂志的发行量也达到10万册之多。

1996年6月，Linux 2.0内核发布，代码量达到了大约40万行，并可以支持多个处理器。此时的Linux已经进入了实用阶段，全球大约有350万人在使用。

1997年夏，在为大片《泰坦尼克号》制作特效使用的160台Alpha图形工作站中，有105台采用了Linux操作系统。

1998年是Linux迅猛发展的一年。1月，Red Hat高级研发实验室成立。随后的Red Hat 5.0获得了InfoWorld的操作系统奖项。4月，Mozilla发布浏览器代码，Firefox开始

成为 Linux 图形界面上的王牌浏览器, Red Hat 宣布商业支持计划, 网罗了多名优秀技术人员开始商业运作。这一年, 王牌搜索引擎“Google”现身, 采用的也是 Linux 服务器。值得一提的是, 由于当时 Oracle 和 Informix 两家数据库厂商明确表示不支持 Linux, 这个决定给 Mysql 数据库提供了充分的发展机会。10 月, Intel 和 Netscape 宣布小额投资 Red Hat 公司, 这被业界视作 Linux 获得商业认同的信号。同月, 微软在法国发布了反 Linux 公开信, 这表明微软公司开始将 Linux 视作了一个对手来对待。12 月, IBM 发布了适用于 Linux 的文件系统 AFS 3.5 以及 Jikes Java 编辑器和 Secure Mailer 及 DB2 测试版, IBM 的此番行为, 可以看作是 IBM 与 Linux 羞答答的第一次亲密接触。迫于 Windows 和 Linux 的压力, 这一年 Sun 开始逐渐开放 Java, 并且在 UltraSparc 上开始支持 Linux 操作系统。1998 年可以说是 Linux 与商业接触的一年。

1999 年, IBM 宣布与 Red Hat 公司建立伙伴关系, 以确保 Red Hat Linux 在 IBM 机器上正确运行。3 月, 第一届 LinuxWorld 大会的召开, 宣布 Linux 时代的来临。IBM、Compaq 和 Novell 宣布投资 Red Hat 公司, 以前一直对 Linux 持否定态度的 Oracle 公司也宣布进行投资。5 月, SGI 公司宣布向 Linux 移植其先进的 XFS 文件系统。对于服务器来说, 高效可靠的文件系统是不可或缺的, SGI 的慷慨帮助 Linux 确立了在服务器市场的专业性地位。7 月, IBM 启动对 Linux 的支持服务和发布了 Linux DB2, 从此结束了 Linux 得不到支持服务的历史, 这可以视作 Linux 真正成为服务器操作系统一员的重要里程碑。

2000 年初始, Sun 公司宣布 Solaris 8 降低售价, Sun 感受到了 Linux 带来的巨大冲击。2 月, Red Hat 发布了嵌入式 Linux 的研发环境, Linux 在嵌入式领域的潜力逐渐被发掘出来。4 月, TurboLinux 公司宣布推出中国首家 Linux 工程师认证考试, 从此使 Linux 操作系统管理员的水准可以得到权威机构的资格认证, 此举大大增加了国内 Linux 爱好者学习的热情。伴随着国际上的 Linux 热潮, 国内的联想推出了“幸福 Linux 家用版”。7 月, 中科院与新华科技合作发展红旗 Linux, 此举让更多的国内个人用户认识了 Linux 这个操作系统。11 月, Intel 与 Xteam 合作, 推出基于 Linux 的网络专用服务器, 此举结束了 Linux 单向顺应硬件开发驱动的历史。

2001 年新年伊始就爆出新闻, Oracle 宣布在 OTN 上的所有会员都可免费索取 Oracle 9i 的 Linux 版本, 从 Oracle 几年前的“绝不涉足 Linux 系统”到如今的主动献媚, 足以从侧面体现 Linux 的发展迅猛。IBM 则决定投入 10 亿美元扩大 Linux 系统的运用, 此举犹如一针强心剂, 令华尔街的投资者们闻风而动。到了 5 月这个初夏的时节, 微软公开反对“GPL”, 这一举动引起了一场大规模的论战。8 月, “红色代码”病毒爆发, 引得许多站点纷纷从 Windows 操作系统转向 Linux 操作系统, 虽然是一次被动的转变, 但使 Linux 获得了进一步的普及。12 月, Red Hat 为 IBM S/390 大型计算机提供了 Linux 解决方案, 从此结束了 AIX 孤单独行的历史。

2002 年则是 Linux 企业化的一年。2 月, 微软公司迫于各国政府的压力, 宣布扩大公开代码行动, 这可是 Linux 开源带来深刻影响的结果。3 月, 内核开发者宣布新的 Linux 系统对 64 位的计算机提供支持。

2003 年 1 月, NEC 宣布将在其手机中使用 Linux 操作系统, 代表着 Linux 成功进军手机领域。5 月, SCO 表示就 Linux 使用的涉嫌未授权代码等问题对 IBM 进行起诉, 此

时人们才留意到,原本由 SCO 垄断的银行、金融领域,份额已经被 Linux 抢占了不少,也难怪 SCO 如此气急败坏了。9 月,中科红旗发布 Red Flag Server 4 版本,性能改进明显。11 月,IBM 注资 Novell 以 2.1 亿收购 SuSE Linux,同期 Red Hat 停止免费的 Linux,顿时业内骂声四起。Linux 在商业化的路上开始渐行渐远,12 月, Linux 2.6.0 内核发布。

2004 年 1 月,本着“天下事分久必合,合久必分”之天理, SuSE Linux 嫁到了 Novell, SCO 继续顶着骂名四处强行“化缘”, Asianux、MandrakeSoft 在涉足 Linux 的五年中首次宣布季度赢利。3 月, SGI 宣布成功实现了 Linux 操作系统对 256 个 Itanium 2 处理器的支持。4 月,美国斯坦福大学 Linux 大型机系统被黑客攻陷,再次证明了没有绝对安全的操作系统。6 月,统计报告显示在世界 500 强超级计算机系统中,使用 Linux 操作系统的超级计算机已经占到了 280 席,抢占了许多原本属于各种 UNIX 的份额。9 月, HP 开始网罗 Linux 内核代码人员,以影响新版本的内核朝着对 HP 有利的方向发展,而 IBM 则准备推出 OpenPower 服务器,仅运行 Linux 系统。

2005 年,刚刚进入中国市场的 Novell、Red Hat 显示出咄咄逼人的势头,其中在 2005 年上半年, Novell 在国内 Linux 服务器销售收入市场占有率达 32.9%,产品出货量市场占有率达 30%。12 月, Trolltech 公司携 Qtopia 2.2 协助信息产业部电信研究院在北京召开“Linux 助力移动通信 2005 中国 Linux 移动通信大会”。Linux 开始在中国进入移动通信领域,同年,由于 Motorola 大力支持的缘故,基于 Linux 的智能手机的销售量在全球强劲增长。

2006 年 4 月,中国的北京中科红旗软件技术有限公司、日本的 Miracle Linux 公司以及韩国的韩软公司联合宣布即将组建一个新的合资企业 Asianux 公司,开发 Asianux 操作系统和开放源码软件。其目标是成为亚洲最主要的统一 Linux 平台 and 世界 Linux 第三极。8 月, Novell 正式发布了其全新的 SUSE Linux 企业版 10,为企业提供了从桌面、服务器到数据中心等应用的计算平台。11 月, Oracle 的 CEO Larry Ellison 在 Oracle Open World 会议上宣布:“甲骨文将提供完整的 Red Hat Linux 支持。若你是 Red Hat 支持服务的客户,可以轻而易举地作切换,改用甲骨文提供的支持”。同月, Microsoft 和 Novell 两家公司达成协议,为 Windows 和 Linux 提供兼容技术,促使两种操作系统可以联袂运行。

2007 年 5 月, Dell 开始推出预装了 Ubuntu 的计算机, HP、Lenovo、ASUS 随后跟进, Linux 在桌面 PC 领域逐渐被普通用户接受。同月, Microsoft 的 CEO Steve Ballmer 声称 Linux 侵犯了 Microsoft 共 200 多项专利,但微软同时表示,不会起诉 Linux 用户,对 Linux 用户而言是个不小的恐吓。8 月, Novell 在与 SCO 法律战中赢得胜利,这一判决意味着“笼罩在 Linux 头上的乌云已经烟消云散”。11 月, FSF 发布了最新的许可协议 GPL v3。

“谁会牵你的手,走过风风雨雨,……”这句歌词曾经代表着千万 Linuxer 的心,如今,这只可爱的小企鹅终于能独当一面,在 IBM、HP、Novell、Oracle 等诸多厂商的支持下,迎着风雪傲然前行。

1.1.1 开源运动

严格意义上的 Linux 仅是指 Linux 内核,不包括外围的应用软件。但没有外围的应

用软件，Linux 内核挑不起操作系统的大旗。说到 Linux 的外围软件，就不得不提 GNU 计划及其领袖人物 Richard Stallman。

下面先说说 Richard Stallman 其人：

Richard Stallman 是开源软件的始创者和精神领袖。他是 GNU (GNU's Not UNIX) 计划的发起人，自由软件基金会 (Free Software Foundation, FSF) 的缔造者，同时还是大名鼎鼎的编辑器 GNU Emacs、GNU 编译器 (GNU Compiler Collection, GCC)、GNU 调试器 (GNU Project Debugger, GDB) 和 GNU Make 的编写者。

Stallman 的生平是和开源历史紧密相关的：

1984 年，Stallman 辞去了在 MIT AI 实验室的工作，开始编写 GNU 软件。

1985 年，Stallman 创立了自由软件基金会 FSF。

1991 年，Linus Torvalds 研发的 Linux 加入到 GNU 计划中，和其他 GNU 软件一起组成了一个完整的自由软件操作系统。

1998 年，FSF 倡导自由软件 (Free Software) 更名为开源软件 (Open Source Software)。但 Stallman 反对这一做法，他认为“Free”一词，表达的不仅仅是技术，更重要的是思想。

除了技术超群，Stallman 对自由信念的追求则更让人钦佩。正是因为他的努力，才有今天百花齐放的开源软件世界。如果没有开源软件，很多人将会用不起计算机，很多企业也会没钱购买昂贵的服务器系统，相应的网络应用、企业应用都将大打折扣。

1. GNU 计划

GNU 计划又称“革奴计划”，是由 Richard Stallman 在 1983 年 9 月 27 日公开发起的。它的目标是创建一套完全自由的操作系统。Richard Stallman 最早是在 net.UNIX-wizards 新闻组上公布该消息，并附带一份“GNU 宣言”来解释为何发起该计划，其中一个理由就是要“重现当年软件界合作互助的团结精神”。

GNU 是“GNU's Not UNIX”的缩写。Stallman 宣布 GNU 应当发音为“Guh-NOO”，与“canoe”发音相同，以避免与 gnu (非洲牛羚，发音与 new 相同) 这个单词混淆。

由于 GNU 遵循 UNIX 系统的接口标准，因此 GNU 计划可以分别研发不同的操作系统组件。GNU 计划采用了部分当时已经可以自由使用的软件如 TeX 排版系统、X Window 视窗系统等。不过 GNU 计划也研发了大批其他的自由软件。

为保证 GNU 软件可以自由地“使用、复制、修改和发布”，所有 GNU 软件都遵循一份在禁止其他人添加任何限制的情况下保证授予全部权利给任何人的协议条款即 GNU 通用公共许可证 (GNU General Public License, GPL)。这个就是被称为“反版权” (或称 Copyleft) 的概念。

1985 年 Richard Stallman 创立了自由软件基金会 FSF 来为 GNU 计划提供技术、法律以及财政支持。尽管 GNU 计划的项目大部分时候是由个人自愿无偿贡献，但 FSF 有时还是会聘请程序员帮助编写。当 GNU 计划开始逐渐获得成功时，一些商业公司开始介入其中进行研发和提供技术支持。其中最著名的公司就是后来被 Red Hat 兼并的 Cygnus Solutions。

到了 1990 年，GNU 计划已经研发出的软件包括一个功能强大的文字编辑器 Emacs、

GNU 编译器 GCC 及大部分 UNIX 系统的程序库和工具。唯一依然没有完成的重要组件就是操作系统的内核（称为 GNU HURD）。

1991 年 Linus Torvalds 编写出了与 UNIX 兼容的 Linux 操作系统内核并在 GPL 许可下发布。随后 Linux 在网上广泛流传，许多程序员参与了其研发过程。1992 年 Linux 与其他 GNU 软件结合，完全自由的 Linux 操作系统正式诞生。该操作系统往往被称为“GNU/Linux”或简称 Linux。

另外，许多 UNIX 系统上也安装了 GNU 软件，因为 GNU 软件的质量比之前 UNIX 的软件还要好。GNU 软件还被广泛地移植到 Windows 操作系统和 Mac 操作系统上。

开源软件的最著名的官方网站为：<http://sourceforge.net>，在这里你可以下载到所需的大多数开源软件。

2. GNU 在 Windows 下的使用

如果想在 Windows 操作系统下使用 GNU，有两个工具可以为你提供帮助，即 MinGW 和 Cygwin。

➤ MinGW

MinGW 即 Minimalist GNU For Windows。它是一些头文件和端口库的集合，该集合允许人们在没有第三方动态链接库的情况下使用 GCC 编译 Windows 32 程序。

在基本层，MinGW 是一组包含文件和端口库的集合，其功能是允许控制台模式的程序使用微软的标准 C 运行时间库（MSVCRT.DLL），该库在所有的 NT OS 和 Windows 95 以上的 Windows 操作系统版本上有效，使用标准 C 运行时间库，你可以使用 GCC 编写控制台模式的符合美国标准化组织（ANSI）的程序，可以使用微软提供的标准 C 运行时间扩展。该功能是 Windows 32 API 所不具备的。另一个组成部分是 W32API 包，它是一组可以使用 Windows 32 API 的头文件和端口库。与标准 C 运行时间库相结合，就可以有充分的权利既使用 CRT（C Runtime）又使用 Windows 32 API 功能。

MinGW 的官方网站为 <http://www.mingw.org>。MinGW 的最新版本为 2007 年 1 月发布的 5.1.3 版本。

部分情况下，为了使用更多的 GNU 工具，研发人员还需要另外安装 MSYS（Minimal GNU（POSIX）SYStem on windows），MSYS 是一个小型的 GNU 环境，包括基本的 bash、make、vim、tar 等 GNU 工具。MSYS 是 Windows 下最优秀的 GNU 环境。MSYS 的最新版本为 MSYS 1.0，官方网站为 <http://www.mingw.org>。

➤ Cygwin

Cygwin 是 Cygwin 公司开发的产品，它提供了 Windows 操作系统下的一个 UNIX/Linux 环境，可以帮助程序研发人员把应用程序从 UNIX/Linux 移植到 Windows 平台，是一个功能强大的工具集。

Cygwin 由两部分组成：

- ✧ cygwin1.dll，它作为 UNIX/Linux 的一个仿真层，提供 UNIX/Linux API 功能。

- ✧ 一组工具，它的功能是负责创建一个 Linux 的观感。

Cygwin 动态链接库可以在除 Windows CE 外的所有 X86 架构下的 Windows 操作系统如 Windows 98、Windows 2000、Windows XP 等上运行。

Cygwin 的官方网址为 <http://cygwin.com>。Cygwin 的最新版本为 2007 年 12 月发布的 1.5.25 版本。

网络管理人员通过 Cygwin 可以很容易地远程登录到任何一台 PC 机, 在 Linux 的 Shell 下解决问题, 在任何一台 Windows OS 计算机上运行 Shell 脚本命令。高级外壳脚本命令可以用标准 shell、sed 和 awk 等创建。标准 Windows 命令行工具甚至可以与 Linux 外壳脚本环境共同管理 Windows 操作系统。

需要说明的是, MinGW 基于 GPL v2 许可发布, 用户可以免费使用, 但 Cygwin 则是商业软件, 需要掏腰包才能使用, 因此 MinGW 在业界有更广泛的用户群。

1.1.2 Linux 内核

Linux 是一款类 UNIX 的操作系统, 除了内核外, Linux 系统中常用的系统程序都是由美国自由软件基金会 FSF 研发出来的, 同时还有不少机构和个人在为 Linux 研发应用程序。

Linux 具有 UNIX 操作系统的程序接口和操作方式, 也继承了 UNIX 稳定高效的特点, 由于其在 GPL v2 的版权下发布, 有着源码开放、应用丰富、可定制、低成本等优点, 逐渐成为一款非常受人欢迎的多用户多任务操作系统, Linux 能够在 X86、ARM、SPARC、Alpha、MIPS、Power PC 等不同的计算机硬件平台上运行。

在长期的发展中, Linux 形成了以下特色:

- 支持多用户多任务。
- 支持多 CPU。
- RAM 保护模式, 程序之间不会相互干扰, 保证了系统能长久无错运行。
- 动态加载程序, 当程序加载 RAM 执行时, Linux 仅将磁盘中相关的程序模块加载, 有效地提升了执行的效率和 RAM 管理。
- 支持多种文件系统, 如 FAT32、ReiserFS、EXT2/EXT3 等。

Linux 内核的版本号分为主版本号、次版本号和扩展版本号等, 根据稳定版本、测试版本和研发版本定义不同版本序列。

稳定的主版本号用偶数表示, 例如: 2.2、2.4、2.6 等。

紧接着是次版本号, 例如: 2.6.13、2.6.14 等, 次版本号不分奇偶数, 顺序递增。每隔 1~2 个月发布一个稳定版本。

然后是升级版本号, 例如: 2.6.14.3 等, 升级版本号部分奇偶数, 顺序递增, 每月几次发布升级版本号, 修正当前稳定版本中存在的问题。

另外一种测试版本, 在下一个稳定版本发布之前, 每个月发布几个测试版本, 例如: 2.6.12-rc1。通过大规模的测试, 可以使发布的稳定版本内核具有很高的稳定性。

还有一类是研发版本。研发版本的主版本号用奇数表示, 例如: 2.3、2.5。也有次版本号, 例如: 2.5.32 等。研发版本是不稳定的, 适合内核研发者在新的稳定的主版本发布之前使用。

Linux 内核的重要特点是具有很强的可移植性 (Portability), 支持多种硬件平台, 能够在大多数 CPU 体系结构上运行。Linux 内核除支持 32 位体系结构外, 对 64 位的体系