

红旗
Linux

[内附光盘]



红旗
红旗 Linux

软件开发技术

中科红旗软件技术有限公司 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



红旗 Linux 软件开发技术

中科红旗软件技术有限公司 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书内容主要为 Linux 的编程方法和技巧的介绍,重点在于 Linux 下的 C 语言编程和系统的调用,以及编译工具、调试工具的使用技巧。在 GNU make, CVS, lex 和 yacc 的应用上也进行了独到的讲解和示范,对 Qt 和 KDE 常用工具的综合使用进行了特别阐述。

本书特别适于编程技术人员和计算机专业人员阅读参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: 红旗 Linux 软件开发技术

作 者: 中科红旗软件技术有限公司

责任编辑: 丁 岭

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 20.5 字数: 486 千字

版 次: 2001 年 3 月第 1 版 2001 年 3 月第 1 次印刷

书 号: ISBN 7-900631-24-0

印 数: 0001 ~ 5000

定 价: 42.00 元

本丛书编委会

特邀编委：孙玉芳 刘 博 焦金生
编委成员：吴 健 贺唯佳 宋力秋
籍 辉 淮晋阳 韩 示

参与本书编写：邢国良 陈 剑 孙燕妮

序

在当今的电脑行业中存在着太多的传奇和神奇,各种在过去不可想象的“天方夜谭”似的财富聚敛、少年得志、一夜成名,都已经在 IT 行业中司空见惯了。在某种意义上说“应网而生”的 Linux 操作系统,在短短几年时间里已经发展成为一个相当完善的系统,绽放出 UNIX 世界的一朵奇葩。

Linux 起源于有悠久历史的 UNIX 操作系统。笔者从事计算机事业三十年。前十年正处于我国还未“开放”的时代,注重于计算机体系结构设计和国产的完全自主的操作系统、汇编解释器和语言编译器“三大件”的研制。后二十年正好与我国改革开放同时,注重于 UNIX 操作系统的研究和开发。可以说,笔者最壮丽的人生献给了 UNIX/Linux。

众所周知,UNIX 经过了三十来年的发展和完善,已相当可靠、稳定。可遗憾的是,它主要运行在大中型服务器和较昂贵的工作站上,广大的普通用户则难得一见,这也正好让微软的 Windows 占据了统治地位。现在有了 Linux,任何人都可以在手边的微机上学习、使用 UNIX/Linux 的各种功能了。

简单地说,Linux 是一套低价或免费使用、自由传播的类 UNIX 操作系统。与 Windows 不同,它可以运行在目前世界上几乎所有比较流行的 CPU 构成的电脑上,特别是 Intel 386 以上 CPU 的电脑上。这个系统是由世界各地在千上万的程序员借助互联网,遵循自由软件精神,以一种完全不同于传统软件工程的方法合作设计和开发出来的。它源码公开,其目的是建立不受任何商品化软件的版权制约,全世界都能自由使用的 UNIX 兼容产品。

正是基于在 UNIX 上研究开发了二十年的经验和以前开发国产系统软件(包括操作系统)的教训,笔者所在的中科院软件研究所于 1999 年 8 月推出红旗系列 Linux 发行产品,以期对国产操作系统的发展,对民族软件产业的壮大做出应有的贡献。

然而要使这面“红旗”能在中国土地上长远飘扬下去,而且能让广大用户所接受,需要做大量艰苦的工作。而其中特别重要的一环就是教育和培训。这一方面的努力可以改变用户的心理形态,另一方面也为“红旗”下一步的发展培养大量的人才和后备军。“红旗”清醒地意识到自己肩负的使命,所以不惜重金利用其他宝贵的资源,组织编写了不同层次、不同目的的成套图书和教材。

本书侧重于在 Linux 下进行软件开发,这意味着本书的重点是面向编程人员而非普通使用者的。本书重点介绍了 C 语言函数库和 Linux 下的系统调用,用 gcc 编译 C 语言程序,用 gdb 调试程序,用 GNU make 管理和集成程序,用 CVS 对软件版本进行控制,用 lex 和 yacc

进行词法和语法分析生成程序构造,用 autoconf 自动配置程序,用 automake 维护 Makefile,用 wpe 对程序的编译、调试和运行进行管理,用 Kdevelop 开发图形用户界面 GUI,特别介绍了 Qt 和 KDE 的使用。

相信本书对于编程人员会有所裨益。当然, Linux (UNIX) 下的编程工具实在是很多, 一本书的篇幅不可能把这些工具都介绍给大家。本书只是择其重点对一些特别常用的编程开发工具做了介绍。读者可以以此为开端进一步学习和掌握 Linux 下其他的开发工具。掌握的工具越多越精通, 你的编程能力就越强, 效率也就越高。本书编著者将乐意与你共享收获的喜悦。

——孙玉芳

前 言

经过世界各地的计算机爱好者和黑客的不断努力, Linux 已经成长为成熟的操作系统, 它符合 POSIX 标准, 不仅可以运行于 Intel 386 以上处理器的 IBM 兼容计算机, 而且还被移植到了 DEC Alpha 和 SUN 的 SPARC 系列计算机上。

只要遵循 GPL 条例, 任何人都可以免费下载 Linux 源代码, 并可以自由修改和发行。随着 Internet 的发展, Linux 在全世界得到了广泛的传播。Linux 所具有的自由开放的特点, 激发了全世界计算机爱好者的热情, Linux 平台上的应用程序不断丰富, 到现在为止, 几乎所有 UNIX 厂商的应用软件和实用工具都被移植到了 Linux 上, 包括 UNIX 平台上的著名的窗口系统——X Windows。这些软件像 Linux 源程序一样遵循 GPL 条例, 任何人都可以修改这些软件。

由于 Linux 的免费、开放、稳定、高效的特性, 吸引了一批著名的计算机厂商的注意力, 包括 IBM、Intel、SUN 等公司都在推动着 Linux 的发展, 并使 Linux 逐步走向商业应用。

Linux 所倡导的自由软件的开发采用了开放和协作的开发模式, 所有软件产品的源代码是公开的, 这种方式可以充分的利用已存在的软件资源, 集中开发人员的集体智慧, 避免重复劳动, 及时地修改产品中的错误, 增强系统功能和稳定性。为自由软件提供技术支持和增值服务的公司不断出现, 其中的一些公司获得了巨大的成功, 如占 Linux 市场份额最大的软件厂商 RedHat。自从 Linux 传入我国, 经过众多优秀程序员和国内软件商的努力, Linux 的中文文化技术不断进步, 并出现了拥有自主产权的中文 Linux 系统。

红旗 Linux 是由中科红旗软件技术有限公司推出的中文化的 Linux 发行版本, 预装了炎黄中文平台和方正 TrueType 字库, 是目前国内率先支持大字符集(GBK)的中文 Linux 操作系统, 实现了 Linux 上的 TrueType 打印功能, 并且从安装到使用提供了全中文化的操作环境。红旗 Linux 提供了能够满足从个人用户到局域网环境下的数据库网络服务的全套解决方案。但是应该看到, 我国的软件行业还很薄弱, 许多关键技术还掌握在国外厂商手中, 自由软件运动为民族软件业的发展提供了契机, 我们应该探讨如何在跟踪国外最新技术的同时开发自主产权的软件产品。一项迫切的工作是推广基于 Linux 的软件的开发和应用, 针对市场上关于 Linux 平台的软件开发的书籍和资料相对匮乏的现实情况, 我们把这本《红旗 Linux 软件开发技术》奉献给广大读者, 希望为推动 Linux 在我国的发展贡献一点微薄的力量。

本书的读者对象

任何对 Linux 下软件开发感兴趣的人都可以在本书中找到有用的东西,不过完全掌握本书内容需要你有一定的 Linux 使用基础。如果你刚刚接触 Linux,对 Linux 的众多命令感到无所适从,无法完成正常操作,或经常对 Linux 中出现的问题感到莫名其妙,我们推荐你首先阅读本丛书的另外三册:

- 《红旗 Linux 桌面版入门》:介绍红旗 Linux 桌面版的基本安装、配置和使用,使你快速掌握 Linux 的基本操作。
- 《红旗 Linux 常用软件》:介绍红旗 Linux 中的各种办公软件、实用工具、娱乐软件等,使你在 Linux 下得心应手地工作和娱乐。
- 《红旗 Linux 技能荟萃》:解答你在红旗 Linux 可能遇到的各种问题,包括安装、使用、配置中常见注意事项,使你可以在红旗 Linux 下解决各种疑难问题并顺利完成系统管理工作。

另外,你应该具有 Linux 下的一些基本实用工具(如 awk 等)的使用和编程经验,Linux 和 UNIX 一样,其设计目标是满足用户不同层次的要求,系统提供给用户灵活的可编程的 Shell 和大量的实用工具,而用户可以利用把各种工具组合在一起完成复杂的任务。

对 Linux 的 Shell 和实用工具的编程需要专门的技巧,事实上,对 Linux 进行软件开发的一个重要的方面就是熟练掌握这些实用工具并根据实际需要选择合适的实现方法。对这部分内容的介绍可以构成一本完整的书,实际上,任何一本介绍 UNIX 的实用工具和 Shell 编程的书都会提供这方面的帮助,因此本书没有包含这些内容。

本书集中介绍在 Linux 下进行高级语言开发所需要熟悉的系统环境和工具,帮助你快速地从一名 Linux 熟练使用者成长为 Linux 下的软件开发人员。UNIX 从诞生的那天起就和 C 语言有密切的关系,可以说 UNIX 是 C 语言程序的最佳编程平台,而 Linux 继承了 UNIX 的这种特性。本书假设你已对 C 语言有了一定的编程基础,你也可以在任何讲解 C 语言使用和编程技巧的书中获得相应帮助,在第 2 章中将对 C 语言的特点及在 Linux 下进行 C 语言开发的方法进行了概述。

对于熟练的 C 语言程序员来说,本书介绍的 Linux 下的编程环境和工具可以作为开发时的详尽地帮助,并使未接触过 Linux 的程序员快速掌握 Linux 的开发方法。

已熟悉 Linux 开发的程序员通过本书进一步理解和掌握 Linux 的编程环境和工具,并可以把本书内容作为快捷的参考。本书的一个特色是详尽地介绍了在 Linux 下进行 C/C++ 开发的集成环境,并着重讲解了开发 Linux 的 GUI 图形应用的工具及技巧。

随着 Linux 在桌面应用中的不断发展,为 Linux 开发 GUI 图形应用变得越来越重要,由于 Linux 下的窗口图形系统刚刚开始流行,GUI 图形开发的相关资料非常匮乏,相应的开发技巧也不为 Linux 开发人员所熟悉,希望本书能对推广 Linux 的桌面应用程序起到积极的

作用。

本书内容简介

本书几乎涉及了在红旗中文 Linux 下进行高级语言开发的各个方面,从基本的软件设计工具 gcc、gdb、make 到 GUI 集成开发环境 Kdevelop,读者能快速掌握 Linux 下的软件开发技术。本书分为以下三篇。

基础篇:

第 1 章 “揭开 Linux 面纱”,介绍了 Linux 的基础知识和特性。首先介绍了 Linux 的发展历史、发布策略、商业应用状况。使读者对 Linux 有一个整体的理解,在此基础上,着重阐述了 Linux 作为一个优秀的操作系统的特性,包括高度灵活性、多用户、多任务、多处理、充分利用处理器保护模式、动态链接、虚拟内存、支持多文件系统、支持多种执行文件格式、强大的网络功能、可编程 Shell、功能强大的工具集等方面。

优秀的程序员和应用软件开发人员应该对操作系统的机理有较深刻的理解,本章试图从操作系统的设计原理上阐述 Linux 的特色,为读者以后进行系统开发打下基础。

第 2 章 “Linux 下的 C 语言编程”,使读者了解如何在 Linux 下进行 C 语言开发,使读者了解 Linux 下 C 语言开发软件的特点。本章首先回顾了 C 语言与 UNIX 的关系,然后简略地介绍了 C 语言的特性。在此基础上,介绍了 Linux 下的 C 语言开发工具及其特点。最后,介绍了 Linux 下如何开发 X Windows 的 GUI 应用程序。

Linux 软件开发环境及工具篇:

第 3 章 “使用 gcc 编译 C 语言程序”,提供了使用 Linux 的编译程序 gcc 编译 C 语言程序的基本知识。首先介绍了 gcc 基本特性,然后详细的介绍了 gcc 的调用格式、一般参数、优化参数、调试参数等。

第 4 章 “使用 gdb 调试程序”,介绍了 Linux 的 GNU 调试工具 gdb 的使用方法。首先介绍了 gdb 的功能和特点,然后介绍了 gdb 的调用参数和基本使用技巧。最后介绍了 gdb 的高级特性及配置。

第 5 章 “使用 make 管理项目”,介绍了 Linux 下的 make 工具的使用知识。首先介绍了如何管理多文件项目,以及如何把项目进行分解,然后介绍了 makefile 的格式,最后,介绍了 make 的常见命令行选项以及如何对 make 进行调试

第 6 章 “版本控制工具 CVS”,介绍了 Linux 下的程序版本管理系统 CVS 的使用方法。CVS 可以跟踪并管理 Linux 下的软件项目的版本,并可以帮助多个开发者协同工作。首先介绍了 CVS 的基本功能,然后详细描述了 CVS 的使用方法。

第 7 章 “lex 和 yacc”,介绍了使用 Linux 下的词法分析程序生成器 lex 和编译程序生成

器 yacc 的使用技巧。首先介绍了 lex 和 yacc 的基本原理和功能。然后详细介绍了 lex 和 yacc 的使用方法。

第 8 章 “使用 autoconf 开发自配置软件”,介绍了 Linux 下的 autoconf 工具使用方法,autoconf 是一个用于生成可以自动配置软件源代码包以适应多种类 UNIX 系统的 Shell 脚本的工具。首先介绍了 autoconf 的功能原理,然后详细介绍了 autoconf 的使用方法,使程序员可以自动配置自己的软件以适应不同的系统。

第 9 章 “使用 automake 维护 Makefile”,介绍了 Linux 下的 automake 工具,automake 可以使程序员自动生成 Makefile 文件。首先介绍了 automake 的功能,然后详细介绍了 automake 的使用方法。

第 10 章 “使用 wpe 集成开发环境”,介绍了 Linux 下的 C 语言集成开发环境 wpe 的使用方法。wpe 可以完成源程序的编辑、调试、运行。

GUI 应用程序开发篇:

第 11 章 “使用 KDevelop 开发 GUI 程序”,详细介绍了 Linux 下的桌面窗口系统的开发工具 KDevelop 的方法。本章首先介绍了 KDevelop 集成开发环境的基本使用方法和开发图形应用程序的一般原理,接着详细介绍了利用 KDevelop 开发 GUI 应用程序的基本方法和技巧,最后通过一个完整的例子使读者了解 GUI 应用程序的开发方法和步骤。

目 录

第一部分 基础篇

第 1 章 揭开 Linux 的面纱	3
1.1 什么是 Linux	3
1.2 谁创造了 Linux	3
1.3 Linux 与自由软件	4
1.4 Linux 的发行版本	5
1.5 Linux 的商业应用	5
1.6 Linux 正在走向成熟	6
1.7 中文 Linux 发行版本	7
1.8 Linux 的设计特性综述	9
第 2 章 Linux 下的 C 语言编程	18
2.1 C 语言与 UNIX	18
2.2 C 语言特性	19
2.3 Linux 下的 C 语言函数库	20
2.3.1 GNU C 语言函数库 glibc	21
2.3.2 其他的函数库	22
2.3.3 库和前导文件的位置	23
2.3.4 库的操作及配置	23
2.4 Linux 下的系统调用	26
2.5 Linux 下的开发工具简介	27
2.6 获得帮助	28

第二部分 Linux 软件开发环境及工具篇

第 3 章 使用 gcc 编译 C 语言程序	35
3.1 gcc 简介	35
3.2 gcc 的安装与设置	36

3.3	调用 gcc	37
3.4	使用优化参数	43
3.5	使用调试选项	44
3.6	gcc 的自定义符号	45
3.7	链接	45
第 4 章	使用 gdb 调试程序	48
4.1	gdb 的基本使用方法	48
4.2	gdb 的基本命令	57
4.2.1	gdb 的命令特性	58
4.2.2	调用 gdb	59
4.2.3	gdb 运行模式的选择	61
4.2.4	为程序设置断点和观察点	62
4.2.5	在 gdb 中获得帮助及其他信息	63
4.3	gdb 的高级应用	67
4.3.1	变量的作用域	67
4.3.2	函数堆栈操作	68
4.3.3	在 gdb 中访问多个源文件	70
4.3.4	在 gdb 中执行 Shell 命令	71
4.4	其他的工具	71
4.4.1	使用 xxgdb 调试程序	71
4.4.2	使用 cproto 为函数产生原型	73
4.4.3	使用 gprof 分析程序运行特性	75
第 5 章	使用 GNU make 管理多文件项目	76
5.1	管理多文件项目	76
5.1.1	为何使用多文件	76
5.1.2	什么情况下分解项目	77
5.1.3	怎样分解项目	77
5.2	基本的 makefile 结构	78
5.3	makefile 的规则	80
5.4	Makefile 变量	80
5.5	伪目标	83
5.6	makefile 的隐含规则	84
5.7	makefile 中的函数	85

5.8	在 makfile 中自定义模式规则	86
5.9	常用的 make 命令行选项	86
5.10	调试 make	87
第 6 章	版本控制工具 CVS	88
6.1	版本控制简介	88
6.2	版本控制工具 CVS 的使用	89
6.2.1	建立 CVS 服务器	89
6.2.2	建立 CVS 项目	91
6.2.3	定义模块	92
6.2.4	版本的分支与合并	93
第 7 章	lex 和 yacc	103
7.1	简介	103
7.2	lex	104
7.3	yacc	106
第 8 章	使用 autoconf 开发自配置软件	110
8.1	为什么需要 autoconf	110
8.2	创建 configure 脚本	111
8.2.1	编写 configure.in 文件	112
8.2.2	使用 autoscan 脚本创建 configure.in 文件	114
8.2.3	使用 ifnames 列举条件	115
8.2.4	用 Autoconf 创建配置脚本 configure	116
8.3	configure 脚本的初始化和输出文件	117
8.3.1	寻找 configure 的输入文件	117
8.3.2	Makefile 中的替换	119
8.3.3	Makefile 中的预定义输出变量	119
8.3.4	确定软件包的安装目录	121
8.4	autoconf 支持的测试宏	121
8.4.1	程序选择测试	122
8.4.2	检查普通文件和程序	123
8.4.3	检查库文件	125
8.4.4	检查库函数	126
8.4.5	检查前导文件	128

8.4.6	结构测试	132
8.4.7	类型定义检查	133
8.4.8	C 编译程序特征测试	134
8.4.9	系统服务测试	135
8.4.10	UNIX 变种测试	136
8.5	编写新的测试	137
8.5.1	检验声明	137
8.5.2	检验语法	138
8.5.3	检验库	139
8.5.4	检验运行时的特征	140
8.5.5	测试程序应注意的几个问题	140
8.5.6	语言选择	141
8.6	储存测试的结果	142
8.7	编写自己的宏	145
8.7.1	宏定义	146
8.7.2	宏的名称	146
8.7.3	宏的引用	147
8.7.4	宏之间的依赖性	148
8.7.5	建议的顺序	148
8.7.6	处理过时的宏	149
8.8	进行手工配置	149
8.8.1	指定系统的类型	149
8.8.2	获取规范的系统类型	150
8.8.3	系统类型变量	151
8.8.4	使用系统类型	151
8.9	利用 autoconf 进行站点配置	152
8.10	运行 configure 脚本	154
第 9 章	使用 automake 维护 Makefile	157
9.1	理解 automake	157
9.1.1	automake 的工作过程	157
9.1.2	automake 支持的包的种类	158
9.1.3	automake 的严格性	158
9.1.4	automake 的统一命名机制	158
9.1.5	派生变量的命名规则	159

9.2 示例软件包	160
9.2.1 myprog 软件包	160
9.2.2 GNU Hello 软件包	161
9.3 创建 Makefile.in 文件	162
9.4 检查 configure.in 文件	164
9.4.1 配置需求	164
9.4.2 automake 能够识别的其他宏	164
9.4.3 自动生成 aclocal.m4	166
9.4.4 automake 支持的 autoconf 宏	167
9.4.5 编写自己的 aclocal 宏	169
9.5 编写顶层 Makefile.am	169
9.6 创建程序和库	170
9.6.1 创建一个程序	170
9.6.2 创建一个库	171
9.6.3 创建一个共享库	172
9.6.4 创建一个程序时使用的变量	172
9.6.5 对其他语言的支持	173
9.6.6 自动依赖性跟踪	173
9.7 automake 可以创建的其他对象	174
9.7.1 可执行脚本	174
9.7.2 头文件	175
9.7.3 与体系结构无关的数据文件	175
9.7.4 创建源代码	175
9.8 创建程序包文档	175
9.9 软件包的自动安装和删除	177
9.10 软件包需要发布的文件	177
9.11 定制 automake 的行为	178
 第 10 章 wpe 的使用	 181
10.1 wpe 简介	182
10.2 wpe 的安装与运行	185
10.3 wpe 的窗口操作和文件管理	188
10.4 源程序的编辑	192
10.5 程序的编译和运行	197
10.6 程序的调试	201

10.7	项目管理	204
10.8	定制 wpe 环境	211
10.8.1	wpe 颜色设置	211
10.8.2	wpe 编辑特性设置	212
10.8.3	文件管理器设置	213
10.8.4	帮助设定	214
10.8.5	编程环境的设置	215
10.8.6	编译器设定	216
10.9	获得帮助	218
10.9.1	wpe 使用帮助	218
10.9.2	系统命令与函数信息	219
10.9.3	info 文档信息	221

第三部分 GUI 应用程序开发篇

第 11 章	使用 KDevelop 开发 GUI 程序	225
11.1	X-Window 编程概念	228
11.2	Kdevelop 的安装和设置	230
11.2.1	KDevelop 软件包的获得	231
11.2.2	系统需求	231
11.2.3	编译和安装	232
11.2.4	配置 KDevelop 环境	232
11.3	Qt 和 KDE 库介绍	235
11.3.1	一个基于 Qt 的程序	235
11.3.2	察看 Qt 帮助	236
11.3.3	用户交互	237
11.3.4	Qt 的 signal/slot 机制	238
11.3.5	KDE 库简介	239
11.4	KDevelop 简介	240
11.4.1	KDevelop 的工具简介	241
11.4.2	一般的开发过程	243
11.4.3	其他工作	244
11.5	使用类浏览器和文件查看器	245
11.5.1	类浏览器	245
11.5.2	文件查看器	250

11.5.3	项目管理	255
11.5.4	项目的编译和运行	263
11.5.5	定制 KDevelop	265
11.6	使用 KAppWizard 创建应用程序	271
11.6.1	KAppWizard 的调用	271
11.6.2	第一次编译	276
11.6.3	源程序框架	278
11.6.4	项目的其他内容	284
11.7	应用程序视图设计	285
11.7.1	使用 Qt/KDE 库视图	285
11.7.2	创建自己的视图	286
11.8	配置菜单和工具条	287
11.8.1	工作原理	287
11.8.2	增加新的菜单	288
11.8.3	增加工具条按钮	288
11.8.4	配置状态条	289
11.8.5	键盘快捷键配置	289
11.9	构造对话框	289
11.9.1	Qt 和 KDE 控件	289
11.9.2	构建新的对话框	292
11.9.3	在项目中添加对话框	294
11.10	程序排错	296
11.10.1	Qt 程序排错	296
11.10.2	KDE 排错宏	298
11.10.3	使用调试器	298
11.11	一个完整的例子	299
11.11.1	使用 KAppWizard 生成新项目	300
11.11.2	生成对话框	302
11.11.3	添加代码	305