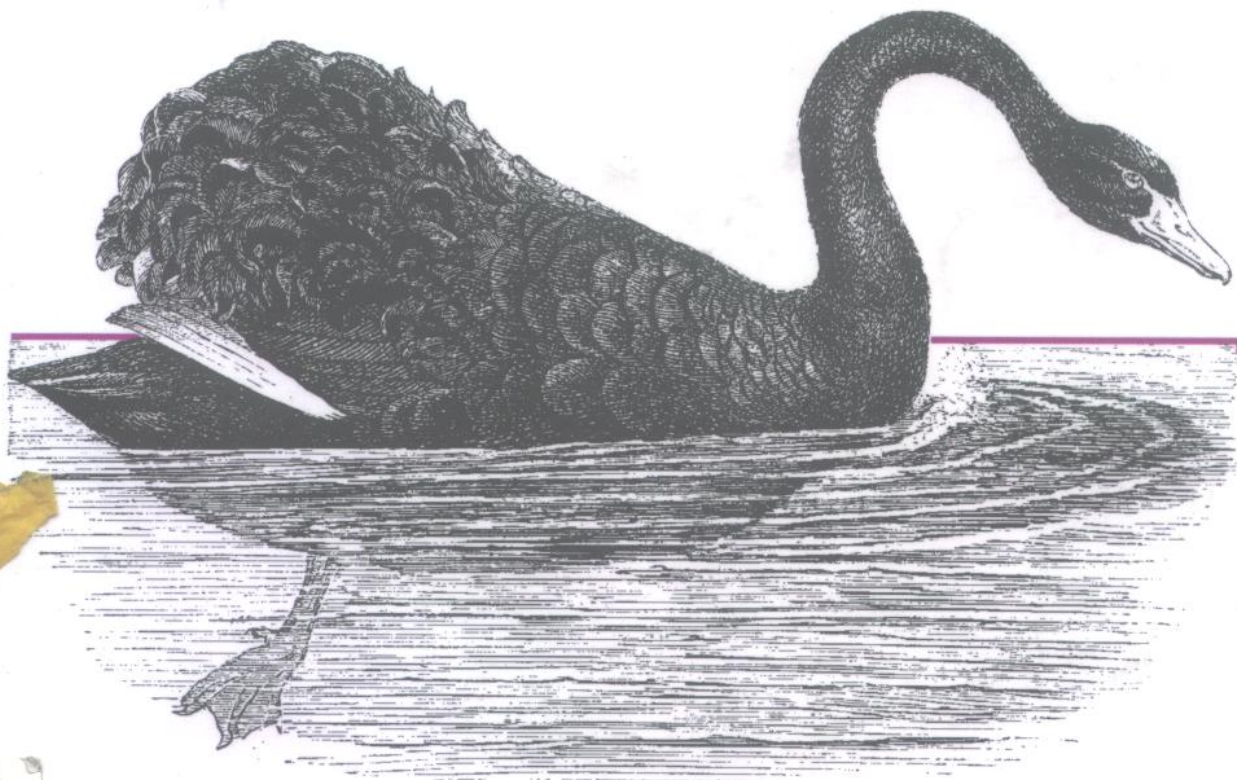


Programming with GNU SoftWare

光盘上包括  
七种GNU免费  
软件

# 用GNU 软件编程



Mike Loukides 著  
Andy Oram

O'REILLY™

石祥生 瞿炯 石秋云 译



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

# 用GNU软件编程

Programming with GNU Software

[美] Mike Loukides 和 Andy Oram 著

石祥生 翟炯 石秋云 译



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

## 内 容 简 介

JS/RS/19

本书介绍了免费软件基金会 (FSF) 和 Cygnus Support 公司提供的 GNU 免费软件。GNU 软件是为 UNIX 操作系统开发的, 其中有些工具已成功地移植到其他操作系统中。

全书共分九章, 即: 免费开发环境, UNIX 操作系统简介, 利用 Emacs 编辑源程序, 利用 gcc 进行编译和链接, 库程序, C 和 C++ 程序的调试, 利用 make 的自动编译, 利用 RCS 管理源文件, 程序的计时和剖析。

本书适合 UNIX 编程人员、大专院校师生参考。

O'REILLY™

Chinese Edition Copyright ©1997 by Publishing House of Electronics Industry. Authorized

Translation of the English Edition Copyright © 1997 by O'Reilly & Associates, Inc.

This translation is published and sold by permission of O'Reilly & Associates, Inc. The owner of all rights to publish and sell the same.

All rights reserved including the right of reproduction in whole or in part in any form.

本书中文专有翻译出版权由美国 O'Reilly & Associates, Inc 公司授予电子工业出版社。未经许可, 不得以任何手段和形式复制或抄袭本书内容。版权所有, 侵权必究。

原 书 名: Programming With GNU Software

书 名: 用 GNU 软件编程

著 者: [美] Mike Loukides 和 Andy Oram

译 者: 石祥生 翟 炯 石秋云

责任编辑: 陆伯雄

印 刷 者: 顺义县天竺颖华印刷厂印刷

出版发行: 电子工业出版社出版、发行

北京市海淀区万寿路173信箱 邮编100036 发行部电话 68214070

URL: <http://www.phei.co.cn>

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 13.25 字数: 300千字

版 次: 1997年11月第1版 1997年11月第1次印刷

印 数: 3500册

书 号: ISBN 7-5053-4397-1  
TP·2031

定 价: 47.00元

著作权合同登记号 图字: 01-97-0992

凡购买电子工业出版社的图书, 如有缺页、倒页、脱页者, 本社发行部负责调换

版权所有·翻印必究

# 关于本书所附光盘

## 如何安装？

执行install.sh脚本文件，并按提示进行操作。例如，如果你安装在/mydrive/cd目录中，那么在提示符下键入/mydrive/cd/install进行安装。详细说明可参见CD-ROM中的read.me文件。

## 光盘上有什么？

在本书所附光盘上有如下程序的二进制文件：

- *GNU Emacs*，一种文本编辑程序。
- *gcc*，一种C和C++的编译程序，它可以立即使自己成为最好的UNIX编译程序，既很强壮，又具有优化功能。
- *GNU库程序*（包括C++库程序）。
- *gdb*调试程序。
- *RCS*，一种备份和维护多个源文件版本的工具。
- *GNU make*，一种管理程序构建的、功能最强的make实用程序版本。
- *gprof*剖析程序（profile）。

该光盘上还包括一整套用于配置、构建和安装这些工具的源程序和脚本程序。

## 适用于哪些平台？

适用的平台包括：

用于Sun SPARC4的SunOS 4.13和 Solaris 2.4

用于HP 9000/700的HP-UX

用于IBM RS/6000的AIX 3

用于SGI Iris/Indigo的Irix 5.3

用于Alpha的Digital UNIX

# 前言

在计算机软件已成为各公司和个人最有利可图的年代里，希望免费共享他人的源程序是不可思议的。然而，令人惊讶的是免费软件基金会（Free Software Foundation）同世界各地的各种组织一起，在创建一个允许通过网络下载软件的编程环境时，却取得了很大的成功。

这一编程环境包括编辑程序、编译程序、源文件控制软件、调试程序、各种其它的实用程序等，这些工具并不是玩具，也不是极不可靠的纸牌搭成的房子，其中有些软件的商品化产品已在市场上参与竞争很多年了，并且已被数百家公司用于它们的生产活动中。Cygnus Support公司和Cyclic Software公司以光盘的形式提供它们的软件，并且已很好地开展了对其软件用户的支持业务。

本书所述的各种工具是为UNIX操作系统开发的，但其中有些工具已成功地移植到其它操作系统中，一开始用于UNIX的很多工具（例如**make**、**lex**、**yacc**等，更不用说C和C++语言）已经成为其它系统的标准。在UNIX社会中，共享和合作精神是最强烈的。

因此，在UNIX中产生一组最综合的免费编程工具是很自然的事情。GNU是免费软件基金会的一个项目，它已经开发了大部分的工具，目前正在继续致力于开发全套的UNIX编程工具。

对于希望在UNIX系统上工作的C语言程序员来说，本书和附带的光盘是一套很完整的起步工具。在该光盘上包含了用于六个主要平台的二进制格式的文件及完整的源程序。本书的出发点认为读者除了知道如何用C语言编程外，对这些软件一无所知，它将引导你深入各种工具，并最终成为一个很有自信心、具有竞争力的用户。因此，即使你的计算机系统已经安装了这些工具（例如，如果你是一位大学生，试图利用大学的系统完成编程工作），你也将会发现本书是很有价值的。

如果你目前还没有UNIX系统，但希望去获取一个这样的系统，那么你的运气很好，因为你可以将完全免费的软件安装在你的计算机中。通过网络和光盘，可以获得两种版本的“Berkeley标准发布(Berkeley Standard Distribution)”软件：FreeBSD和NetBSD，它们可以在各种硬件上运行。要获得广泛采用的Linux系统也是很容易的，目前它在Intel x86硬件上运行，而且正在移植到Alpha和其它平台上。本书所述的大多数工具通常是随同这些系统一起提供的，O'Reilly & Associates出版公司出版了与所有这些系统有关的各种书刊。最后，免费软件基金会正在致力于开发一种类似于UNIX的操作系统，称为HURD。

我们希望本书能为工程师、大学生、准备或已经从事于UNIX系统工作的任何人，打开进入这一功能强大的编程环境的大门，并在该编程环境中充分发挥自己的才能。

## 本书内容的编排

在编排各章节时的出发点是要使读者能循序渐进、轻松自如地进入UNIX系统和使用盘上的各种工具。一开始首先介绍你将使用的一些基本实用程序，如Emacs文本编辑程序和C编译程序；接着介绍你乐于使用的几种较先进的工具，如**make**和RCS。前两章是一般性的论述，并没有直接涉及盘上的软件，后面的章节才具体讨论盘上的每种工具。

第1章，**免费软件开发环境**。说明使用免费软件的好处，为什么要开发免费软件，以及通用公开许可证(General Public License)对使用免费软件有什么限制。

第2章，**UNIX操作系统简介**。如果你不知道UNIX的各种约定和基本命令，本章将为你提供各种必要的基础知识。

第3章，**利用Emacs编辑源程序**。从程序员的角度专门介绍这种令人难以置信的工具，它具有至系统中每种其它工具的接口。

第4章，**利用gcc进行编译和链接**。介绍能使得你的C或C++程序顺利通过编译的一系列实用程序，以及最重要的各种命令选项。

第5章，**库程序**。介绍可以从盘上的库程序中获得的、最普遍使用的各种调用，包括处理出错、信号和时间的调用。

第6章，**C和C++程序的调试**。详细地讨论**gdb**程序(GNU调试程序)。

第7章，**利用make的自动编译**。介绍一种在对付较复杂程序时你乐意使用的工具。如果你已经在Microsoft的系统上工作，可能已经很熟悉**make**，但GNU版本的**make**的功能要强大得多。

第8章，**利用RCS管理源文件**。讨论目前你可能还没有想到要使用的另一种工具，但当你有了一个长远的项目时，该工具是非常关键的。RCS可保存各种文件的各个老的版本，利用这些版本就可重新创建以前的系统，排除掉错误，做一些很有用的其它工具。

第9章，**程序的计时和剖析**。讨论如何测定你的程序所使用的系统资源，其重点是讨论剖析程序**gprof**。

附录A，**Cygnus Support公司简介**。简要地介绍Cygnus Support公司的情况，本书附带的光盘中的大部分软件都是该公司提供的。

附录B，**利用源程序构建GNU软件**。全面而概要地介绍免费软件基金会和Cygnus Support公司使用的配置和构建系统，如果你需要重新编译任何一种工具，这将能帮助你理解需要做些什么操作。然而，本附录的内容并不完整，还应查阅有关的Cygnus资料。

附录C，**数据表达形式**。列出了各种数据类型的典型长度。

附录D，**GNU的通用公开许可证**。介绍该通用公开许可证的全文，该许可证提供了分发GNU软件的各种规定。

附带的光盘中包含了本书介绍的全部软件，在该盘上的read.me文件中介绍了如何去

安装和设置各种工具。

## 如果你遇到了困难

随同本书一起提供的光盘中包含着大量功能非常强的软件，O'Reilly & Associates出版公司将一如既往地支持本书，如果你在理解本书的文本时遇到了困难，我们将会帮助你，并尽量提供有关安装的任何信息，但我们并不准备超出书本的内容去提供如何使用这些工具的专门建议。

如果你需要更多的专门知识，可以去购买深入介绍每一种工具的书籍，我们已出版了论述各种工具的下列书籍：

|             |                                                         |
|-------------|---------------------------------------------------------|
| Emacs       | <i>Learning GNU Emacs</i>                               |
| RCS         | <i>Applying RCS and SCCS</i>                            |
| <b>make</b> | <i>Managing Projects with make</i>                      |
| C语言         | <i>Practical C Programming</i>                          |
| C++语言       | <i>Practical C++ Programming</i> (《实用C++编程大全》电子工业出版社出版) |
|             | <i>C++: The Core Language</i>                           |

Cygnus Support和Cyclic Software两家公司都可以提供完整的技术支持服务，Cygnus Support公司可以为Emacs、C和C++编译程序、**gdb**和**gprof**等软件提供技术帮助，但不支持RCS和**make**。此外，该公司与一些大型的公司或组织建立了长期的合作关系，但并不包括个人的临时要求在内。因此，要求它提供帮助时，可打电话给800-CYGNUS-1或415-903-1400，或者发电子邮件给support@cygnus.com。Cyclic Software公司可以为各种版本的RCS和CVS软件提供一套类似的服务，欲了解其详细情况，请查看它的Web站点<http://www.cyclic.com/>。

如果你在光盘中的某个程序内发现了一个错误，请告诉我们，我们一定将它输入Cygnus的错误数据库中。

如果你在构建或使用软件时遇到了一个较复杂的问题，可是又不准备利用Cygnus Support或Cyclic Software公司的支持服务，那么怎么办呢？幸运的是免费软件有一个长处，很多新闻组可以在用户之间相互提供帮助，这些新闻组是：

|                       |                         |
|-----------------------|-------------------------|
| <i>gnu.emacs.help</i> | 可提供使用GNU Emacs软件的帮助。    |
| <i>gnu.gcc.help</i>   | 可提供使用GNU C函数和编译程序的帮助。   |
| <i>gnu.g++.help</i>   | 可提供使用GNU C++函数及编译程序的帮助。 |

当你提出问题时，在数小时内这些新闻组的人员就会予以解答。

对于每一个“help(帮助)”组，有一个相应的“bug(错误)”组，如果你认为在实用程序中发现了一个真正的错误，就可以进行核对。由于其它的GNU实用程序没有“help”组，有时人们就将问题提交给“bug”组(顾名思义，这些组仅仅服务于报告真正的错误)，例如：

*gnu.gdb.bug* gdb中存在的问题。

*gnu.utils.bug* 所有其它GNU实用程序中存在的问题。

最后，有一个*comp.software.config-mgmt*，它是专门讨论RCS和其它版本的控制工具。

## 本书中的约定

在本书中使用下列约定：

斜体：表示文件和目录名、函数名、程序变量、一般的重点等。在例子中，利用斜体插入注解，这并不是你键入的文本部分。

黑体：表示命令名、选项、键盘的键、用户名、C预处理器命令(如**#if**)等。

等线体：表示编程语言的关键字，例如**int**和**struct**。在例子中，用于表示程序代码、输入或输出文件、命令和程序运行的输出等。

等线黑体：：用于例子中表示你在终端上输入的命令或输入。

等线斜体：用于例子中表示一个命令的类属(可变)部分。在特定情况下，可以用专门的单字替代。例如：

`rm filename`

就表示键入命令rm，后面跟着一个文件名。

\$：用来表示外壳(shell)提示符，不同外壳的缺省外壳提示符是不一样的，可以由用户修改。

本书中使用两种标记法来表示使用的控制键。一种是大多数读者所熟悉的，即**CTRL-X**，你必须按下**Control**键不放，同时键入字符“x”。在讨论Emacs时，我们利用在它的资料中使用的标记法，即**C-x**表示其操作与**CTRL-X**相同。当你与**Control**键一起键入一个字母时，shift键是无关的。

其它键的使用方法是相同的(例如，**Return**表示回车)，所有的用户输入操作结束后，都应跟随一个**Return**。当然，另外说明则除外。

# 目录

## 前言

|                               |           |
|-------------------------------|-----------|
| <b>第1章 免费软件开发环境 .....</b>     | <b>1</b>  |
| UNIX——程序员的舞台 .....            | 2         |
| 什么是免费软件 .....                 | 3         |
| 本书中的软件 .....                  | 8         |
| 二进制程序的安装 .....                | 9         |
| 工具的更新 .....                   | 9         |
| 免费软件的培植 .....                 | 10        |
| <b>第2章 UNIX操作系统简介 .....</b>   | <b>13</b> |
| UNIX的登录 .....                 | 13        |
| UNIX命令 .....                  | 14        |
| UNIX的文件系统 .....               | 16        |
| 联机资料 .....                    | 24        |
| 标准输入和输出 .....                 | 26        |
| 控制程序执行 .....                  | 28        |
| 外壳(shell)程序 .....             | 31        |
| 外壳的设定 .....                   | 37        |
| 其它基本工具 .....                  | 42        |
| <b>第3章 利用Emacs编辑源程序 .....</b> | <b>45</b> |
| Emacs编辑程序入门 .....             | 46        |
| C方式的基本编辑操作 .....              | 48        |
| 在Emacs内编译 .....               | 65        |
| 外壳窗口 .....                    | 67        |
| 标志(Tag) .....                 | 68        |
| 某些其它的方式和命令 .....              | 70        |
| <b>第4章 利用gcc进行编译和链接 .....</b> | <b>73</b> |
| C编译过程 .....                   | 73        |
| C程序的汇编 .....                  | 82        |

|                              |            |
|------------------------------|------------|
| 程序的链接 .....                  | 83         |
| 创建库程序 .....                  | 87         |
| 交叉编译 .....                   | 87         |
| <b>第5章 库程序 .....</b>         | <b>93</b>  |
| 错误处理 .....                   | 93         |
| 信号 .....                     | 95         |
| 时间 .....                     | 98         |
| 系统接口问题 .....                 | 101        |
| C++类 .....                   | 105        |
| 库程序和许可证 .....                | 108        |
| <b>第6章 C和C++程序的调试 .....</b>  | <b>109</b> |
| 为gdb执行编译 .....               | 109        |
| 启动gdb.....                   | 110        |
| 基本的gdb命令 .....               | 111        |
| 变量的作用域和环境 .....              | 127        |
| 调用栈的上移和下移 .....              | 128        |
| C++程序 .....                  | 136        |
| Emacs的接口 .....               | 137        |
| 命令名的补全和缩写 .....              | 138        |
| 连接现有进程 .....                 | 141        |
| 小结 .....                     | 142        |
| <b>第7章 利用make的自动编译 .....</b> | <b>145</b> |
| 创建makefile文件 .....           | 146        |
| make的调用 .....                | 155        |
| make和RCS.....                | 157        |
| 出错消息 .....                   | 157        |
| 补充说明 .....                   | 158        |
| <b>第8章 利用RCS管理源文件 .....</b>  | <b>159</b> |
| 版本树 .....                    | 159        |
| 基本操作 .....                   | 160        |
| 有关签收的其它信息 .....              | 165        |
| 新版本和老版本 .....                | 166        |
| 其它功能 .....                   | 167        |

|                                     |            |
|-------------------------------------|------------|
| 补充说明 .....                          | 170        |
| <b>第9章 程序的计时和剖析 .....</b>           | <b>171</b> |
| 简单计时 .....                          | 172        |
| 剖析简介 .....                          | 172        |
| 利用gprof生成剖析文件 .....                 | 173        |
| <b>附录A Cygnus Support公司简介 .....</b> | <b>181</b> |
| <b>附录B 利用源程序构建GNU软件 .....</b>       | <b>183</b> |
| <b>附录C 数据表达形式 .....</b>             | <b>191</b> |
| <b>附录D GNU的通用公开许可证 .....</b>        | <b>195</b> |

## 本章内容

- UNIX——程序员的舞台
- 什么是免费软件
- 本书中的软件
- 二进制程序的安装
- 工具的更新
- 免费软件的培植

# 第1章

## 免费软件开发环境

长期以来，UNIX一直以它的软件开发环境而著称，大量第一流的创新工具是UNIX在市场上取得成功的众多原因之一。

然而，对于一些好的工具，用户们感到有些力不能及，这并不是说UNIX厂家没有做这方面的工作，而是因为几乎所有的UNIX厂家都将开发工具从操作系统本身中分离出来。因此，如果你要做任何编程工作，就要再花费600至1000美元，这一金额对一个大公司来说不算什么，但对于一名学生或一个家庭用户却是一笔很大的开支。

本书介绍了一组软件开发工具，它们与相应的UNIX工具同样出名，可以提供相同的（或更强的）功能，而且与UNIX用户以前使用的各种工具高度兼容。这类工具就是免费软件基金会（Free Software Foundation,FSF）提供的编译程序和相关的开发工具，通常称为GNU工具。

这些工具与UNIX工具不同，它们是“免费的”，但它们的精神实质与安装在大学校园内的早期UNIX系统是相同的。这类工具可以不费分文地从很多地方得到，包括其源程序在内，而且鼓励用户将这些工具提供给你的朋友。你不必为一个“开发系统”向UNIX厂家支付费用，就可从FTP或UUCP的档案库中得到GNU工具的源程序<sup>①</sup>，这样做可能会占用你的电话线路几个小时，但除了电话费外，一切都是免费的。花几十块钱，你就可以买到本书及附带的光盘，从而获得一个完整的、经过预编译的开发环境。从其它厂家也可以获取类似的经过预编译的软件包。

利用光盘分发软件具有很多优点，其中最重要的是软件已在几种最广泛使用的计算机和操作系统上构建和测试过，除此以外，还能节省你的大量时间。虽然我们也可从网络上取得软件包，编辑几个配置文件，键入**make**，好像一切工作都完成了，但实际情况却并非如此简单。

利用光盘分发软件还可以解决恼人的“鸡与蛋”问题。如果你得到了免费软件的源程序，就需要一个编译程序（和相关的实用程序）去构建可执行程序；如果你已具有该厂家的开发系统，就可以立即开始工作；但如果你没有，那么就无法将数兆字节的免费源程序加载到磁盘上。通过提供适用于几种常用体系结构的预先构建好的软件，光盘就可以使你摆脱这种制约。

---

<sup>①</sup> 遗憾的是操作系统有时要求一些GNU并不支持的功能，例如，如果你目前使用的是SGI的Irix系统，为了得到某些关键的库程序，就需要购买SGI的开发系统。

---

本书为相当传统的用户提供了GNU软件开发环境的指南，我们将讨论这些工具是什么、如何使用它们和如何将它们装配在一起。我们并不深入探讨它们的奥秘，只是使你能熟悉和适应这些工具最经常的用法。因此，虽然我们将花一些时间讨论“gcc -o”（以最优优化方式编译C程序），但却不去探讨“gcc -mno-half-pic”。

首先，我们将讨论免费软件是什么，它存在的理由是什么，对于软件开发人员来说它意味着什么。

## UNIX——程序员的舞台

就工具的完善程度和灵活性而言，UNIX可以与市场上的任何编程环境相媲美，目前在整个计算机领域内广泛使用的很多工具都起源于UNIX，只需看一看C语言就知道这一点。

但是，在开始使用各种UNIX工具之前，必须克服一个障碍。这就是使用这些快速和高效的工具需要你学习一些非常简洁的命令，甚至敲键操作（在Emacs的情况下）。很多人力图避免使用UNIX，是因为几乎没有一种它的工具能支持图形或“指点-点按”界面，不过在使用该系统一段时间后，这些命令和敲键操作就会快于“指点”和“点按”。更重要的是你可以组合各种命令和使它们自动执行。此外，不要忘掉成本优势，你可以在一台连接串行线的廉价终端上使用这些工具。

因此，如果你知道打算做什么，本书中的种种工具将能帮助你去完成你的作业，其速度比图形用户界面要快很多。作为一个程序员，你所需要的是一个功能强大的文本界面和肯定能提供这种功能的各种工具。当然，我们也已经可以看到向着图形界面发展的某种迹象，例如，目前Emacs已能支持菜单和鼠标器。

集成是本书中各种免费UNIX工具的另一强项。例如，在Emacs编辑程序中，在修改程序的同时可以发出一个编译命令。为了构建你的程序，该编译命令将运行**make**实用程序。如果其源程序存储在RCS源文件控制系统中，**make**可以自动将它取出来。在我们讨论这些工具的整个过程中，你将会看到为了方便起见而使它们组合在一起的各种方法。

## 强大的功能体现在何处

前面已经说过，免费软件工具是命令启动的，某些命令的功能也是十分强大的。例如，在**gdb**调试程序中，**print**命令可以显示任何一种变量，可以是一个整数标量，可以是一个串的数组，也可以是一个具有不同数据类型数字的复杂数据结构。

因此，在学习的第一阶段是找到一组可以完成你的大部分任务的命令，接着是将它们组合在一起。例如，Emacs允许你将若干个敲键操作组合成一个包，然后用两个敲键动作就可去执行它，这就是一个“宏”。你也可以将各种命令放在启动文件中，每当启动Emacs时就会执行这些命令。**gdb**具有类似的初始化文件，以进行调试会话。

---

## 本书的作用

如果你是一个学生，恰好在UNIX系统上学习编程；或者你是另一个环境的程序员，有一个项目要在UNIX系统上完成，本书将能使你自如地使用UNIX和它的各种工具。在本书中并不讨论C或C++语言，但是将会告诉你如何去构建、运行和调试程序。

如果你是一个较有经验的UNIX用户，需要一组功能强大的工具，本书和附带的光盘将能为你提供一个完整的编程套件，你可以跳过第2章（UNIX操作系统简介）。但是即使是一个了解UNIX的程序员，也会在本书的其它章节中得到非常有用的启发。

当你开始使用一个新的操作系统时，常常会感到好像需要同时学习所有的内容，本书却采用另一种方法，即明智地循序渐进地进行学习。

首先，在简要地介绍UNIX以后，你将学习文本编辑程序；一旦你学会编辑程序，就开始学习编译程序和某些很有用的C和C++库调用；接着学习一些可选用的工具，例如调试程序和源文件控制系统。

当然，一书中不可能包罗每一种工具的全部内容，但学习了本书后，你就不必经常去查阅其它的资料，你所需内容的百分之九十五应该包含在本书中。当我们讨论某些工具时，将会列出一些可用的其它资料，你可以从光盘上联机安装这些资料，也可以从免费软件基金会订购其印刷品资料。

## 什么是免费软件

自从计算机问世以来，软件开发人员一直在出售他们所宠爱的程序。而对程序员来说，特别是在大学里，它们总是感到应免费地与他人共享其工具；许多计算机厂家也常常积极鼓励它们的用户去共享其软件，例如，DEC的用户群（DECUS）具有一个各成员提供的免费软件库。

当然，随着网络的普及（在大学里是80年代初，而其它用户则在90年代初），共享软件就变得非常容易。当你编写了一些你认为是非常有用的东西时，就可以很容易地通过电子函件发送给你的朋友，或者发送给某个FTP档案库或电子公告牌（在DOS情况下），供其他人选用。你可以“免费地”取得软件，修改它，增添你自己的功能，并将新的版本提供给朋友们。

因而，总是有大量的免费软件来回传送，这一类可以免费获取的程序，一般情况下并不是那种大型的重要程序，其商业价值也不大，但这并不是说这些软件是无用的。我们无法想像某个人销售像**sps**这样的程序（一种较高版本的UNIX **ps**命令），但你也不能说**sps**是没有用的。在80年代初，开始出现了一些具有重要商业价值的免费软件，到了80年代末和90年代初这已成为很普遍的一种倾向。整个操作系统（例如Linux，一种与UNIX兼容的操作系统，目前非常普遍地用于Intel x86硬件上）、编译程序（如**gcc**和**g++**）、解释性的编程语言（例如Perl和Tcl）、编辑程序（如Emacs）以及其它一些主要的工具等，现在都已经可以免费获得了。

---

## 免费软件的种类

免费软件可以分成三类，在各类之间的界线不是十分明确，但总的说来，大多数软件都可以归纳在下面的三类中：

**公用领域软件（Public domain software）** 某些软件很明显地属于公用领域，也就是说任何人都可以免费地利用这种软件去做想做的任何事情，对这一类软件没有任何限制。

**免费分发的软件（Freely distributable software）** 大量的免费软件，特别是来自各个大学的软件，都是开发它的个人或组织“拥有的”。由Berkeley开发的BSD UNIX部分就属于这一类，X Window系统和许多小型程序也属于这一类。在这种环境中，“拥有权”意味着人们要将这类软件作为另一个项目的一部分再次分发时，必须取得原始拥有者的同意；同时还意味着，虽然你可以按照你的需要修改该软件，但在未经许可的情况下不能使用原来的软件名去分发已修改的软件。也就是说，如果你修改了BSD UNIX，没有“拥有者（加州大学）”的许可，你不能将修改过的BSD UNIX的拷贝提供给朋友们。

**通用公开许可证（GPL）软件（Genral Public license（GPL）software）** 这是指那些受免费软件基金会的通用公开许可证（也称为“无版权（copyleft）”）管辖的软件，在附录D“GNU的通用公开许可证”中介绍了这一许可证。GPL免费软件是免费的，也就是说，确保获取该软件的任何人可以继续利用相同的名字分发它，而且与如何获得该软件和如何修改该软件无关。关于这个问题还将作详细说明。

顺便说一说，某些人认为GPL代表“GNU Public License（GNU公开许可证）”，这当然是由提倡GNU的人们提出来的。但是，GPL还用于大量的其它软件，例如RCS和Linux操作系统，因此GPL的G代表“通用（General）”比GNU更确切。

在本书中，术语“免费软件”一般是指“GPL或某些类似的术语所许可的软件”，但很多人却使用这一术语去包括所有三类软件。

花一些时间去理解为什么必须使用GPL是值得的。前两种免费软件在目前可以保证其源程序是免费的，即你可以得到程序并修改它。然而，却无法阻止一种非常普遍的情况，即它们无法防止一种商业投机行为（比如某个计算机公司），它可以从网上取得免费软件，进行较大的修改，然后对它采用具有限制性的许可证，例如不允许你查看源程序，要想使用该软件就必须支付可观的年许可证费等。

如果你认为这种情况似乎有点牵强附会，那么看一看大多数UNIX系统的联网软件就清楚了。大多数的UNIX网络软件都是以BSD UNIX为基础的，这一程序属于“免费分发的软件”，你可以从网络上取得原始的BSD网络程序，根据你的需要作任何修改。但是UNIX厂家会将这一软件改进后的源程序提供给你吗？一般是不会的，除非你愿意支付一大

---

笔钱给它们，这在任何情况下你都是不会愿意的。随着时间推移，大量的软件已经改变了公用的性质，其原始程序通常仍然可以获得，但你具体需要的一些程序却是商品化的产品，根本就不是免费的。

免费软件基金会（FSF）创建于80年代中期，其目的是推动开发永远是免费的软件。当某些公司选用最好的免费软件和进行改进后，不再允许该软件流通时，FSF的通用公开许可证就可确保防止免费软件的失踪。它专门申明，使用该软件的任何人有权重新分发它，包括其源程序在内，修改后的软件也必须是可重新分发的。实际上，这就意味着你不能将一个具有制约能力的许可证用于修改后的免费程序<sup>①</sup>。特别是在GPL中申明，这类软件必须是永远可以分发的，包括其源程序在内，或者采取一种折衷的方法，即别人请求时应提供其源程序。

GPL（或者它的一个稍有限制性的版本LGPL）适用于免费软件基金会创建的所有软件，也适用于并非由免费软件基金会创建和分发的大量软件。

本书中讨论的软件和包含在光盘上的软件都在GPL的管辖之下，其中的大多数软件是免费软件基金会开发的，或者是与它们有关的。

### 免费但并不廉价

有一种错误的概念，认为“免费软件”就是“不要花钱的软件”，实际情况并非如此。没有任何东西可以阻止某个人销售含有免费软件的盘片，甚至当你通过FTP从他的网络中获取该软件时，也不能拒绝要求你支付费用。事实上，通用公开许可证（GPL）明确地表明销售该软件是允许的，但是你不能限制别人再次销售你卖给他的软件。例如，本书附带的光盘就不是免费的，你也可以从免费软件基金会、Cygnus Support公司、Cyclic Software公司和其它公司购买这些软件的拷贝。

使用“免费”这一个词具有政治色彩，它实际上是指“从许可证限制下解放出来”。这是免费软件基金会意图的关键之处，即创建一整套可以无拘束地流通的软件（一个操作系统加上全部必要的实用程序）。

### 免费但并非低质量

免费软件并不适用于俗语所说“便宜无好货”。的确，大量的公用领域软件和共享软件有很多错误，而且保管不善，但从整体上说，免费软件基金会分发的软件属于保管得很好、不断更新的软件，其结果，这类软件往往优于它的最好的商品化产品。在第4章中将讨论gcc和g++编译程序，它们受到了广泛的称赞，它们所产生的程序好于大多数厂家的编译程序所产生的程序。当在任何软件中发现错误时，很快就能予以纠正。免费软件基金会对发现错误报告的响应速度使得任何商品化软件厂家望尘莫及。

对于免费软件经常谈论的另一个问题是安全性，认为免费软件充满了特洛伊木马、病

---

<sup>①</sup> 这并不意味着你不想分发修改后的版本时却要强制你去分发，你可以自由地保留修改后的版本，在很多情况下这或许是一个好主意。但如果你决定分发你的改进后的版本，就必须使用与原始软件相同的名字。

---

毒和其它有害的东西，你只要使用它就会使系统立即瘫痪。对某些公用领域软件可能的确如此，但这种现象是非常稀少的，至少在UNIX中是如此，甚至还没有听说过<sup>①</sup>。

只有依靠你自己的小心谨慎来对付真正的公用领域软件和共享软件，但由免费软件基金会分发的软件是完全值得信赖的。就像任何软件一样，它可能有错误和安全漏洞，任何一个厂家无论它们作出任何担保，都不能保证其软件是“无错误”的。不过对于免费软件，你可以相信这样的担保，即不会故意引入一些有害的东西来伤害你。

最后，某些用户认为免费软件是没有技术支持的。前面已经说过，如果商品化软件的厂家能像免费软件基金会那样快速响应故障报告，那么我们这个世界就会变成天堂了；然而，这并不是说就不需要传统的客户支持服务去提供纠正错误所必须的一切。Cygnus Support和其它一些公司可以有偿提供这种支持服务，Cygnus Support公司提供的技术支持与任何商品化产品所提供的支持是相同的。我们承认，在免费软件界有些人认为：“支持是不必要的，你已经具有源程序，因而你自己就可以解决问题。”然而，如果你想得到支持，就不要听他们的。

### 免费但在商品化应用程序中却很有用

对免费软件和商品化软件之间的关系，存在着几种不正确的理解，从而阻碍了它们放在一起使用。最普遍的一种误解（实际上与本书内容无关）是认为GPL控制的软件不能与商品化产品一起分销，这恰恰不是真实的，通用公开许可证要求：

- 免费软件可以带着源程序分发，或者在要求时提供其源程序。
- 对免费软件的使用和重新分发没有附加的限制。

另一种较普遍的误解认为与GNU工具“有关的”程序就变成了“免费软件”，就不能作为商品销售，即GPL要求gcc编译程序的输出应受到同一个许可证的限制。再一次需要强调这也不是真的，虽然免费软件基金会希望gcc的用户意识到有一种责任感，使得他们的软件能以不同的方式免费获取，但其许可证并没有强制你这么去做，只有当你采用了免费软件的源程序，才必须使你的程序免费地获取。

还有几个造成这种误解的边缘区。由于你不能将一个典型的许可证用于受GPL管辖的程序，所以也就不能将“免费”程序用于商品化许可的产品中，这就意味着库程序成为一个特殊的问题。采用“免费”软件库（如libg++或GNU版本的libc）中的程序所创建的程序，必须服从GPL的规定，因此在分发该程序时，还要分发其库程序。

解决这一问题有几种方法，最明显的是当你编译时不要求使用GNU库，你可以使用商品化的C或C++库。然而，如果你要求使用“免费”库，免费软件基金会就认为对使用库程序的限制将会阻止人们去使用它们的软件，那种没有人意识到能免费使用的免费软件对它们是无助的。为此，该基金会为库程序创建了一个专用的GPL版本，即LGPL。LGPL

---

<sup>①</sup> 即使在DOS和Mac系统中，也充满着各种病毒，但现已表明，“免费软件”实际比商品化软件安全。当然，我们并不为此作担保。