Wrox程序员参考系列

GTK+/GNOME程序设计

（英） Peter Wright 著

钟 鸣 石永平 等译

机械工业出版社
China Machine Press

本书详细介绍了两种Linux图形用户界面开发技术：GTK+/GNOME。主要内容包括：GUI的基本概念，GLib，GTK+及GNOME的简介，各种GUI元素的创建与使用，gIDE，Glade等等。最后通过两个综合实例来巩固所学知识。本书实例丰富，理论联系实际，是一本实践性很强的编程参考书。本书虽然适合GTK+和GNOME编程的初学者学习，但不管是程序设计新手还是编程专家，都能从本书中获益。

本书版权登记号：图字：01-2000-4095

图书在版编目（CIP）数据

GTK+/GNOME程序设计/（英）莱特（Wright, P.）著；钟鸣等译. -北京：机械工业出版社，2002.1
（Wrox程序员参考系列）
书名原文：Beginning GTK/GNOME Programming
ISBN 7-111-09327-5

Ⅰ.G… Ⅱ.①莱… ②钟… Ⅲ.Linux操作系统-程序设计 Ⅳ.TP316.81

中国版本图书馆CIP数据核字（2001）第065335号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：谢君英 张鸿斌
北京忠信诚印刷厂印刷·新华书店北京发行所发行
2002年1月第1版第1次印刷
787mm×1092mm1/16·31印张
印数：0 001-4 000册
定价：49.00元

译 者 序

本书是一本介绍Linux图形用户界面程序设计的初学者指南。它介绍了两种Linux图形用户界面开发技术，即GTK+和GNOME。GTK+/GNOME是功能强大的自由软件，利用GTK+/GNOME和其他的程序库，可以在Linux上编写出很高级的图形界面的应用程序。

本书是为在Linux中进行图形用户界面程序设计的人员编写的。书中介绍了基本概念，详细描述了GTK+和GNOME的各种实用工具，不论是初学者还是编程专家都可以利用本书快速掌握GTK+和GNOME的开发。

本书还通过许多有趣而实用的例子讨论了图形用户界面应用程序的开发思路以及各种工具和函数的实际使用。学完本书后你完全可以在Linux上自己编写漂亮的图形用户界面应用程序，因此这是一本既适合初学者又适用于专业程序设计人员的不可多得的好书。

参加本书翻译的人员有：钟鸣、石永平、王君、张文、魏允韬、郝玉洁、田晓涛、耿娜、何江华、梅刚、谢卫锋、李晓军、王联华、罗光磊、伊向群、孙登峰、樊伟、何粼、赵彦萍。全书由刘晓霞同志审校。

由于译者水平有限，难免有错误或不当之处，敬请读者批评指正。

2001年7月

前　言

欢迎使用本书。这是一本易于使用的关于Linux GUI程序设计指南。

虽然不可否认Linux作为一个操作系统功能很强，但人们过去常常指责它在用户界面友好方面有所欠缺。不过，如果说这种指责在过去还有点道理的话，现在已经不再成立了。随着强有力的便于使用的GUI（图形用户界面）程序设计工具的不断增加，Linux已经具备了与大多数流行的桌面操作系统一争高低的能力。本书介绍两种Linux GUI开发技术——GTK+和GNOME。

利用GTK+/GNOME和其他的程序库，Linux程序设计员可编写出既适合有经验的用户又适合初学者的GUI应用程序。GTK+/GNOME程序库不仅功能强大，而且还是自由软件。本书的目的旨在帮助读者赶上GTK+/GNOME的发展。在你读完本书后，就能够为Linux应用程序编写非常高级漂亮的GUI了。

本书读者

本书是为对在Linux中进行GUI程序设计感兴趣的人员编写的。不管你是程序设计新手还是编程专家，为了掌握GTK+/GNOME技能，使用本书是明智的选择。本书详细描述了GTK+和GNOME的各种实用工具以及大量实用的、说明性很强的例子，可以利用本书快速掌握GTK+和GNOME的开发。

本书内容

本书介绍了GTK+和GNOME所提供的各种程序设计实用工具。这是一本通篇都是例子、将理论应用于实践的实用教程。

我们从与GUI有关的一般概念（如X Window System、窗口管理器及桌面管理器等）开始介绍。然后说明为什么我们会认为进行应用程序开发采用GTK+/GNOME比采用其他应用程序好。书中还简要地介绍了各种GTK+/GNOME库。

第2章概述了GLib，它为GTK+和GNOME库提供了基本的数据管理功能。该章介绍了GLib的数据类型，以及宏、内存管理、表和树，还介绍了GLib提供的错误处理函数。

第3章开始对GTK+的深入介绍，首先从介绍这个面向对象的开发库着手。这一章中将学习怎样编写和编译第一个GTK+程序并理解信号和事件。

第4章论述怎样控制用户界面。包括控制窗口的标题、尺寸和位置、给窗口添加按钮以及建立包装框和表，等等。这一章中要掌握GTK+中的容器概念。

第5章进一步探索GTK+窗口小部件的奇妙世界。掌握怎样给窗口小部件加标题并向用户提供一种将信息输入的手段。这一章还说明了怎样为窗口小部件建立触发器、复选框、单选钮以及帧。

第6章介绍怎样显示项目列表，使用户能选择列表中的一项或多项并判断出用户选择的是哪

些数据项。本章还处理组合框的显示和配置。

第7章集中处理GTK+中的对话框，包括建立和使用内建的对话框，如建立和使用文件和颜色选择对话框。这一章还说明怎样构造一个对话样式窗口以及怎样模式地显示它。

第8章研究菜单，介绍怎样建立菜单、在菜单中建立复选和单选项并添加快捷键。在这一章中还要学习上下文菜单或弹出式菜单。

第9章介绍一些更为高级的窗口小部件。介绍怎样给窗口添加工具提示、状态栏、工具栏以及进度栏。这一章还要介绍怎样使用调节按钮和建立日历。然后介绍水平和垂直滚动条。

GIMP Drawing KIT（GDK）是专门为功能很强的图像处理程序包GIMP开发的。第10章将介绍怎样利用灵巧的GDK功能来更好地构造GTK+应用程序。我们将了解什么是样式以及样式怎样影响窗口小部件，考虑怎样使用字体和颜色对话框。该章还要学习怎样利用GDK处理图像和图形。

了解了开发库GTK+的强大功能后，将进一步介绍功能更强的GNOME。第11章介绍GNU的"源树"，它是安排GNOME应用程序的文件和目录的标准。接着介绍GNOME为文本定位提供的宏以及构成GNOME的库。这时会看到利用GNOME建立窗口、菜单工具栏和状态栏是何等的容易。

第12章讲述GNOME对话框。首先介绍怎样在GNOME中方便快速地建立和定制对话框。我们介绍了GNOME给程序设计员提供的标准预配置对话框。说明怎样在GNOME中建立About、消息和属性框。

第13章了解GNOME为特殊用户和普通用户提供的窗口小部件，这些窗口小部件包括GnomeCalculator、GnomeNumberEntry、GnomeColorPicker、GnomeFontPicker、GnomeDateEdit、GnomeEntry、GnomeFileEntry和GnomeIconEntry等。该章还对GnomeDruid和GNOME开发人员可用来建立向导的窗口小部件进行了详细的研究。

第14章给出了libgnome库中的更为常见和有用的函数，并说明怎样使用它们。我们将介绍什么是配置文件以及怎样使用它们。该章还介绍怎样在GNOME应用程序中使用声音，怎样给应用程序添加联机帮助并方便、快速地完成一些简单的Internet功能。

第15章将介绍一个名为GnomeCanvas的功能很强的、可扩展的、面向对象的窗口小部件。该章介绍了两种类型的GnomeCanvas（GDK模式和平滑模式），并说明了GnomeCanvas的坐标和组合。我们将掌握画布项的八种现成的类型以及怎样给它添加事件处理程序。最后，简要地浏览一下仿射变换。

一点也不奇怪，GNOME具有自己的集成开发环境（gIDE）。第16章讨论这个功能强大的开发工具，我们从获得并安装它入手，详细介绍它的各种功能。

第17章详细介绍另一个功能强大的GNOME开发工具——项目生成器Glade。这一章的第二部分为一个实用指南，介绍怎样利用Glade建立一个计算器。

本书最后两章给出两个应用程序，利用它们解释开发GNOME应用程序可能出现的问题以及怎样解决这些问题。第一个应用程序为一个图像浏览程序；第二个应用程序（球和弹簧）利用了GnomeCanvas。

本书后面的附录包含了有用的参考资料，最后两个附录给出第18、19章中两个应用程序的代码。

使用本书的要求

需要一套Linux/UNIX，并带有由GTK+和GNOME组成的完整的程序包（库、核心系统等）。几乎所有主要的Linux分发包都带有运行GNOME所需的内容。缺省时，它们一般都会安装GNOME，不过GNOME有可能不是缺省的活动桌面。有的章节和例子要求在系统上安装某些别的程序包（如gIDE和Glade等）和库。获得这些程序包并安装它们的信息会在本书中相应的地方给出。

为保持GTK+/GNOME及有关的程序包和库的版本为最新版本，要求能够进行Internet链接。还需要使用Internet下载本书中的源代码。

我们假定读者具有应用Linux/UNIX的知识，并具有一定的C程序设计语言知识。本书的目的是帮助读者学习使用GTK+和GNOME进行编程。

源代码

我们尽量提供了能够说明各章所介绍的概念的例子和代码片段。本书中完整的源代码可从下面的站点下载：

http://www.wrox.com

这个源代码可在GNU Public License（GNU公共许可证）的条件下得到。虽然本书中给出了几乎所有必需的代码，但建议读者保存一个拷贝以节省代码录入时间。

请将你的想法告诉我们

为了使本书尽可能对你有用，我们已经尽了力，因此很想知道你对本书的看法。我们渴望知道你想要什么和需要知道什么。

谢谢你对我们工作所提出的意见，并请你对我们未来的工作提出批评和赞扬。如果你有话要说，请让我们知道。我们的电子邮件地址或站点为：

feedback@wrox.com

http://www.wrox.com

本书英文版原书书名：Beginning GTK+/GNOME Programming

英文原书书号：ISBN 1-861003-81-1

目　录

第1章 GTK+/GNOME概述

为什么要学习利用某种新语言或技术编写代码呢？你可能会说，“为了编写应用程序”，其实，根本上是由于这种新技术提供了一些以前所没有的东西。如果要学习的是一种新语言，有可能该语言提供了新的数据库或面向对象的程序设计功能。如果要学习的是一种新技术，可能这是一种使你能对某种新平台进行开发的简单技术。

GTK+/GNOME属于后一种情形，它们提供了使你能够方便地开发Linux应用程序的技术。但是，正如可从本书中看到的那样，还有许多学习GTK+/GNOME并利用它们进行开发的理由。不知你是否意识到GTK+已成为一种功能非常强的跨平台开发技术。它目前可用于Windows、Linux以及许多Unix和类Unix风格的操作系统。虽然GTK+使开发GUI很方便和有趣（有时候），但GNOME提供了一组非常高级的函数使这种开发更上一个层次，这组函数使你能够在很短的时间内建立功能完整强大的用户界面。当然，从财务费用方面来看，用GTK+/GNOME进行开发的另一理由是它们是免费的。我们将简要地讨论一下这句话的意思，并说明为什么这对程序设计人员很重要。

本章将做一些简要的回顾。对GTK+和GNOME的世界有一个良好的理解非常重要，我们要理解对开发人员及自由软件来说它们有何意义、从数据库的角度来说它们由什么东西组成以及利用这些技术实际上可以开发出什么样的应用程序等。

那么，请泡上一杯咖啡，找张椅子坐好，从这里开始，你的开发应用程序的方式将要产生巨大的变革了。

1.1 X Window System

没有比较就没有鉴别，不将一事物与它事物进行比较就不能判断出该事物的好坏。GTK+/GNOME起先是为了简化建立基于X Window System的应用程序而研制的。因此，有必要先考察一下什么是X Window System，以及它的来历。

X（X Window System一般只称为X）是麻省理工学院于1984年发表的。麻省理工学院计算机实验室的目标是要建立一种窗口化的图形用户环境，这种环境应该既能够在本地单机上工作，也允许客户机利用单台主机提供这种服务。目前，X仍然能够工作。它运行在客户机/服务器模型上，其中的主机提供客户机应用程序（位于主机或远程终端上）可利用的通用图形服务。

目前，Linux与X的某个版本一起发行，没有许可限制，分发费用也不高；其名称恰如其分地称为XFree86。Free表示可以免费得到且对什么用户都可以使用它这个事实。86是指XFree86是在Intel x86系列处理器（8086、286、386、486、奔腾以上的处理器）上研制运行的。

然而，XFree86并不是一个像Windows、MacOS或BeOS那样典型的图形用户环境，简单地将它安装在一台机器上并不代表就可以利用它完成任何事情。X所完成的工作只不过是提供应用程序可用来在屏幕上绘图的服务以及监视用户利用键盘和鼠标所做的事情。它不提供自己的用

户界面，也没有运行应用程序、定位计算机硬盘的手段，可以说它不提供用户所期盼的任意一种美妙的功能。这些功能全都属于窗口管理器的范围。

窗口管理器本质上是用户与X之间的一个接口。它处理窗口上的图形，通常要控制一个正在运行的应用程序有关的事项。此外，大多数窗口管理器提供某种虚拟桌面系统，用户实际上可在其中得到二个、三个、四个或更多的X显示，并利用鼠标简单的点击每个显示，并可在每个显示上的应用程序之间进行切换。

例如，在图1-1所示的Window Maker（Window Maker是一种常见的窗口管理器）屏幕中，屏幕左上角的图标能够快速访问窗口管理器提供的不同虚拟工作区。窗口管理器也能向用户提供某种实际运行软件的机制。在Window Maker中，屏幕右边的图标提供公共应用程序的快速访问。此外，右击鼠标将显示一个如图1-2所示的菜单系统，此菜单系统能够提供更多的功能。

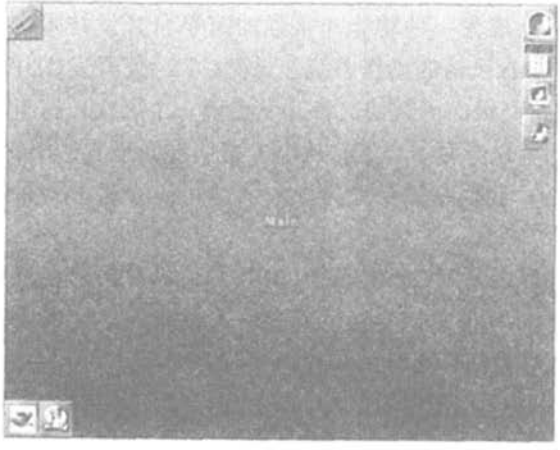

图1-1 Window Maker（一种常见的窗口管理器）

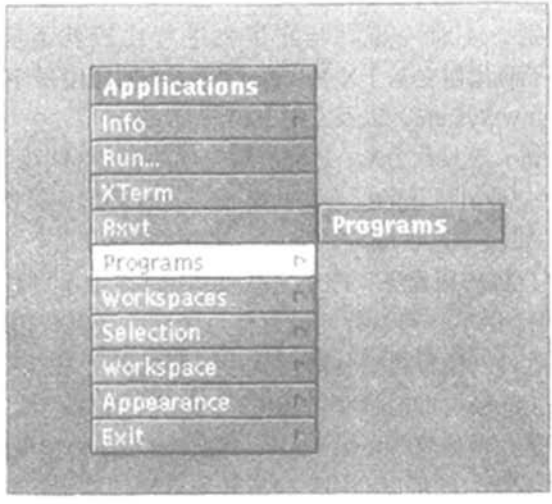

图1-2 关联菜单

总之，窗口管理器（如Enlightenment、Window Maker、FVWM等等）使X更好用、界面更友好。

1.2 桌面管理器

GNOME实际上是一个首字母缩略词，代表GNU Network Object Model Environment（GNU网络对象模型环境）。用不太时髦的话来说，GNOME就是平常所说的桌面管理器。虽然窗口管理器建立在X之上并使系统便于使用，而GNOME却是建立在窗口管理器之上（可以选择建立在哪一个窗口管理器上）的，它提供更进一步的服务（图形的和后台的服务），使用户界面功能更强。

从图1-3所示的屏幕中可以看到，我们仍然运行在Window Maker窗口管理器内（虽然大多数Linux分发包含Enlightenment作为所选择的窗口管理器，但使用哪个窗口管理器完全是个人的喜好）、不过由于增加了GNOME使其可用性更强了。

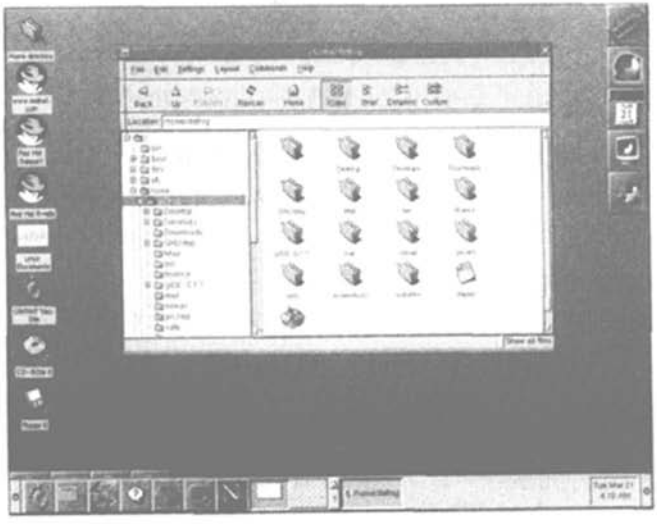

图1-3 在Window Maker中运行GNOME

屏幕左边的图标给出了对诸如用户主目录、软盘驱动器以及各种网站等方便的单击访问手段。可以料想到，这些东西完全是可以定制的。屏幕底部的GNOME面板同样也是可以定制的。单击面板中的图标能够快速方便地访问应用程序以及窗口管理器中别的工作区，单击最左边的图标将弹出访问GNOME自己的应用程序组的菜单，如图1-4所示。

GNOME远不止只是在窗口管理器上提供了一组按钮，它还提供了一组服务，这组服务是设计来规范用户在所用的应用程序中给出的界面。GNOME的应用程序利用GTK+库实际地描绘出它们的用户界面。你大概已经看到过这种界面。所有使用相同图形窗口小部件（窗口小部件为控件、按钮、菜单等组成应用程序用户界面的东西）设置的GNOME应用程序都有共同的外观。如果懂得怎样在一个应用程序中使用GTK+的标记对话框，就会知道在另一个应用程序中怎样使用它。这表示每当用户遇到一个新的应用程序时，所需的学习时间较少。

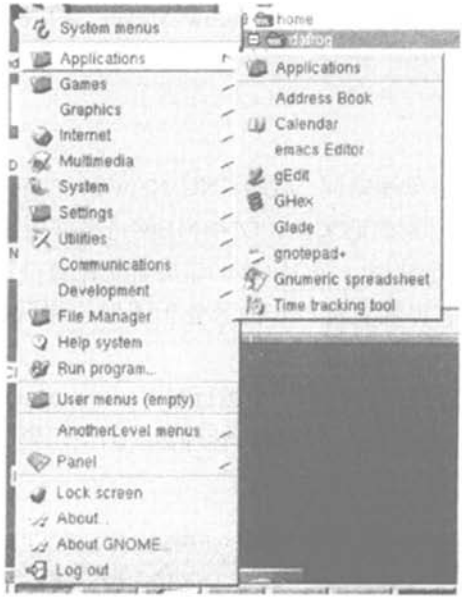

图1-4 GNOME访问本身的应用程序组的菜单

此外，各种GNOME库和框架为使用它们的应用程序提供了一组常见的服务。例如，支持拖曳使用户能方便地将数据从一个正在运行的应用程序拖到另一个正在运行的应用程序。而且，GNOME的拖放与旧的X拖放标准是兼容的，它也与Motif库在X早期引入X的拖放标准是兼容的。这表示GNOME的应用程序能与传统的X应用程序一起工作。

应该指出的是，GNOME并非是喜欢X的用户的唯一可用的桌面管理器。还有一个称为KDE的桌面管理器可供使用（见图1-5）。

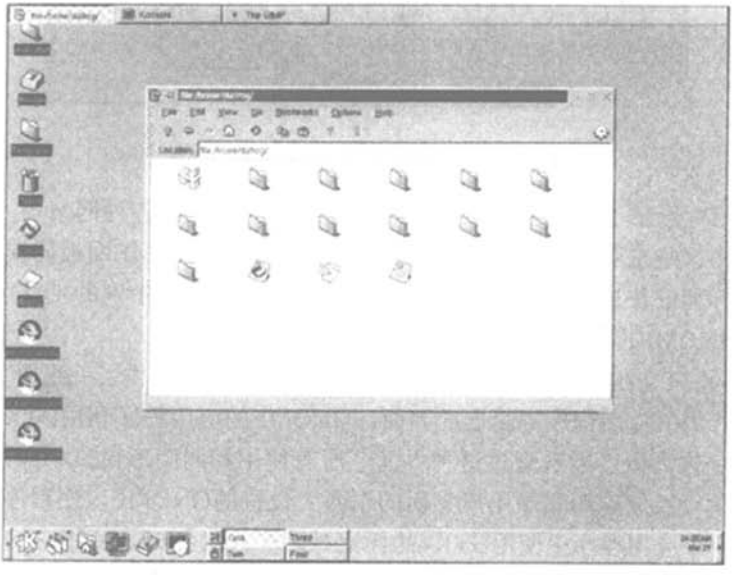

图1-5 KDE

虽然KDE具有与GNOME非常类似的外观，但差别也很多。KDE并不仅仅依靠在窗口管理器上提供服务，它本身也是一个窗口管理器。在内部，GNOME应用程序利用GTK+库绘制用户界面，而KDE使用的则是来自Troll Tech Inc的一个名为Qt的库。正如下一节所述，这正是坚持采用GNOME和GTK+进行开发而不利用KDE进行开发的原因。

值得指出的是，虽然你不可能同时运行KDE和GNOME，但只要需要，没什么东西能阻止你在KDE中运行GNOME和GTK+应用程序，或者在GNOME中运行KDE应用程序。它们实际上可以一起运行。但是，为了一致起见，建议在GNOME桌面系统中运行本书中的应用程序。

1.3 为什么要使用GTK+和GNOME

从用户的观点来看，不能否认KDE和GNOME这样的桌面管理器都是好东西。它们标准化了桌面系统并使其功能呈指数级地增加，把X的世界推入了新的千年。它们使X对具有诸如Windows和MacOS这样的GUI使用经验的用户极具竞争力。

而从程序员的角度来看，他们对应该学习哪一种桌面管理器并不非常清楚。首先，正如前面所述，可以在KDE中运行GNOME，反之亦然。因此，在使用X时程序员应该花时间和精力来学习哪一种桌面管理器呢？

这里有一种假定，即本书的大多数读者使用的都是Linux。为什么你使用Linux呢？Windows可用的应用程序比Linux多。Windows的应用程序容易得到完全商业化的技术支持。而Linux的应用程序则不然。那么，究竟为什么要使用Linux呢？这个问题的答案可能会是“它与众不同”、“它不是Microsoft的”、“它是免费的”或者“它很酷”。所有这些答案都可以归结为一个：自由。让我来解释一下。

GNU/Linux（或就像大家所知道的那样，称为Linux）是免费的。现在有许多心怀不满的人躲在房间里抱怨为了得到软件拷贝要付很多钱。这不是我指的免费的含义。自由软件的想法最初来自20世纪50年代的一批程序黑客，他们决定在大学的大型计算机上玩一个雅致的把戏。后来，这个主意由Richard Stallman（自由软件基金会的奠基人）正式地推广了。这些黑客（当然，此处这个词的意思指的是那些以解析代码为乐，而不是为了盗窃你的信用卡数据的那些人）有一个灵巧的系统，姑且称之为代码提取者。如果我在该大型计算机上编写一个应用程序，可能会将我的纸带扔在读带机下的抽屉里。然后，当天晚上，你可以进入计算机房，检查抽屉得到我的纸带。这样你就能够完全自由地研究我所做的工作，纠正其中的错误，对其进行改进，当然也可以从中学到知识。这样你可以迅速地得到提高。“你完全自由地”对我的代码做任何想做的事情。现在，知道免费的意思了吗？

如果信步走入本地的软件商店，拿起最新发行的Windows或你又爱又恨的字处理器，很可能在撕掉封面上的紧包装时，会掉出一个小纸片，其上盖有许可的词语。这些许可词语所说的东西一般是：在没有读懂此许可之前甚至连紧包装都不应该撕掉；在这之后说的是你不能拷贝此软件（可能允许有一份备份拷贝）、与朋友分享、获得其源代码、借用其中的代码段、对其进行改进、学习，或者除了在一台且仅在一台机器上安装并保留它以外，一般不能做别的事情。此许可一般还说，违反这些规定是一种比谋杀还坏的犯罪，结果是你和你的家庭注定要遭到蔑视和唾弃，等等诸如此类的软件行业对这种行为的典型谴责。

现在，考虑一下上述内容。你可能为这个发亮的小盒子付了数百美元，但根据那个小纸片上的规定，能用它做的事情受到很大的限制。而自由软件不受授权软件的这些限制。

对于自由软件，可以进行拷贝。可以研究其代码并学习其中的技巧。可以改进其程序包并重新分发它。可以借用其中的代码块。可对它做任何想做的事情而完全没有约束。Richard Stallman和他的FSF的研究小组——GNU工程的创立者将上述内容正式化为一个称为GPL（GNU Public License，GNU公共许可证）的东西。用这种许可证发行的软件基本上提供了做上述事情的所有自由。

但是，也存在一个限制。FSF渴望全世界都以他们为榜样。他们希望自己的源代码在全世界都可以得到，希望软件界一般不要约束它的用户。然而，他们确实不希望在自由软件的基础上做所有这些开发工作只是为了让某个公司窃取它。出于这个原因，如果有人开发了一个库（如GTK+）并在GPL之下发表它，则不允许你用这个GPL的库开发一个商业性的、需要许可的应用程序。

请不要惊慌。我知道你们都在盘算怎样利用自己新得到的GTK+和GNOME知识淘金呢。完全可以。还有另一种许可证，称为Lesser GPL（次GPL）或LGPL。LGPL通常也称为Library GPL（库GPL），在将它用于一个程序库时，基本不对该库进行限制，而且还允许你在商业性的软件中使用它。GTK+和GNOME以及构成它们的库（后面要研究）全都是LGPL的。如果你想开发一个使用它们的商业性的应用程序，尽管去做，没问题。没谁会阻止你，控告你或臭你。是否与其他开发人员共享你的成果或代码中的秘诀，全凭你的良心了。

那么，在KDE和GNOME之间选择哪一个作为开发平台呢？很好，这是一个很实际的问题。GNOME利用GTK+完成其应用程序用户界面的绘制工作。正如所述，它们是LGPL的库，可以访问它们的源代码，而且全球自由软件团体踊跃地对其作出贡献以使其更完善。而KDE使用Qt库完成所有绘制工作。那是一个商业性的程序库。其上附加有一个不同的许可证（从版本2起为QPL），如果你打算利用它做任何商业性的事情，那么需要为Qt的商业版付费。无需高深的哲理都会知道，利用Qt进行开发基本上不会对全球自由软件团体作出什么贡献。而对GNOME和GTK+进行开发基本上是为一个免费工程作贡献。因为你的应用程序需要这些免费的程序库，这样会使更多的人知道这些免费库的好处，而且你很可能也会免费提供自己的代码。

那么，选择哪一个进行开发最好呢？照我的看法（也可能是你的看法，如果你买了这本书的话），选择GNOME和GTK+最好。

除了自由软件这个理由外，我们还发现GNOME和GTK+向开发人员提供了一组功能很强的API。这些API简化了X应用程序的开发工作，特别是与利用X本身的API从头进行开发相比显得更为简单。

1.4 准备使用GTK+和GNOME

这样，大家都作了选择，准备安装GNOME并开始编写GTK+和GNOME的应用程序了（当然首先是在阅读了本书内容之后）。做什么事情呢？从何处入手？所需的是构成GNOME和GTK+的整套GNOME程序包（库、核心系统等）。需要提醒的是，除非你对自己的Linux套件、安装的应用程序、升级库以及诸如此类的东西很满意，否则工作起来不会顺利。必须说的是，在我第

一次安装像GNOME这样的大程序包时，学习了许多Linux套件的东西，我读了许多书和README文件。因此，如果你希望启动GNOME并正常运行，最好是搞一套最新发行而你自己又喜欢的Linux分发包。

所有最新发行的Redhat、Debian、SuSE、Slackware等大型软件都包含有启动和运行GNOME的一切东西；并作为Linux安装过程的组成部分安装其所有组件。除了作为Linux安装过程的一部分安装GNOME外，没有别的更快、更方便和更安全的方法了，因此我们竭力建议你这样做。

如果你确实想升级自己的Linux系统以支持GNOME，则需要访问www.gnome.org，这是GNOME在Web上的主页。在其中的下载页上，可以找到开始工作时所需的东西，GNOME下载页面如图1-6所示。

图1-6 GNOME的下载页面

你很快就会注意到，可以用大量不同形式得到GNOME的各种组件。有tarball，它们是保存需要编译的源代码的压缩档案文件。有支持RPM系统的RPM程序包（如Redhat）以及Debian用户的DEB程序包。选择哪一种完全根据你的喜好，当然还要看你的系统能够支持哪种形式。其他类型的文件也要检查你的系统，以保证具有支持GNOME运行所需的各种应用程序和库，对于不同的发行版本这种需求是不同的。除了下载某个核心GNOME程序包，试安装它以外，确实没什么更好的办法知道你运行GNOME需要些什么了。

1.4.1 Tarballs

所需做的第一桩事是找出系统的include和lib目录。一般它们位于/usr目录下，而在某些系统