光盘内附书中范例

乔林　编著

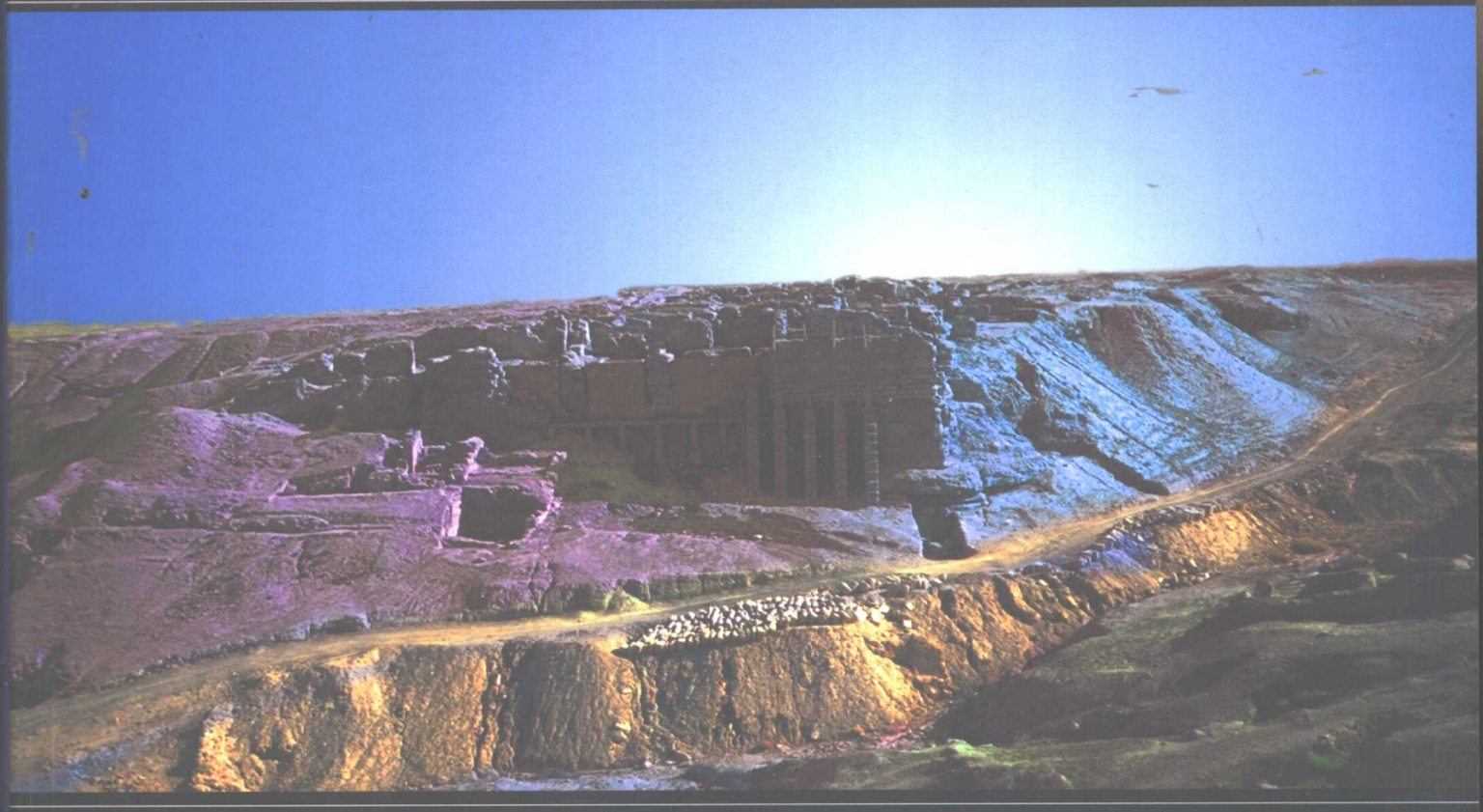
您想站在3年后软件开发的风口浪尖上吗？

您想编写既可以在Microsoft Windows下运行

也可以在Linux下运行的跨平台软件吗？

从现在起就开始学习Linux编程吧！

Kylix是您精通Linux编程的敲门砖。



Kylix程序设计

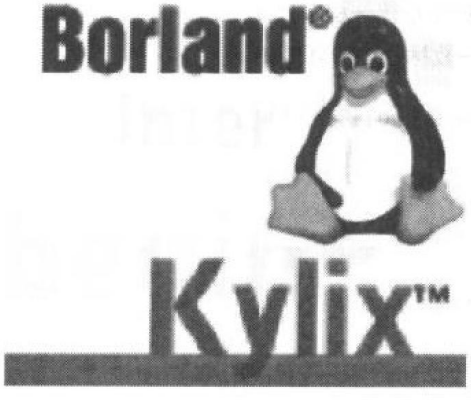
进　阶　教　程

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Kylix 程序设计

——进阶教程

乔 林 编著



中 国 铁 道 出 版 社

2001 年 • 北京

（京）新登字 063 号

内 容 简 介

本书以多个应用程序实例为基础，介绍了 Kylix 程序设计的基本方法。内容涉及窗体与窗口、标准信息框、应用程序与多窗体界面设计、鼠标与键盘输入、记录与指针、过程与函数、面向对象程序设计的基础知识与基本理论、继承、多态性与动态联编等。

书中详细剖析了各个实例，使读者学会正确的思考方法，以及如何正确的将思考方法转化为正确的程序代码。附带光盘中给出了书中所涉及的实例和练习的全部源代码。

本书是 Kylix 程序设计的中级读物，适合计算机软件开发人员和一般计算机人员，尤其是 Linux 爱好者使用。如果与本系列的其他图书配合使用效果更佳。

图书在版编目（CIP）数据

Kylix 程序设计进阶教程/乔林编著.—北京：中国铁道出版社，2001.7

ISBN 7-113-04285-6

Ⅰ.K… Ⅱ.乔… Ⅲ.①数据库系统—程序设计—软件工具，Kylix—教材

②PASCAL 语言—程序设计—教材 Ⅳ.TP311.56

中国版本图书馆 CIP 数据核字(2001)第 044883 号

书　　名：Kylix 程序设计——进阶教程
作　　者：乔　林
出版发行：中国铁道出版社（100054，北京市宣武区右安门西街 8 号）
策划编辑：严晓舟
特邀编辑：王占清
封面设计：冯龙彬
印　　刷：北京兴顺印刷厂
开　　本：787×1092　1/16　印张：29.25　字数：704 千
版　　本：2001 年 8 月第 1 版　2001 年 8 月第 1 次印刷
印　　数：1～5000 册
书　　号：ISBN 7-113-04285-6/TP・589
定　　价：53.00 元

前　言

可以毫不夸张地说，Kylix 是 Linux 发展的里程碑——它的出现是革命性的。

关于 Kylix

近几年来，Linux 无疑是业界上升最快的操作系统。作为最具竞争力的企业环境之一，由于无比低廉的价格，以及与价格相比实在太好的可靠性，Linux 广泛应用于 Web 应用服务器，并迅速成为 Windows 操作系统的主要竞争对手。

然而现实情况时，在 Linux 环境下开发应用程序不仅繁琐困难而且效率太低，程序员不得不进行大量的重复性劳动以编写 X Window 图形用户界面的程序代码，这种局面十分类似 Visual Basic 问世前 Windows 操作系统所面临的难局。随着 Linux 的发展，业界迫切需要一种快速开发工具，以弥补 Linux 下应用程序不足以及编程复杂的欠缺。

1999 年的 9 月 28 日，Borland/Inprise 公司正式宣布开发 Linux 环境下的快速开发工具，时隔仅一年半，Borland/Inprise 公司就适时推出了 Kylix。

Kylix 实际上是 Delphi 的 Linux 版本，它与 Delphi 一脉相承。Kylix 的出现，彻底终结了 Linux 过于艰深、曲高和寡的历史，每个普通的程序员都可以使用 Kylix 快速开发 Linux 下的应用程序。Kylix 是 Linux 发展的里程碑，它使得 Linux 第一次可以在桌面操作系统上与 Windows 一争高低。

为什么要使用 Kylix/Delphi

Kylix，或者说 Delphi for Linux，是运行于 Linux 环境下的 Delphi。无论是窗体设计器、代码编辑器还是对象检查器、部件面板，Kylix 与 Delphi 的运行界面都几乎不存在什么差别。此外，Kylix 下的编程习惯与 Delphi 几乎完全相同：键盘响应模式与鼠标操作完全一样，菜单和命令也几乎完全一样。

从集成开发环境上看，Kylix 与 Delphi 最大的不同是使用 CLX 部件库代替了 VCL 部件库。CLX 与 VCL 在众多方面都是相似的，CLX 的最大优势是基于 Qt 工具包，支持跨平台开发。

使用过 Delphi 的读者将会发现向 Kylix 中移植 Delphi 的代码是最容易不过的事情。除了 CLX 部件开发，大部分 Delphi 程序都只需要按照 Kylix 指出的方法做一些小改动就可以在 Linux 下直接编译运行。

您想站在 3 年后软件开发的风口浪尖上吗？您想编写既可以在 Microsoft Windows 下运行也可以在 Linux 下运行的跨平台软件吗？从现在起就开始学习 Linux 编程吧！Kylix 是您精通 Linux 编程的敲门砖。

关于本丛书

本丛书的宗旨是通过大量或深或浅的实例讲解 Kylix 的内容。整套丛书使用多个有趣的例子讨论 Object Pascal 语言的基本要素、Kylix 编程的基本与高级方法。本丛书引出的概念不是按照传统的教学顺序组织的，而是按照对读者是否有用的顺序组织的。读者会发现，这样的组织方式将使学习过程更有效。

贯穿整本丛书的是读者应该遵循什么样的步骤，应该采取什么样的思考方法，以及如何将自己的思考转化为正确的程序代码。因此，丛书中包含大量实例和练习，这些实例和练习有的难、有的易，而且在很多情况下还有一定的关联。我们希望读者在阅读本丛书的同时也上机实践。

每一章后面都包含一些问题与练习，它们的答案通常可以在书中找到线索，但也有一些练习必须通过上机实践才能完成。笔者深信，只有实践才是出真知的最佳途径。当然我们也要指出，有一些练习是稍难的。笔者同样深信，只有独立完成稍难一点点的问题才是迅速提高的唯一途径，太易的练习没有任何意义，而太难的练习则失去了实践的价值。对于较难的练习我们将给出一定的提示，请读者务必完成它们。

本丛书是如何组织的

本丛书包括四个部分：

- 《Kylix 程序设计——基础教程》：基础教程着重解决安装和使用 Kylix 面临的基本问题。考虑到 Linux 下的程序员大多对 C/C++ 很熟悉，而对 Object Pascal 语言不太了解，该书详细介绍了 Object Pascal 语言，对于初学者有极大的帮助。
- 《Kylix 程序设计——进阶教程》：进阶教程着重讨论《基础教程》遗留下来的 Object Pascal 语言的高级特性以及如何设计漂亮的应用程序界面。本书同时研究了如何使用 Kylix 开发数据库应用程序。
- 《Kylix 程序设计——实战教程》：详细讨论使用 Kylix 开发高级应用程序、多线程应用程序、Internet 应用程序、数据库应用程序、跨平台应用程序设计的基本技巧。
- 《Kylix 程序设计——类库参考手册》：提供完整的 CLX 类库参考手册，免去读者在帮助系统中跳来转去的困扰。

本丛书的内容由浅入深，没有阅读前面的内容就直接学习后面的章节可能会给读者带来一定的麻烦，当然对相应内容有相当程度了解的读者又另当别论了。

本丛书是集体劳动的结晶，参加编写的有乔林、林杜、费广正、王程铭、王志海、何隽、计莉琴、朱辰麒、乔木、蒋培、蔡宏、汪巨涛、陈爱华、王冰、乔森、徐斌、马友兰、胡一敏、李享、刘丰收、朱琨、张喜二等。由于编者学识水平有限，书中难免有缺点、错误和不当之处，恳请读者批评指正。

本丛书的读者对象

本丛书包含了使用 Kylix 编程的方方面面。即使读者没有任何计算机编程经验，也可以阅读本丛书。为理解本丛书的内容，读者应该熟悉 X Window 操作系统的基本知识。对读者

最低限度的假设是能够知道 X Window 的基本操作，能够知道如何使用 X Window 环境就足够了。

我们再一次地重申，本丛书的对象包含了 Kylix 的初级到高级的所有读者：

- 初学者：本丛书的《Kylix 程序设计——基础教程》和《Kylix 程序设计——进阶教程》是极好的学习材料，初学者可以从中学到几乎全部的 Object Pascal 语言细节和基本使用技巧。
- 中级读者：可以通过学习本套丛书澄清很多以前没有完全了解的概念。
- 高级读者：我们敢保证，您肯定会在本套丛书的字里行间发现很多在其他书中看不到的内容——那是作者使用 Kylix 时的独特体会，也许是本丛书最具价值的部分——这部分内容对于您理解 Borland 公司在实现 Kylix 时的设计原则有莫大的裨益。要知道，Kylix 本身是如何实现的与我们使用她时采用什么样的编程方法和编程习惯息息相关。

“边做边学”的方法是最有效的。每学完一个例子，尝试着改变一点点，或者添加一点东西，并改变一些代码将帮助读者体验进步和成功的乐趣。

编 者

2001.3

目　录

第 1 章　窗体与窗口

读者在《Kylix 程序设计——基础教程》中已经学习了创建和使用窗体的基本方法与技巧，本章将继续这部分内容，深入研究窗体和窗口各元素或属性之间的细微关联。

本章讨论窗体与窗口的基本概念与基本属性集。在本章里，我们将学习 X Window 窗口系统的一般概念，如何设置窗体与边框的样式、如何设置窗体的位置与大小、如何设置窗体上各部件的比例，以及如何在运行检索窗体的各个部件。

本章内容包括:

- X Window 系统与窗口
- 窗体与边框的样式
- 部件的字体比例
- 窗体的位置与大小
- 窗体中的部件

1.1　X Window 系统与窗口

在 X Window 中，最常见的用户界面元素是窗口，因而使用 Kylix 开发应用程序时最常操作的也是窗口部件——不仅仅是 TForm 部件，其他许多 Kylix 部件也都是窗口。

在《Kylix 程序设计——基础教程》中，我们已经讨论了如何使用菜单和工具栏设计一个文本编辑器。本章及后面三章将详细研究用户界面设计的基本技巧。

我们假定读者已经阅读过《Kylix 程序设计——基础教程》或者了解《Kylix 程序设计——基础教程》所引入的许多概念。如果您对某些内容不熟悉，请参阅《Kylix 程序设计——基础教程》。

1.1.1　X Window 系统与 X 服务器

术语 X Window 系统，或简称为 X 系统，一般指几个功能部件的支持系统，这些功能部件可以使各种位图显示器上基于窗口的图形输出更加容易[①]。

X 系统的核心是 X 服务器（X Server），一种运行于计算机上的进程（process）。需要显示输出的应用程序（称为 X 客户程序，X Client）通过一种内部通讯机制与 X 服务器进行信息传递，共同完成显示任务。

在 X 服务器和 X 客户程序间的通讯要遵守一种定义完备的协议，称为 X 协议（X

---

① Naba Barkakati 著，魏永明、李铁民、游华云等译，RedHat Linux 奥秘（第三版），电子工业出版社，北京，2000。

Protocol)。此外，X 系统还包括例程库 XLib，它包含用于完成显示任务的程序代码。

X Window 系统最初是在 1984 年由 MIT 开发的，它的设计与开发得到了 MIT 计算机科学实验室和 MIT Athena 项目的共同赞助。最初的商业化 X 版本与 1986 年运行于 DEC 的 Ultrix 操作系统中，版本号为 X10R3（版本 10，第 3 次发行）。

X10 的反馈信息促使该项目成员重新设计 X11 的协议。1987 年 1 月，第一次 X 技术会议召开，11 个主要的计算机供应商（当时）宣布联合起来支持并促使 X11 标准化。同年 9 月，X11R1 发布。

为保证 X 能在一个公开组织的控制下持续发展(与 Linux 有很多发布版本一样，各大计算机公司的 Unix 系统也各自为政，互不协作)，MIT X 联盟与 1988 年 1 月成立。在 Robert W. Scheifter（X 体系结构的主要设计者之一）的领导下，X 联盟对于 X 的成功起到了很大的作用。

随着 X 在工作站上逐步获得广泛应用，X 联盟继续在几个方面对 X 进行改进，包括支持 C++ 语言和提供面向对象的工具包 Fresco，这些扩展都成为 1994 年 4 月发布的 X11R6 版本的标准功能。

在所有的 X11 发行版本中，X 协议一直保持稳定，所有功能增强都是通过使用 X11 协议扩展支持范围而不是修改 X 协议。大多数 Linux 操作系统现在都使用 X11R6 版本的 X 系统。

当用户工作于一个典型的 X 显示状态下，X 服务器即运行于计算机上，并控制着监视器、键盘和鼠标等设备。X 服务器对 X 客户程序要求打开窗口或在窗口中绘制图形的命令作出适当的反应。

Linux 操作系统下的 X 服务器名称为 XFree86："X" 表示使用 X 协议，"Free" 意味着其为免费的，"86" 表示其支持 Intel x86 系列处理器。

X 服务器负责在显示器上输出内容。因此，X 服务器必须访问和使用显示卡或图形加速卡。PC 的显示卡和图形加速卡是如此众多，以至于要创建一个可以支持所有显示卡和图形加速卡的 X 服务器基本上就是一种不可能的任务。

幸运的是，大多数显示卡和图形加速卡都使用几种典型的芯片，XFree86 因而提供了多个 X 服务器，每个 X 服务器都支持一种特定芯片。

即便如此，XFree86 服务器也必须能够区别对待不同公司使用同一种芯片设计的显示卡或图形加速卡——即使是使用同一种芯片，它们的工作模式与工作方法也可能是不同的，甚至是不兼容的。这也就意味着，X 服务器必须了解显示卡和图形加速卡的详细设计规范。

除此之外，X 服务器同样需要了解显示器、键盘和鼠标的详细设计规范。

注意，在任何时候都是 X 服务器控制各种设备，而不是我们的应用程序控制各种设备。

1.1.2 X Window 图形用户界面

简单地说，应用程序的用户界面确定了它的外观和操作方式。当用户界面使用图形时，我们习惯称其为图形用户界面（graphics user interface，GUI）环境。这些 GUI 环境需要频繁地使用鼠标进行操作。

Linux 下最常见的 GUI 环境有 KDE 和 GNOME 等，RedHat 即带有此两个 GUI 环境。一般地，Linux 的 GUI 环境包括四个部分：

- GUI 窗口系统：管理图形在显示屏幕上的输出；完成基本的文本和图形绘制功能。
- 窗口管理器（Window Manager）：允许用户移动和调整窗口的大小；负责各种窗口的显示，它通常需要为窗口添加装饰性的结构（例如，窗口边框）；管理窗口的输入焦点，用户可以使用鼠标和键盘改变窗口的输入焦点，即确定哪个窗口接收输入。窗口管理器事实上充当着 X 客户程序的角色。
- 开发工具包（Development Tool-Kit，DTK）：定义了完整的编程接口。程序员可以使用开发工具包编写应用程序，使应用程序能够充分利用窗口系统提供的功能。
- 风格向导（Style Wizard）：建立基本约定，指定应用程序用户界面的外观和操作方式。风格向导仅限于用户界面中的那些公共元素，例如，按钮、窗口边框与标题栏等。

一个窗口管理器就是一个特殊的 X 应用程序（X 客户），它负责所有运行在屏幕上的各种应用程序窗口之间的交互。我们需要窗口管理器控制每个应用程序窗口的位置与大小。没有窗口管理器的支持，我们不可能改变一个窗口的位置与大小。

例如，考虑同时运行的两个 X 应用程序（X 客户），它们显示同一个屏幕上，其中的任何一个程序都不知道另一个程序的存在与需求。假设我们首先运行第一个程序，然后运行第二个程序，第二个程序的窗口占满了整个屏幕，显然，我们这时无法看到第一个程序的窗口了。为此，我们需要一个窗口管理器提供从第二个程序切换到第一个程序的工具和手段。

GNOME 本身不包括窗口管理器，它使用的是其他窗口管理器，这同时也意味着它可以使用多种窗口管理器，例如 WindowMaker 和 Enlightment 等。

KDE 则不然，它不仅提供窗口系统，还自备了一套窗口管理器 KWM（K Window Manager）。自然，KDE 同样可以搭配其他的窗口管理器。

1.1.3 买双 ANTA 运动鞋：X Window 与 Microsoft Windows

熟悉 Microsoft Windows 的读者肯定会对 Linux 下的 X Window 感到迷惑：窗口就是窗口，哪有这么多的概念！

在 Linux 操作系统中，X Window 服务器负责操纵硬件，X 客户程序负责维护和提供显示信息，X 客户与 X 服务器之间通过 X 协议进行通讯。

X Window 的 GUI 环境与 X Window 服务器是两回事。勉强打个比方，前者相当于我们对 Microsoft Windows 的基本理解，后者相当于我们对 Microsoft Windows 中硬件驱动程序的