

e 时代自由软件系列

Kylix 实例与技巧

博嘉科技 主编 李晓娟 杨丽丽 编著

国防工业出版社

National Defence Industry Press

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

 时代自由软件系列

Kylix 实例与技巧

博嘉科技 主编
李晓娟 杨丽丽 编著

国防工业出版社

·北京·

内 容 简 介

Kylix 是 Borland 公司近期推出的 Linux 平台下的快速应用程序开发环境。本书按照 Kylix 的体系,系统、全面地介绍了使用 Kylix 设计和实现 Linux 应用程序的各方面知识。其中包括:Kylix 的产生背景、集成开发环境的使用、常用组件和一些较深入的编程问题,如异常处理、多线程、跨平台开发、图形图像处理 and 数据库应用程序的开发等。

本书适用于 Kylix 的初学者和具有在别的编程环境下编程经验的人员使用。本书源代码请于 <http://www.ndip.com.cn/computer/> 免费下载。

图书在版编目(CIP)数据

Kylix实例与技巧/李晓娟,杨丽丽编著. —北京:
国防工业出版社,2002.6
(e时代自由软件系列)
ISBN 7-118-02782-0

I . K… II . ①李…②杨… III . Linux 操作系统—
软件工具, Kylix IV . TP311.65

中国版本图书馆 CIP 数据核字(2002)第 007200 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 29¼ 677 千字

2002 年 6 月第 1 版 2002 年 6 月北京第 1 次印刷

印数:1—3000 册 定价:39.00 元

(本书如有印装错误,我社负责调换)

前 言

在众多的操作系统中,Linux 操作系统以其独有的自由开放性、稳定性,赢得了众多用户的喜爱,并得到飞速发展。然而,对于 Linux 来说,要迈出下一步,成功地成为一个主流平台,将需要大量的支持 Linux 平台的应用软件。而 Kylix 出现的目的就是让开发人员能够开发出这样的软件。

Borland 公司发布的 Kylix,是一种基于 XWindow 的 Linux 快速开发工具。Borland 一直以 Borland C++ ,Delphi,C++ Builder,JBuilder 等系列开发工具著称,Kylix 秉承了 Borland 完善的可视化 IDE 开发环境、强大的调试功能和丰富的 CLX 组件,可与 Linux 下的 Delphi 相比。

如今,除了 Kylix 以外,Linux 平台上还没有任何其他进行快速应用程序开发的工具。因此,无论是 Linux 的开发团体,或是成千上万的 Linux 开发人员,还是想在 Linux 平台上开发应用程序的 Windows 开发人员,都迫切地希望使用 Kylix。Kylix 将为 Linux 带来大量的开发者和应用软件,同时也将为 Delphi 和 Delphi 程序员带来一个福音——在 Windows 和 Linux 下进行跨平台的开发!

为了帮助广大程序员尽快了解并精通 Kylix 编程,也为了使 Delphi 程序员能够迅速过渡到 Linux 平台,我们编写了这本书,希望能达到抛砖引玉的作用。

本书内容广泛,兼容性强,涉及了 Kylix 编程的大部分领域,不仅适用于 Kylix 的入门者,而且也适合于 Borland Delphi,C++ Builder 及 VB 的用户平稳地向 Kylix 过渡。在本书中,我们着重体现了丰富的编程实例,这些例程精练、实用、完整。同时,作者将自己在开发程序过程中总结的经验和教训、一些读者在编程过程中遇到的问题等融合在本书中,并且使这些编程技巧尽量独立,便于读者快速查找。

本书内容概述:

全书共分为 9 章,内容安排如下:

第 1 章对 Kylix 进行了简要的介绍。通过对 Kylix 的简介、Linux 的发展需求、Kylix 与 Delphi 之间的比较、Kylix 的技术特点以及安装等方面的讨论,让大家对 Kylix 的框架有一个基本了解。

第 2 章介绍 Kylix 的集成开发环境。Kylix 与 Delphi 拥有一样的集成开发环境,本章通过一个简单的例程,向大家展示 Kylix 简单而又强大的完美开发环境。

第 3 章向读者介绍 Kylix 的核心 CLX 以及 Kylix 基本组件的使用。对各个基本组件的使用方法分别以实例进行介绍,让大家快速掌握 Kylix 组件的使用。

第 4 章介绍了 Kylix 异常处理机制。讲述了异常的基本概念,介绍了 Kylix 中预定义的标准异常类,以及异常的语法、作用和其他的一些高级特性。最后,简介了如何定义并

使用新的异常类。

第5章介绍图形图像的应用开发。通过几个实例循序渐进地介绍 Kylix 的图形图像组件的使用。在本章中,我们应用 Kylix 的图形函数,开发了各种图形的特殊效果。最后,通过一个综合实例的开发,展示了一个实用的绘图软件的编写方法,相信这对大家会有所启发。

第6章介绍了怎样用 Kylix 进行基本的数据库开发,包括表格、字段、数据库敏感组件等内容。Kylix 在支持数据库方面可谓尽善尽美,它使用 dbExpress 连接多种数据库格式,可进行桌面、客户机/服务器和多层数据库系统的开发。本章通过详尽的实例,为下一章的客户机/服务器应用程序开发奠定良好的基础。

第7章介绍了 Kylix 的 C/S 应用程序开发方法。本章首先从各大型数据库的链接和 SQL 语言谈起,详细地分析了 C/S 开发中的问题,并用一个实例循序渐进地指导读者开发一个客户机/服务器应用程序——学籍管理系统。

第8章介绍了多线程技术。主要介绍多线程的基本概念、创建多线程应用的过程,以及线程中最重要的状态转换、线程调度和安全等问题。

第9章简介 Kylix 的跨平台开发技术。介绍如何实现 Windows 和 Linux 系统下的应用程序间的相互移植,以及在两种不同的操作系统下开发应用程序的异同。本章对在不同环境中编写可移植的代码提供了有用的指导方针。

本书所有实例源文件均放在网页 <http://www.ndip.com.cn/computer/> 上,供读者免费下载与参考。

编写人员:

本书由李晓娟、杨丽丽主编并定稿,参加本书编写的还有张志、赵辉、李忠琼、杨克贵、李世生、张光学、李德鹏、王笙链、李思民、朱鑫鑫、李童、汪洁、唐芳、蔡嫣、何静、张勤、殷仕佳、李霞、陈伟、陈剑萧、刘新、李挺钊、任倩、肖薇、赵玲玲、胡容、刘岩、袁道华、刘唯识和娄明等,在此一并致谢。

对于本书,虽然我们都全力以赴进行编写校对,但个人的学识、能力毕竟有限,如果发现了书中的谬误,欢迎您指正批评,并提出宝贵的意见和建议。愿意参加“Kylix 实例与技巧轻松掌握”的培训学习,欢迎联系:

电话:(028)5404228

E-mail:bojiakeji@163.net

通讯地址:四川大学西区建筑学院成都博嘉科技资讯有限公司(邮编:610065)

编 者

目 录

第 1 章 Kylix 概述	1
1.1 Kylix 简介	1
1.1.1 Kylix 的概念	1
1.1.2 选择 Kylix 进行项目开发的原因和优点	1
1.1.3 Kylix 面向的开发人员	2
1.1.4 Kylix 的 3 个版本	3
1.2 Kylix 与 Linux	3
1.2.1 Linux 的需求	3
1.2.2 Kylix 与 Linux 下的其他开发工具的区别	4
1.2.3 Kylix 对 Linux 的推进作用	4
1.3 Kylix 与 Delphi	5
1.3.1 Delphi 简介	5
1.3.2 Kylix 与 Delphi 的联系	6
1.3.3 Kylix 与 Delphi 的区别	6
1.4 Kylix 的技术特点	7
1.4.1 高效的 Linux 内码应用软件	7
1.4.2 快速的集成开发环境	7
1.4.3 强大的代码编辑器和智能代码识别	8
1.4.4 先进的集成调试环境	9
1.4.5 基于构件的可视化开发	10
1.4.6 基于 Apache Web Server 开发	12
1.4.7 更为开放的数据库开发	12
1.4.8 简单的国际化开发和强大的第三方组件支持	15
1.4.9 构件源代码开放	15
1.4.10 完善的在线文档及详细的帮助文件	15
1.5 Kylix 的安装	16
1.5.1 系统需求	16
1.5.2 支持的 Linux 版本	16
1.5.3 安装过程	16
第 2 章 集成开发环境	21
2.1 IDE 全貌	21

2.2	IDE 基本结构	22
2.2.1	菜单及快捷工具栏	23
2.2.2	工程管理器、工程浏览器和 To - Do Lists	25
2.2.3	对象仓库	27
2.2.4	组件板	28
2.2.5	窗体和窗体设计器	29
2.2.6	对象监视器	32
2.2.7	代码编辑器和代码浏览器	33
2.3	定制 IDE	38
2.3.1	组织工作区	38
2.3.2	定制组件板	41
2.3.3	设置工程选项	43
2.3.4	设置工程与窗体的模板	44
2.3.5	设置工具项	45
2.4	Kylix 应用程序开发过程	46
2.4.1	创建工程	46
2.4.2	构造用户界面	46
2.4.3	编写代码	47
2.4.4	编译与调试	48
2.4.5	发布应用程序	50
2.5	编程入门——创建第一个 Kylix 应用程序	53
2.5.1	创建新工程	53
2.5.2	设计用户界面	55
2.5.3	编写代码	70
第 3 章	CLX 的基本使用	78
3.1	理解 CLX	78
3.1.1	CLX 概述	78
3.1.2	CLX 层次图	81
3.2	CLX 组件的通用属性、方法和事件	87
3.2.1	通用属性	87
3.2.2	基本事件和方法	90
3.3	常用组件的基本使用	91
3.3.1	Kylix 组件面板	91
3.3.2	窗体及 Frame 组件	91
3.3.3	Frame 组件	96
3.3.4	文本类组件	97
3.3.5	专用输入类组件	104
3.3.6	增减按钮编辑框 TSpinEdit	107

3.3.7 按钮及选择类组件	108
3.3.8 列表类组件	117
3.3.9 成组管理类组件	141
3.3.10 可视化反馈类组件	153
3.3.11 对话框类组件	159
3.3.12 网格类组件	163
3.3.13 其他类组件	169
第4章 异常处理	172
4.1 异常概述	172
4.2 异常类	173
4.3 异常处理语法	174
4.3.1 try...except 语句	174
4.3.2 try...finally 语句	176
4.3.3 异常的嵌套	177
4.4 异常处理的作用	179
4.4.1 代码保护	179
4.4.2 资源分配保护	179
4.5 异常处理高级特性	181
4.5.1 使用异常实例对象	181
4.5.2 异常的传递	182
4.5.3 异常处理的顺序	183
4.5.4 再次引发异常	184
4.5.5 哑异常	184
4.6 自定义异常	185
4.6.1 创建异常类	185
4.6.2 引发异常	186
第5章 图形处理	187
5.1 图像编程概述	187
5.1.1 图形坐标系	187
5.1.2 屏幕的刷新	188
5.2 绘图组件	188
5.2.1 TImage 组件	188
5.2.2 TShape 组件	190
5.2.3 TPaintBox 组件	191
5.3 Canvas 对象的属性和方法	191
5.3.1 Canvas 对象的属性	192
5.3.2 Canvas 对象的方法	209

5.4 综合实例	244
5.4.1 用户界面设计	244
5.4.2 处理多个画图工具	245
5.4.3 鼠标的响应	248
5.4.4 图像文件的基本操作	260
5.4.5 剪贴板的应用	263
5.4.6 绘图程序的完善	266
第6章 桌面数据库应用程序开发	269
6.1 数据库基础知识	269
6.1.1 关系数据库	269
6.1.2 数据库分类	270
6.1.3 数据库的安全性	271
6.1.4 事务处理	273
6.2 数据库系统组成结构	273
6.2.1 数据库系统的结构	273
6.2.2 数据库的三级模式	273
6.2.3 数据库管理系统	275
6.2.4 数据库应用程序	276
6.2.5 数据集	278
6.2.6 数据链接	281
6.3 数据浏览	282
6.3.1 浏览数据的一般步骤	282
6.3.2 数据库导航组件 TDBNavigator	283
6.3.3 TDBGrid 组件	284
6.3.4 简单字段显示类组件	294
6.3.5 列表字段显示类组件	296
6.3.6 查找字段显示类组件	300
6.4 数据管理	302
6.4.1 一个简单的单表数据的管理	302
6.4.2 数据管理的流程	303
6.4.3 主从表数据管理	307
6.5 数据查询	311
6.5.1 使用 Locate 函数	311
6.5.2 使用 Lookup 函数	312
6.5.3 使用 Filter	312
第7章 客户机/服务器	325
7.1 客户机/服务器体系结构	325

7.1.1	客户机/服务器的提出	325
7.1.2	客户机/服务器体系结构	326
7.2	使用本地 InterBase 服务器	327
7.2.1	准备工作	328
7.2.2	SQL 语言简介	328
7.2.3	使用 InterBase 的 isql 创建数据库	343
7.3	创建客户机/服务器应用程序	349
7.3.1	连接数据库	349
7.3.2	使用单向数据集	357
7.3.3	使用 Provider 组件	365
7.3.4	使用客户端数据集 TClientDataSet	367
7.3.5	创建客户机/服务器应用程序	367
第 8 章	多线程程序设计	406
8.1	线程的基本概念	406
8.1.1	线程和进程	406
8.1.2	多线程与单线程	406
8.1.3	多线程的优点	407
8.1.4	线程中的几个问题	407
8.2	线程的创建	408
8.2.1	Kylix 中的线程类	408
8.2.2	Kylix 线程类的创建	412
8.3	线程状态和线程控制	419
8.3.1	线程的五大状态	419
8.3.2	线程状态转换控制	421
8.4	线程优先级和线程调度	422
8.5	线程协调和死锁	423
8.5.1	线程互斥	423
8.5.2	线程同步	431
8.5.3	CLX 中线程安全	439
8.5.4	死锁	439
8.6	调试多线程应用程序	440
第 9 章	跨平台开发	442
9.1	几种移植的技术	442
9.1.1	平台指定移植	442
9.1.2	跨平台移植	442
9.1.3	Windows 仿真移植	443
9.2	CLX 和 VCL 的比较	443

9.2.1	CLX 和 VCL 基本结构比较	443
9.2.2	CLX 带来的不同	443
9.2.3	CLX 中没有支持的特性	445
9.2.4	不能移植的特性	445
9.3	Kylix 和 Delphi 之间的比较	446
9.3.1	Kylix 和 Delphi 单元比较	446
9.3.2	CLX 对象构造子	448
9.4	Windows 和 Linux 的比较	449
9.4.1	Windows 和 Linux 的环境差异	449
9.4.2	Linux 下的目录结构	451
9.4.3	Linux 下的编程差异	451
9.5	编写跨平台应用程序	452
9.5.1	编写跨平台代码的基本步骤	452
9.5.2	前期的准备	453
9.5.3	在 Windows 和 Linux 中共享源文件	453
9.5.4	使用条件编译	454
9.5.5	终止条件编译	455
9.5.6	发送消息	455
9.5.7	内联汇编代码	456
9.5.8	消息和系统事件	456

第 1 章 Kylix 概述

本章对 Kylix 进行了简要的介绍。通过对 Kylix 的简介、Linux 的发展需求、Kylix 与 Delphi 之间的比较、Kylix 的技术特点以及安装等方面的讨论,让大家基本了解 Kylix 框架。

1.1 Kylix 简介

1.1.1 Kylix 的概念

传说中,“Kylix”是古希腊的一种浅酒杯,与其他酒杯的不同之处在于它有一对提耳和一个伸到杯底的长把,就是那种当你旅行到“Delphi”城时喝酒用的杯子。

而在这里,“Kylix”则是指 Borland 公司经过 18 个月的努力,于近期推出的一种新的程序开发工具。Kylix 在 Linux 平台上使用,并且是 Linux 平台上第一个集图形用户界面(GUI)、Internet/Web 及数据库应用开发于一体的快速应用程序开发工具(Rapid Application Development, RAD),使用户能够在 Linux 操作系统下进行快速应用程序的开发。

Kylix 计划及其产品需求是以 1999 年 7 月 Borland.com 网站上,对超过 24 000 受访者的调查为依据的。调查报告显示:软件开发人员热切期望在 Linux 平台上仍然使用主流 RAD 开发工具,以便快速开发专业应用程序。为顺应大家的要求,Borland 公司推出的 Kylix 提供了一个包含可视化、双向工具(Two Way Tool)的集成开发环境,内建一个全新高速的编译器,能够编译 Object Pascal 开发语言,并且可以实现 Linux 版本的 Borland VCL(Visual Component Library,可视化组件库)构架,帮助 Delphi 用户由 Windows 平台平稳地向 Linux 平台移植。Kylix 能够支持 RedHat Linux、Corel Linux 以及其他主要的 Linux 版本。总的来说,Kylix 是集 Inprise/Borland 公司 15 年开发工具经验和在该领域拥有的广大用户群之优势,而开发出的连接 Windows 与 Linux 两大平台的开发工具。

1.1.2 选择 Kylix 进行项目开发的原因和优点

对于 Linux 来说,要想成为一个主流的开发平台,将需要大量应用软件的支持。而 Kylix 的出现就是为了让开发人员能够快速开发出这样的应用软件。

Kylix——Linux 平台上的快速开发工具——将 Web 程序、桌面程序和数据库程序开发集成在一起,并将 Borland 最优秀的编译技术与高效的可视化设计环境结合在一起,极大地提高了 Linux 应用程序的开发效率。Kylix 是目前 Linux 平台上惟一拥有以上全部优点的开发工具。

目前,除了 Kylix 以外,还没有其他的可以在 Linux 操作平台上进行快速应用程序开

发的工具。因此,无论是 Linux 开发人员,还是想在 Linux 平台上开发应用程序的 Windows 开发人员,都迫切地希望使用 Kylix 程序开发工具。开发人员能够利用 Kylix 提供的可视化环境更加快速地编写各种复杂的 Linux 应用程序,这是其他开发工具无法比拟的。对于 Web 程序、桌面程序和数据库程序的开发,Kylix 独特的可视化设计工具、调试工具和本地代码编译器可以快捷而有规则地把程序设计原型转化为软件产品。

1.1.3 Kylix 面向的开发人员

谁需要选择 Kylix? 简单地说,任何想在 Linux 平台上开发高质量、本地代码编译的应用程序的开发人员都应该选择 Kylix。也就是说,Kylix 非常适合以下几类开发人员。

1. Delphi 程序员

作为 Delphi 开发人员,在 Linux 下使用 Kylix 开发将具有得天独厚的优势。首先,Kylix 的界面与开发者熟悉的 Delphi 界面非常相似,比较容易上手。其次,Kylix 继承并发展了用户熟悉的 Windows 可视化编程模式,允许程序员快速地搭建本地 Linux 桌面应用程序;此外,Kylix 保留了和 Delphi 相同的工具集(包括窗体设计、目标检测、调试器和编译器),甚至新的库组件 CLX 模块(也是从 Delphi VCL 发展而来的),所以用 Delphi 开发的程序可以比较容易地移植到 Linux 上。如果开发人员把 Kylix 和 Delphi 结合起来,只需掌握一套基本的技巧和使用相同的基本代码,即可在 Linux 和 Windows 上开发跨平台的应用程序。

2. Linux 应用程序的开发者

Linux 下的传统开发工具不仅难以学习,并且其开发速度也不尽人意。而 Kylix 的出现使得 Linux 应用程序的开发人员找到一个高效的 RAD 开发环境,使得用户能够更加快速、轻松地开发应用程序。

3. Apache 开发者

使用 NetCLX 将加快 Apache 开发者进行 Apache Web Server 的开发速度。NetCLX 集成了浏览器、服务器和数据库开发技术,这将有助于用户发布可升级的 Web 应用程序。

4. 数据库开发者

使用 Kylix 可以在 Linux 里实现基于 Oracle 和 IBM DB2 环境的高性能电子商务应用程序。通过高速 dbExpress 数据库驱动器,可以将现存的 Oracle、DB2、InterBase 和 MySQL 数据库轻松地整合到应用程序里。使用 Kylix 提供的一套功能强大的数据仓库 GUI 组件,可以快速开发出数据中心应用程序。Kylix 使用 SQL Monitor、SQL 测试和调试组件对数据库应用程序的性能和反应能力进行测试,通过 Cached Updates 提升服务器的响应能力。

5. Web 开发人员

Kylix 可以加速 Apache Web 服务器的开发。Kylix 的 NetCLX 把浏览器、服务器和数据库开发技术结合起来,可以快速开发出支持高吞吐量的 Web 应用程序。使用已有的 65 个 Internet 组件可以轻松地吧 HTTP、FTP、SMTP 和 NNTP 等 Internet 功能整合到应用程序中。

6. VB 开发人员

Kylix 的编程模块与 VB 的事件驱动可视化编程模块相同,所以 VB 程序员可以使用

Kylix 很快地在 Linux 里编写各类程序。事实上,在 Linux 里除了 Kylix 外,还没有一个编程工具能在可视设计、交互调试器、内码编译器、数据库、桌面应用等方面跟 VB 如此接近。和 VB 不同的是,Kylix 应用程序使用 Kylix 内置的内码编译器,速度非常快,1min 可编译 4 百万行语句。

总的来说,Kylix 是开发人员理想的选择。它将辅助开发人员有效、快速地构造 GUI、Internet、DataBase 和 Web Server 应用程序。

1.1.4 Kylix 的 3 个版本

Borland 推出了 3 个 Kylix 版本:一个版本是价值 1999 美元的为接入数据库或网络服务器而制造的服务器开发工具;一个版本是价值 999 美元的桌面应用程序的开发工具;第三个版本是“公开版”,它是价值 99 美元的一个软件包(同时可以免费下载),这一版本只允许用户在 GPL 授权下开发开放性资源软件。

服务器版本的目的是帮助开发人员创建出企业级、数据驱动的 Apache Web 伺服器程序。这套软件包括了内置的多个数据库驱动软件,如 Oracle、DB2、MySQL 和 InterBase 等。服务器版本同时也包含了 NetCLX,这是一组 Apache Web 伺服器敏感组件,用于建立 Apache DSO 或 CGI(公共网关接口)应用程序。由于 NetCLX 组件与 Delphi 的 WebBroker 组件兼容,因此,它也被用来创建微软的 ISAPI 和网景的 NSAPI 服务器端应用程序。用户可以轻松地将一个 Delphi WebBroker 应用程序移植到 NetCLX,反之亦然。可以预言,由于 Kylix 具有以上优点,当前大量基于 Microsoft 的 IIS ISAPI Web 服务器软件将迅速地过渡到基于 Linux 的 Apache 上。

桌面版本主要是针对客户端和 GUI 应用程序开发的,以构建单层和客户机/服务器的数据连通性。这个版本包括 XML、MySQL 和 InterBase 数据链接。桌面 Kylix 的 DataCLX 是一组简单易用的数据访问构件,用于应用程序中,能够立即与数据库相链接。DataCLX 的底层是 DBDirect,这是一个跨平台的数据访问引擎。DBDirect 和 ODBC 或 JDBC 相似,但与之相比,DBDirect 规模更小、速度更快,同时,DBDirect 也是可扩充的。

公开版则只是包含了这套工具的有限的集成开发环境的版本。

这 3 个版本具有以下一些共同点:本地代码 Delphi ELF(Executable and Linkable format)格式编译器;可视化快速应用程序集成开发环境;Borland CLX 组件构架和可定制的各类库,包括 Web、Internet、数据库、Linux 系统和图形用户界面组件。同时,它们都包含优化的 Object Pascal 编译器、可执行的 32 位 Linux 源代码和共享库。

1.2 Kylix 与 Linux

1.2.1 Linux 的需求

首先回想一下早期的 Windows 编程。10 年前 Windows 3.0 发布时,编写 Windows 程序唯一的办法是基于 C 语言的 API,因为 Windows 程序很难开发,所以 Windows 也难以普及。

一个产品改变了一切:Visual Basic。用户也许对 VB 不以为然,但它是一个在恰当时

机推出的恰当的产品。VB 让程序员摆脱了复杂的 GUI、内存管理和任务管理,让他们可以专心于本职工作。对于那些没有时间记忆 Windows API 的程序员来说,VB 是一个很好的开发工具。

几年以后,Borland 发布了 Delphi。从那一天起,Delphi 就是一个比 VB 还要简单易用的工具。它更高效、更易于使用,具有更高性能。因此,如果想开发 Windows 程序,Delphi 是最好的选择。

正是由于这些简单易用的开发工具,才使得 Windows 编程成为广大程序员津津乐道的话题,并开发出大量基于 Windows 平台的应用程序,从而推动 Windows 的迅速普及。

Linux 由于其可靠性、可缩放性和低价的版权费用,进一步推进了 Internet 的发展,成为 Web 上的应用程序服务器,使得低成本的 Internet 应用平台成为可能;并快速地取代 Windows,成为大家所喜爱的桌面操作系统。然而,在 Linux 上开发应用程序的开发人员大都使用 GNU 的 GCC 编译器。虽然 GCC 是一个非常稳定的编译器,但是程序员必须一行一行地编写程序代码,同时必须对 Linux 系统非常了解。如果还想开发 X Windows 的应用程序,那么程序员还必须学习 X Window API 或者是 Linux 上最流行的两个图形化界面 GNOME 和 KDE 的 API。由于这个要求非常高,因而至今为止,只有少数人能够在 Linux 上开发应用程序,并且程序开发的周期也非常漫长。所以,使用传统开发工具开发基于 Linux 的应用程序是一件让开发人员头痛的事。传统的 Linux 开发工具不仅很难学会使用,而且它们的开发速度也远远不能达到开发人员的要求,使得 Linux 上应用程序的数量远远不及 Windows 平台上应用程序的数量,不能满足各个领域使用者的需求,造成了 Linux 不能迅速普及的情形。

对于 Linux 而言,要想进一步地应用和推广,必须拥有大量的应用程序的支持。基于 Linux 平台的开发人员需要一个简单、标准的开发环境以便能够快速实现应用程序的开发。而 Kylix 正是惟一满足这一要求的开发工具。Kylix 基于 Borland 公司现有的先进技术,并拥有与 Delphi 相同的开发模式,因此 Kylix 将为 Linux 带来大量的应用程序开发者,同时也将为 Delphi 和 Delphi 程序员带来一个福音——在 Windows 和 Linux 下进行跨平台的开发。

1.2.2 Kylix 与 Linux 下的其他开发工具的区别

Linux 下的其他开发工具,如 GCC/Emacs、MetroWerks Code Warrior 和 Cygnus Code Fusion 都很适合于系统级的开发,比如重新构建 Linux 的内核。而 Kylix 则是强调开发高性能的 Linux 应用程序。Kylix 开发环境完全摒弃了 Linux 下传统的“Compiler/Editor/Debugger”开发模式,从根本上简化了大量的复杂的应用程序技术,如 GUI、数据库、网络、Client/Server(客户机/服务器)和分布式对象技术。如果开发人员对操作系统的内核感兴趣,那么当前的这些工具可能是最适合的。但如果开发人员偏爱于开发应用程序,那么,Kylix 将是最佳的选择!

1.2.3 Kylix 对 Linux 的推进作用

随着 Borland Kylix 的发布,Linux 程序开发环境将得到显著的改进。

首先,Kylix 就像是 Windows 平台的 Delphi 或者是 Visual Basic 一样,提供了可视化

的集成开发环境。使程序员不需要像以前那样安装各种各样的编译器和调试器,所有的工作都集中在 Kylix 的环境中,这跟 Delphi 和 Visual Basic 是一样的。并且,用户可以根据自己的需要和操作习惯定制开发环境。可以多窗口显示结果,包括编辑代码、组件属性、调试窗口等。还可以用树状结构直观地列出类、组件、代码之间的关系。另外, Kylix 提供了程序员任务列表、Console 程序的制作模板、对象的属性、多对象的项目管理器等,还可以支持第三方的 API 工具。因此 Kylix 可以说是一种强劲的 RAD 工具。

其次, Kylix 隐藏了编写 Linux 应用程序的复杂性。它提供了 165 个可视和不可视的控件,并且都带有源程序,可以被重用和扩展。Kylix 的这些标准控件便于实现应用程序的快速开发,让程序开发人员可以更容易地创建菜单、按钮和链接数据库。

更值得一提的是, Kylix 不只拥有好看的外表,更使用了 Borland 的 cutting-edge 技术和 Delphi 最优秀的编译技术,使得利用 Kylix 开发的应用程序具有极高的速度。因此用 Kylix 编译的应用软件,比用解释性代码生成的软件快 5 倍~50 倍。尤其是在执行计算分析应用和大型的数据操作应用时,其高效性更加明显。

大多数使用 Borland Delphi 和 Microsoft Visual Basic 的用户将容易地将所掌握的技巧、知识和应用程序转移到 Linux 平台上来,快速地推动 Linux 成为主流操作系统。Kylix 的成功开发将成为 Linux 发展的里程碑,对 Linux 的发展起到巨大的推进作用。

1.3 Kylix 与 Delphi

1.3.1 Delphi 简介

接下来先介绍一下 Delphi。如果读者对此已非常熟悉,请跳过此节。

Delphi 使用 Object Pascal 语言,它是一种现代的,久经世故的程序语言。Delphi 基于传统的 Pascal 语言,并增加了类对象(单继承)、接口(和 Java 接口相似,用于实现多继承)、异常处理和内置的 Unicode Strings 支持。同样, Delphi 的编译器和优化器都能以非常快的速度将 Delphi 编译成为本地 X86 代码。

Delphi 的组件构架是面向对象的,并可以进行扩展。GUI 构造器使用户能够拖放窗体上的组件;对象监视器用于帮助用户检测和修改窗体上组件的属性,包括窗体本身的属性;用户还可以随意设置对象的标题、大小、字体、颜色等,而不用添加任何代码。

Delphi 使用户能够轻松地封装自己的组件类,并将它们集成到 IDE 中;用户也可以非常简便地开发新的组件或从现有的类派生出新的组件类;拖放用户定制的组件,与其他的开发人员共享组件。

Delphi 的窗体表现为 Form 类的一个实例。用户可以使用 Delphi 的标准窗体类 Tform 来编写自己的窗体类,也可以从其他的窗体派生出新的窗体类(这时子窗体类继承了父窗体的所有组件和行为)。使用事件句柄来实现窗体上组件的行为。比如,用户可以通过设置按钮的 OnClick 属性来定义自己何时点击了该按钮。Delphi 的集成开发环境将自动声明一个适当的方法,并将该方法定向到按钮的 OnClick 属性。用户可以通过定义 OnClick 来实现所需要实现的行为。

组件构架包括数据库访问组件。它可以自动绑定到数据库源的可视化控件和其他大量的组件,其中包括信息决策组件、报表组件、通用的对话框组件和第三方开发者提供的成千上万的其他组件等。

Delphi 的集成开发环境包括一个源程序编辑器、GUI 构造器和调试器。这些工具是被集成在一起的。比如,当用户拖放一个构件到窗体上时,GUI 构造器将显示出这个新的控件。同时,源代码编辑器自动把控件的信息添加到窗体类声明中。

Delphi 不仅支持 ActiveX、Com 和 OLE 技术,它还支持更多的可移植技术,如 CORBA。

1.3.2 Kylix 与 Delphi 的联系

1. Kylix 就是“Delphi for Linux”

如果用户使用过 Kylix,将发现 Kylix 的集成开发环境和 Delphi 的集成开发环境是一样的。更为难能可贵的是,Kylix 的使用方法和 Delphi 也是一样的。因此,用户可以将 Kylix 看成是 Linux 下的 Delphi。Kylix 有一组和 Windows 下的 Delphi 版本很相近的特征集,并依照现有的 Delphi 方案和编程技巧进行设计。尽管 Kylix 和 Delphi 的外观相同,但它们之间也有一定的差别。使用 Kylix 创建的是 Linux 应用程序和共享对象,而使用 Delphi 创建的是 Windows 应用程序和动态链接库 DLL。

2. Kylix 的跨平台性

对于专业程序员来说,要开发一个多平台的商业应用程序有很多种途径,但是它们实现起来都不太容易。所以,跨平台开发工具往往备受关注,Java 正是由于其与平台无关性博得了广大程序员的青睐。Kylix 开发者可以从 Delphi 已有的大量源代码开放的程序中获益,不过这些现有的资源也不是全部可用。由于 Delphi 一直就是一个单一平台的语言,有的代码只能在 Windows 下使用,而有的则直接对 PC 硬件进行操作,所以大部分 Delphi 应用程序和库在转到 Kylix 时都需要进行一些改动,有的甚至需要很大的改动。这些改动大多数都围绕着开发人员对 Win32API 的调用,或者是 Kylix 下第三方组件的可用性问题。虽然用户必须改变一些源代码,但用 Delphi 和 Kylix 构建跨平台的应用程序将会是最快,也是最简单的方法。

2001 年上半年,Borland 公司已经推出 Delphi 6.0,这是 Borland 公司的 Windows 平台开发工具,它同样是基于 CLX 的组件构架。因此,开发人员能够将 Kylix 下的源程序直接导入到 Delphi 中生成 Windows 平台下的代码,也可以将 Delphi 的源程序重新编译为 Linux 平台下 Kylix 的源代码。因此,通过 Delphi 6.0 和 Kylix,Borland 将帮助开发人员开发出具有平台独立性的产品,而不是局限于某个特殊的平台。

1.3.3 Kylix 与 Delphi 的区别

从集成开发环境上比较,Kylix 与 Delphi 最大的区别在于 Kylix 采用的是支持跨平台的 CLX 组件库,而 Delphi 采用的是 VCL 类库。CLX 与 VCL 有很多相似之处,但 CLX 不仅是一个可视化组件库,更是一个跨平台的组件库。Kylix 上所使用的支持跨平台的 CLX 组件库是基于 Qt 的,而 Delphi 采用的 VCL 类库是基于 Win 32 API 的。