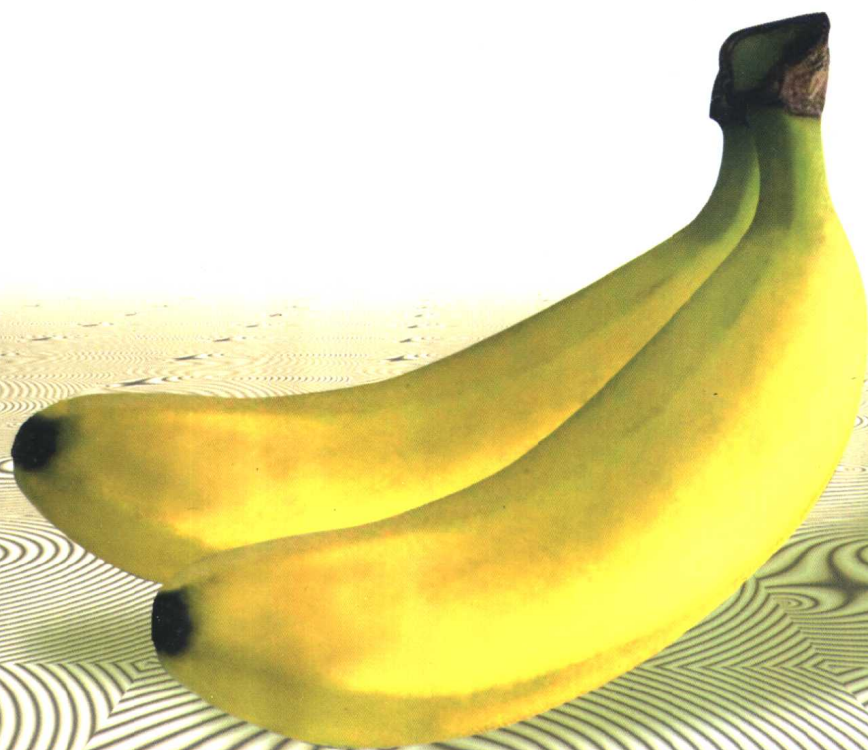


Linux 系统应用丛书

汪晓庆 等编著

KYLIX

应用开发与实例



机械工业出版社
China Machine Press

Linux 系统应用丛书

Kylix 应用开发与实例

汪晓庆 等编著



机械工业出版社

Kylix 是世界上第一个基于 Linux 平台的、完整的、高性能 RAD 开发工具，它极大地提高了 Linux 平台上 GUI 应用程序的开发效率，并对 Linux 世界产生了极其深远的影响。本书结合大量实例，使用通俗的语言深入浅出、清晰系统地介绍了 Kylix 应用开发环境与编程技巧，为使用 Kylix 开发功能强大的 Linux 应用提供了全面、充分的资源。

本书适用于希望快速掌握并精通 Kylix 编程的各层次的读者。

图书在版编目 (CIP) 数据

Kylix 应用开发与实例/汪晓庆等编著. —北京：机械工业出版社，2003.1

(Linux 系统应用丛书)

ISBN 7-111-11142-7

I. K... II. 汪... III. Linux 操作系统—软件工具, Kylix—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2002) 第 084373 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划：胡毓坚

责任编辑：田 梅

责任印制：路 琳

北京机工印刷厂印刷·新华书店北京发行所发行

2003 年 1 月第 1 版·第 1 次印刷

1000mm × 1400mm B5·16.375 印张·619 千字

0001-5000 册

定价：38.00 元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话：(010) 68993821、68326677-2527

封面无防伪标均为盗版

前 言

Linux 是当今最具竞争潜力的操作系统之一,其源码开放、性能可靠、易于升级以及价格低廉等特点使之迅速成为 Windows 操作系统的有力竞争对手。然而不得不承认,使用 Linux 传统工具进行应用程序的开发是非常困难的,它不像 Windows 已有的开发工具(如 VC++、VB、Delphi、C++ Bulider 以及 PowerBuilder 等)那样符合开发人员的习惯并具有极高的开发效率。Kylix 的诞生可以称作是一个里程碑,它极大地提高了 Linux 平台上 GUI 应用程序的开发效率,使 Win32 开发人员能够轻松地转向 Linux。

鉴于 Kylix 的优秀特性,我们在积累了大量开发经验的基础上编写了本书。本书的内容共分为五个部分。第一部分背景与基础知识。这部分内容使得用户能够全面了解 Kylix 产品的功能与特性并掌握 Kylix 编程语言。第二部分 Kylix 快速入门的最佳途径。通过阅读这一部分,用户可以轻松地掌握 Kylix 的安装与卸载、熟悉 Kylix 集成开发环境以及通过一个快速应用开发实例迅速地掌握 Kylix 程序设计。第三部分深入、详尽地介绍了 Kylix 数据库的应用与开发,这里用户可以看到对 db-Express 的详细描述,以及用户可用于访问数据的其他机制。这一部分也覆盖了使用数据访问组件与数据感知控制组件、创建与执行查询、读写数据、建立与显示主从关系等内容。第四部分介绍了一些前沿的 Kylix 应用与开发,包括多线程应用、Internet 应用以及跨平台应用的创建与开发。第五部分则贯穿本书内容,详细描述了两个 Kylix 综合应用开发实例,可帮助读者大幅度提高其实际应用开发能力。

本书由汪晓庆主编,参加编写的人员还有赵军锁、徐建新、王晓红、陆朝甫、党岗。另外,在本书的编写过程中还得到了符辉、张台华、王佳、宋崇实、党铮、庞平等同志的支持与帮助,在此表示衷心的感谢!

由于作者经验有限,书中可能存在一些纰漏,敬请广大读者不吝赐教、批评指正。

编 者

目 录

前言

第一部分 背景与基础知识

第 1 章 Kylix 基本知识	3	2.5.1 赋值语句	38
1.1 Kylix 发布的背景及意义	4	2.5.2 函数与过程调用语句	39
1.2 Kylix 产品分类	5	2.5.3 goto 语句	39
1.3 Kylix 功能与特点	6	2.5.4 复合语句	39
1.4 Kylix 与 Delphi	10	2.5.5 with 语句	40
1.5 小结	11	2.5.6 条件语句	41
第 2 章 Object Pascal 语言	13	2.5.7 case 语句	42
2.1 常量	14	2.5.8 循环语句	43
2.1.1 真实常量	14	2.6 过程与函数	45
2.1.2 类型常量	15	2.6.1 过程与函数的声明	45
2.2 变量	17	2.6.2 过程与函数的调用	47
2.2.1 初始化变量	18	2.6.3 调用习惯	48
2.2.2 动态变量	18	2.7 类与对象	48
2.2.3 线程变量	18	2.7.1 类类型(class types)	49
2.3 数据类型	19	2.7.2 域	50
2.3.1 简单类型	20	2.7.3 方法	51
2.3.2 字符串类型	24	2.7.4 类引用类型	54
2.3.3 结构类型	25	(class - reference type)	54
2.3.4 指针类型	30	2.7.5 接口类型(interface types)	54
2.3.5 过程类型	31	2.8 程序与单元	55
2.3.6 Variant 类型	32	2.8.1 程序结构及语法	55
2.4 表达式与运算符	33	2.8.2 单元结构及语法	56
2.4.1 表达式	33	2.8.3 简单程序示例	57
2.4.2 运算符	34	2.9 小结	61
2.5 Object Pascal 语句	38		

第二部分 Kylix 快速入门

第 3 章 Kylix 安装与卸载	65	3.2 正式安装	69
3.1 安装前准备	66	3.2.1 安装身份	69
3.1.1 硬件环境需求	66	3.2.2 Kylix 安装程序	70
3.1.2 软件环境需求	66	3.2.3 安装步骤	70
3.1.3 问题的解决	67	3.2.4 产品激活与注册	71

3.3 Kylix 产品升级	71	4.6.1 使用代码浏览器	102
3.4 卸载	71	4.6.2 定制代码浏览器	102
3.5 小结	71	4.7 项目管理器	103
第 4 章 Kylix 集成开发环境	73	4.8 项目浏览器	104
4.1 主窗口	75	4.9 对象观察器	104
4.2 菜单	75	4.10 窗体设计器	105
4.2.1 File 菜单	75	4.11 对象库	107
4.2.2 Edit 菜单	77	4.11.1 将对象库的项添加到	
4.2.3 Search 菜单	80	项目	107
4.2.4 View 菜单	81	4.11.2 将模板添加到对象库	108
4.2.5 Project 菜单	82	4.11.3 指定用户模板为默认	
4.2.6 Run 菜单	84	模板	108
4.2.7 Component 菜单	86	4.12 小结	110
4.2.8 Tools 菜单	87	第 5 章 Kylix 快速应用	111
4.2.9 Help 菜单	88	5.1 Kylix 应用程序开发步骤	112
4.3 工具栏	88	5.1.1 创建/打开 Project	112
4.3.1 标准工具栏	89	5.1.2 设计用户界面	112
4.3.2 显示工具栏	89	5.1.3 编写代码	114
4.3.3 桌面工具栏	90	5.1.4 编译、调试与运行项目	115
4.3.4 调试工具栏	91	5.1.5 制作发行版本	115
4.3.5 Internet 工具栏	91	5.2 快速应用程序开发实例	117
4.3.6 定制工具栏	92	5.2.1 开始一个新的应用程序	117
4.4 组件板	93	5.2.2 分析用户界面需求	117
4.4.1 使用组件板	93	5.2.3 创建文本编辑区域	118
4.4.2 框架	94	5.2.4 添加菜单及工具栏	
4.4.3 定制组件板	94	支撑组件	120
4.5 代码编辑器	96	5.2.5 添加菜单	126
4.5.1 代码编辑器快捷菜单	97	5.2.6 添加工具栏	128
4.5.2 使用代码编辑器浏览代码	99	5.2.7 编写事件句柄	129
4.5.3 在代码编辑器中获取帮助	99	5.2.8 创建一个 About 对话框	135
4.5.4 查看并编辑窗体代码	100	5.2.9 完善并运行应用程序	137
4.5.5 定制代码编辑器	101	5.3 小结	138
4.6 代码浏览器	102		

第三部分 Kylix 数据库应用开发

第 6 章 Kylix 数据库应用开发入门	141	设计	142
6.1 数据库应用开发基本步骤	142	6.2.1 数据库类型	142
6.2 Kylix 数据库应用结构		6.2.2 数据库应用结构	143

6.3 常用数据库应用开发	步骤	192
组件	8.2 TClientDataSet 组件	192
6.4 制作数据库应用发行	8.2.1 TClientDataSet 属性、方法	
版本	及事件介绍	193
6.5 一个简单的数据库应用	8.2.2 TClientDataSet 的使用	203
开发实例	8.3 TDataSetProvider 组件	203
6.5.1 开始一个新的应用	8.3.1 TDataSetProvider 属性、	
6.5.2 设置窗体属性值	方法及事件介绍	204
6.5.3 分析用户界面需求	8.3.2 TDataSetProvider 使用方法	
6.5.4 添加 ClientDataSet 与	与技巧	207
DataSource 组件	8.4 TXMLTransform 组件	207
6.5.5 创建数据库表记录显	8.4.1 TXMLTransform 属性、	
示区域	方法及事件介绍	207
6.5.6 创建一组数据库浏览	8.4.2 TXMLTransform 使用	
按钮	技巧	210
6.5.7 创建 MEMO 字段信息	8.5 TXMLTransformProvider	
显示区域	组件	212
6.5.8 创建 GRAPHIC 字段信息	8.5.1 TXMLTransformProvider 属性、	
显示区域	方法及事件介绍	212
6.5.9 创建应用程序关闭按钮 ..	8.5.2 TXMLTransformProvider	
6.5.10 运行应用程序	使用技巧	214
6.6 小结	8.6 TXMLTransformClient	
第7章 理解与使用数据集	组件	215
7.1 数据集的类型	8.6.1 TXMLTransformClient 属性、	
7.2 打开与关闭数据集	方法及事件介绍	215
7.3 设置数据集状态	8.6.2 TXMLTransformClient	
7.4 导航数据集	使用技巧	216
7.5 查找数据集信息	8.7 Provider 组件使用方法	
7.6 显示与编辑数据子集	与技巧	218
7.7 修改数据	8.8 小结	228
7.8 数据集事件	第9章 数据控制组件	229
7.9 小结	9.1 数据控制组件的公共	
第8章 数据访问组件	使用特性	230
8.1 TDataSource 组件	9.2 TDBGrid 组件	231
8.1.1 TDataSource 属性、方法及	9.2.1 TDBGrid 属性、方法及	
事件介绍	事件介绍	232
8.1.2 TDataSource 创建与使用	9.2.2 TDBGrid 应用实例	235

9.3 TDBNavigator 组件	237	9.12 TDBLookupListBox 组件	261
9.3.1 TDBNavigator 属性、 方法及事件介绍	237	9.12.1 TDBLookupListBox 属性、 方法及事件	262
9.3.2 TDBNavigator 应用实例 ..	239	9.12.2 TDBLookupListBox 应用 实例	263
9.4 TDBText 组件	240	9.13 TDBLookupComboBox 组件	265
9.4.1 TDBText 属性、方法及 事件介绍	240	9.13.1 TDBLookupComboBox 属性、 方法及事件	265
9.4.2 TDBText 应用实例	241	9.13.2 TDBLookupComboBox 应用 实例	266
9.5 TDBEdit 组件	243	9.14 小结	267
9.5.1 TDBEdit 属性、方法及 事件	243	第 10 章 dbExpress 组件	269
9.5.2 TDBEdit 应用实例	244	10.1 TSQLConnection 组件 ..	270
9.6 TDBMemo 组件	246	10.1.1 TSQLConnection 属性、 方法及事件	271
9.6.1 TDBMemo 属性、方法及 事件	246	10.1.2 TSQLConnection 使用 技巧	276
9.6.2 TDBMemo 应用实例	249	10.2 TSQLDataSet 等单向数据 集组件	281
9.7 TDBImage 组件	250	10.2.1 TSQLDataSet 等属性、方法 及事件介绍	283
9.7.1 TDBImage 属性、方法及 事件介绍	250	10.2.2 TSQLDataSet 等单向数据集的 使用方法与技巧	288
9.7.2 TDBImage 应用实例	251	10.3 TSQLMonitor 组件	302
9.8 TDBListBox 组件	251	10.3.1 TSQLMonitor 属性、方法 及事件介绍	302
9.8.1 TDBListBox 属性、方法及 事件	252	10.3.2 TSQLMonitor 的使用步骤 与方法	304
9.8.2 TDBListBox 应用实例	252	10.4 TSQLClientDataSet 组件	305
9.9 TDBComboBox 组件	254	10.4.1 TSQLClientDataSet 属性、 方法及事件介绍	305
9.9.1 TDBComboBox 属性、方法及 事件介绍	254	10.4.2 TSQLClientDataSet 使用方法 与技巧	308
9.9.2 TDBComboBox 应用实例 ..	255	10.5 小结	308
9.10 TDBCheckBox 组件	256		
9.10.1 TDBCheckBox 属性、方法及 事件	256		
9.10.2 TDBCheckBox 应用 实例	258		
9.11 TDBRadioGroup 组件	259		
9.11.1 TDBRadioGroup 属性、方法 及事件	259		
9.11.2 TDBRadioGroup 应用 实例	260		

第 11 章 理解与掌握字段组件	309	12.1.1 导航客户端数据集中 的数据	330
11.1 动态字段组件	310	12.1.2 限制记录的出现	333
11.2 持久性字段组件	311	12.1.3 表示主从关系	337
11.2.1 创建与删除持久性字段	312	12.1.4 约束数值	340
11.2.2 定义特殊持久性字段	313	12.1.5 使数据只读	340
11.2.3 设置持久性字段属性 与事件	315	12.1.6 编辑数据	341
11.3 字段组件方法	317	12.1.7 排序与索引	342
11.4 操作字段值	318	12.1.8 表示计算值	344
11.4.1 访问字段值	318	12.1.9 使用自维护合计	345
11.4.2 转换字段值	320	12.1.10 添加定制信息	346
11.4.3 检查字段当前值	321	12.2 从其他数据集复制数据	347
11.4.4 为字段设置默认值	321	12.3 使用带 Provider 的客户 端数据集	348
11.4.5 指定约束	321	12.4 使用 TSQLClient- DataSet	352
11.5 使用对象字段	322	12.5 使用基于文件数据的 客户端数据集	355
11.5.1 显示 ADT 与 Array 字段	322	12.5.1 创建一个新的数据集	355
11.5.2 使用 ADT 字段	323	12.5.2 从文件或流装入数据	356
11.5.3 使用 Array 字段	324	12.5.3 将修改信息合并入数据	357
11.5.4 使用 DataSet 字段	325	12.5.4 将数据保存至文件或流	357
11.5.5 使用 Reference 字段	326	12.6 小结	357
11.6 小结	327		
第 12 章 使用客户端数据集	329		
12.1 使用客户端数据集 操作数据	330		

第四部分 Kylix 前沿应用开发

第 13 章 多线程应用	361	第 14 章 Internet 应用	373
13.1 定义线程对象	362	14.1 Web Broker 与 WebSnap 技术特性	374
13.1.1 初始化线程	363	14.2 基础知识	375
13.1.2 编写线程函数	364	14.3 WWW 工作方式	377
13.1.3 编写清除代码	367	14.4 Web 服务器应用类型	378
13.2 线程同步	367	14.4.1 CGI stand-alone	378
13.2.1 避免同时访问	367	14.4.2 Apache DSO Module	378
13.2.2 等待其他线程	369	14.4.3 转换 Web 服务器应用 目标类型	379
13.3 执行线程对象	371	14.5 使用 Web Broker	379
13.4 调试多线程应用	372		
13.5 小结	372		

14.5.1 使用 Web Broker 创建 Web 服务器应用	380	14.9 调试服务器应用	431
14.5.2 Web Broker 应用结构	380	14.9.1 调试 CGI 应用	431
14.5.3 Web 调度器	382	14.9.2 调试 Apache DSO 应用	432
14.5.4 动作项	382	14.10 小结	433
14.5.5 访问客户端请求信息	385	第 15 章 跨平台应用	435
14.5.6 创建 HTTP 响应报文	387	15.1 Windows 与 Linux 环境区别	436
14.5.7 生成响应报文内容	388	15.2 Delphi/Windows 与 Kylix/Linux 的区别	438
14.5.8 在响应中使用数据库 信息	392	15.3 将 Windows 应用程序 移至 Linux	443
14.6 使用 WebSnap	395	15.3.1 移植方式	443
14.6.1 基本 WebSnap 组件	396	15.3.2 移植应用程序步骤	444
14.6.2 使用 WebSnap 创建 Web 服务器应用	399	15.3.3 编写可移植代码	445
14.6.3 高级 HTML 设计	402	15.3.4 一些无法移植的特性	449
14.6.4 登录支持	403	15.3.5 消息与系统事件	449
14.6.5 调度请求	411	15.3.6 在 Windows 与 Linux 间 共享源文件	450
14.7 使用 XML 文档	416	15.4 跨平台数据库应用	450
14.7.1 使用文档对象模型	417	15.4.1 将数据库应用程序 移植至 Linux	453
14.7.2 使用 XML 组件	418	15.4.2 更新 dbExpress 应用 程序数据	454
14.7.3 使用 Data Binding 向导 提取 XML 文档	420	15.5 跨平台 Internet 应用	456
14.8 使用 Web Services	423	15.6 小结	456
14.8.1 编写支持 Web Services 的 服务器	423		
14.8.2 编写 Web Services 客户端	430		

第五部分 Kylix 应用开发实例

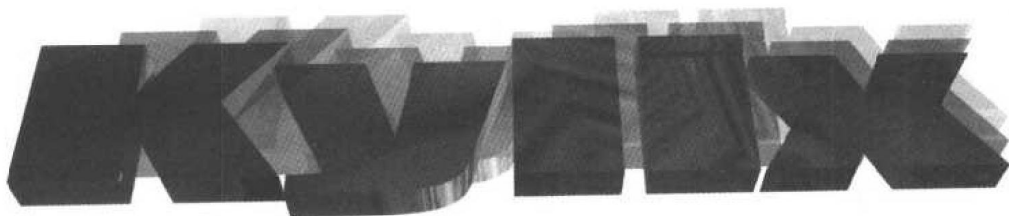
第 16 章 数据库应用开发实例	461	16.4.2 数据模块窗体设计	467
16.1 安装与卸载 InterBase 数据库	462	16.4.3 雇员管理窗体设计	470
16.1.1 系统需求	462	16.4.4 数据新增窗体设计	471
16.1.2 安装 InterBase 数据库	462	16.4.5 数据浏览与编辑窗 体设计	474
16.1.3 卸载 InterBase 数据库	463	16.4.6 雇员查询窗体设计	477
16.2 需求分析与方案设计	464	16.4.7 工资调整历史资料 查询窗体设计	478
16.3 创建新项目	466	16.4.8 错误消息窗体设计	480
16.4 窗体设计	466	16.5 程序代码编写	481
16.4.1 主窗体设计	466		

16.5.1 主窗体相关事件句柄 编写	481	16.7 小结	501
16.5.2 雇员管理窗体相关事件 句柄及过程编写	483	第 17 章 Web 应用开发实例	503
16.5.3 错误消息窗体相关事件 句柄编写	488	17.1 创建一个新项目	504
16.5.4 数据新增窗体相关事件 句柄编写	488	17.2 创建 EmployeeTable 页面 ...	505
16.5.5 数据浏览与编辑窗体相 关事件句柄编写	493	17.3 将数据组件添加至 Employee Table 页面模块	505
16.5.6 雇员查询窗体相关事件 句柄编写	495	17.4 创建用于显示数据的 网格	506
16.6 实例运行	497	17.5 添加编辑窗体	508
		17.6 添加错误报告	511
		17.7 运行 Web 应用实例	512
		17.8 小结	513

第 一 部 分

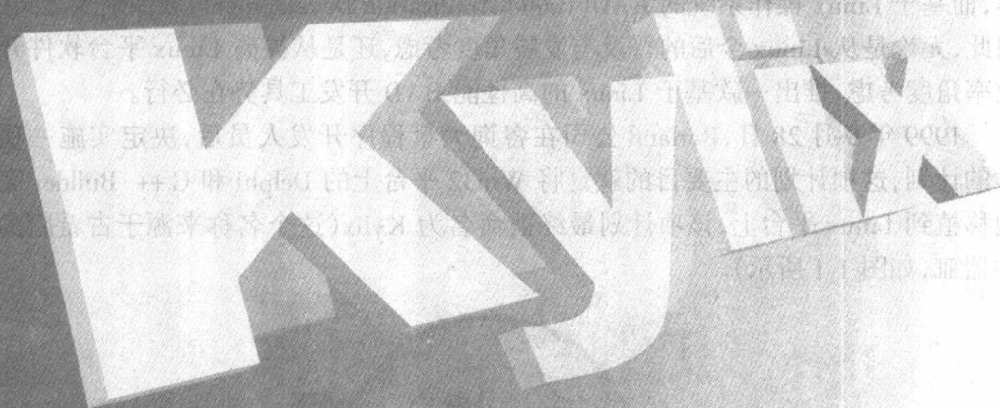
背景与基础知识

本部分内容主要介绍了 Kylix 的背景与基础知识,包括 Kylix 发布的背景及意义、Kylix 产品的分类及功能特点、Kylix 与 Delphi 的主要异同以及 Kylix 编程语言 Object Pascal 的语法以及主要特点。这部分内容能使读者全面了解 Kylix 产品并熟练掌握 Kylix 编程语言 Object Pascal,为 Kylix 编程奠定良好的基础。



第 1 章

Kylix 基本知识



本章要点:

- 什么是 Kylix
- Kylix 发布的背景及意义
- Kylix 功能与特点概述

1.1 Kylix 发布的背景及意义

Linux 是当今最具竞争潜力的操作系统之一,其源码开放、性能可靠、易于升级以及价格低廉等特点使之迅速成为 Windows 操作系统的有力竞争对手。在这种情况下,市场对 Linux 应用软件的需求也空前增长。然而不得不承认,使用 Linux 传统工具进行应用程序的开发是非常困难的,它不像 Windows 已有的开发工具那样符合开发人员的习惯并具有极高的开发效率。相比之下,基于 Windows 的可视化集成开发环境已拥有许多成熟的产品,如 VC++、VB、Delphi、C++ Builder 和 PowerBuilder 等,而基于 Linux 操作系统的 RAD(Rapid Application Development)软件则少之又少。因此,无论是从 Linux 今后的普及与发展角度考虑,还是从提高 Linux 平台软件开发效率角度考虑,推出一款基于 Linux 的高性能 RAD 开发工具势在必行。

1999 年 9 月 28 日,Borland 公司在咨询大量程序开发人员后,决定实施一项庞大的计划,这项计划的主要目的就是将 Win32 平台上的 Delphi 和 C++ Builder 完整地移植到 Linux 平台上,该项计划最终被命名为 Kylix(这个名称来源于古希腊的一种器皿,如图 1-1 所示)。

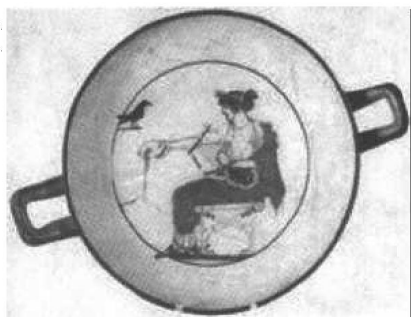


图 1-1 Kylix 名称的来源——古希腊的一种器皿

2001 年 1 月 31 日,Borland 公司在美国纽约 LinuxWorld 2001 大会上正式发布了 Kylix。同年 10 月,Borland 公司又推出了 Kylix2 以加速 Linux 平台上电子商务应用的开发。2002 年 8 月,Borland 公司又宣布推出同时支持 C++ 与 Delphi 语言(Object Pascal)的 Kylix3。Kylix 是世界上第一个基于 Linux 平台的完整的、高性能 RAD 开发工具,也是惟一个将 RAD 基于组件的可视化桌面、数据库以及 Internet 应用开发优点与 GNU/Linux 操作系统快速原生代码编译器的强大功能相结合的开发平台,其重要意义不亚于 Borland 推出的世界上第一个 IDE(集成开发环境)——Turbo Pascal。正是以极为出色的 Pascal 编译器为基础的 Delphi 为 Borland 在全球赢得了众多开发者的支持,并成为 Win32 平台最主要的开发工具之一。Kylix 即是 Delphi 在 Linux 平台上的具体实现,它将极大地提高 Linux 平台上 GUI(图形用户界

面)应用程序(尤其是商业和企业的数据库应用)的开发效率。数以万计技术熟练的 Win32 开发人员将能够轻松转向 Linux,难以计数的 Win32 商业应用将被移植到 Linux 平台上。

Kylix 的产生,可以说是 Linux 发展的结果。对于开发人员而言,它的出现将极大地解放生产力;对于 Linux 的爱好者而言,它的出现将丰富 Linux 上的应用程序;对于 Borland 公司而言,它是 15 年以来最重要的项目之一;对于 Linux 世界而言,它的出现将会深深影响其今后的发展。

1.2 Kylix 产品分类

Kylix 的产品主要有三个版本:

(1) 专业版本。该版本是为专业开发人员设计的,提供了 ANSI/ISO C++ 与 Delphi 语言(Object Pascal)两种功能强大的面向对象语言编程。借助该版本,用户可以快速、轻易地创建高性能的 GUI、数据库、Web 应用以及可以跨平台的各种应用,通过使用 XML、XSL、SOAP 以及 WSDL,可以构造符合工业标准的 Web 服务客户端。该版本通过高速数据库驱动 dbExpress 提供了对本地数据库 MySQL、Borland InterBase、RedHat Database 以及 PostgreSQL 的完全支持。

(2) 企业版本。企业版本包含了所有主要的功能与特性,该版本提供了 ANSI/ISO C++ 与 Delphi 语言(Object Pascal)两种功能强大的面向对象语言编程。借助该版本,用户可以快速、轻易地创建高性能的 GUI、数据库、Web、Web 服务应用以及可以跨平台的各种应用。该版本的 BizSnap 借助 Web 服务和 XML 技术拓展了 Linux 和 Apache 的电子商务能力,简化了跨越不同平台的全球客户、供应商、商业合作伙伴以及雇员之间电子商务与 Web 服务的无缝集成,基于组件的 Web 应用开发平台 WebSnap 可以将 Apache Web 服务器转化成强大的企业级、数据库驱动的 Web 应用服务器。另外,DataSnap 为创建多层、企业级的数据库解决方案提供了可扩展的中间件。该版本提供了对 Oracle、DB2、Informix、InterBase、MySQL、RedHat Database 以及 PostgreSQL 的高速数据库驱动。

(3) 开放版本。该版本提供了 ANSI/ISO C++ 与 Delphi 语言(Object Pascal)两种功能强大的面向对象语言编程,利用其可视化的开发环境以及 FreeCLX™组件框架,用户可以快速地创建功能强大、高性能的 GUI 应用。该版本可以从 Borland 免费下载,适用于仅仅基于开放源码(Open Source)进行应用开发的程序开发人员。基于该版本所开发的应用必须遵循 GPL(GNU General Public License,一个免费软件分发许可)。开放版本只提供 GPL 许可的 CLX,该 CLX 只能用于开发 GPL 应用。

除了以上三个主要的版本之外,用户还可以获取 Kylix 的试用版本与教育版本,这两个版本有特殊的许可与配置限制。

1.3 Kylix 功能与特点

在 Kylix 发布之前, Linux 已经拥有了一些 IDE, 如 JX、KDevelop 等, 也有一些界面生成工具, 如 GLADE(gtk+ 界面生成器)等, 但没有一个可以和 Kylix 相提并论, 就功能来讲, Kylix 是空前的。Kylix 集成了目前世界上最快速的编译器、可视化的 GUI 环境、Web 和数据库应用软件的开发环境, 其主要功能与特点可概括为以下几个方面。

1. 高性能原生 Linux 应用程序

在进行复杂计算或处理大规模数据集记录时, 代码的执行速度变得非常重要。Kylix 高性能、最优化的 32 位原生 C/C++ 与 Object Pascal 编译器可直接生成机器代码, 而不是低效率地解释执行代码, 因此 Kylix 的应用程序比使用解释执行的应用程序快 5~50 倍。

Kylix 为开发快速且易于发布与维护的高效应用程序提供了解决方案。使用 Kylix 可以编译动态装载包(Dynamically Loaded Packages), 从而在数量级上减小执行程序的大小。动态装载包是一种独特的、居于领先地位的编译技术, 能够把应用程序分成独立的、可重用的编译模块, 并保证应用程序的高效执行。动态装载包使应用程序模块化, 使代码在需要执行的时候才调入内存, 这种机制使主要执行代码精简快速。当进行功能扩展时, 只需将那些要改进的模块进行替换即可。

Kylix 可以轻松创建可重用的共享对象库和独立的可执行程序。Kylix 只需编写一次共享对象代码, 就能在任何可使用共享对象(.so)的应用程序或语言中重复使用。开发人员能创建独立于运行时的可执行程序, 而且可以自由分发, 不再另外增加分发成本。

Kylix 使用编译指示支持内联汇编程序, 因此可以在 Object Pascal/C++ 与汇编代码之间进行轻松切换, 方便使用最新处理器技术。

2. 中性的窗口管理器

Kylix 既可以在 KDE 与 Gnome 下工作, 也可以在其他窗口管理器(如 MWM、WindowMaker 等)下工作。Kylix 不针对或依赖特定的窗口管理器, 也不支持 KDE、Gnome 桌面环境的专有特性。考虑到 Borland 已经宣布支持 Gnome, Kylix 将来的版本可能会支持 Gnome 的专有特性, 图 1-2 为 Gnome 下的 Kylix 界面图。

3. 快速集成应用开发环境(IDE)

在开发应用程序时, 开发人员所需要的全部功能都汇集在一个高度集成的开发环境中, 该环境包括当今最优秀的代码编辑器、调试器和跟踪器, 以及更易理解的错误提示和高亮色的句法显示。完全可定制集成开发环境和可停驻、可浮动工具条, 使开发人员能够定制极具个人风格的工作环境, 从而提高开发效率。在高版本的 Kylix 中, 用户还可以根据需要进行选择分别使用不同语言的 IDE, 即 Delphi IDE(使用 Object Pascal 语言)或者 C++ IDE(使用 C++ 语言)。