

Linux 下的 Delphi

——Kylix 使用与开发

姜玉琴 谢松县 编著

北京大学出版社
http://cbs.pku.edu.cn

Linux 下的 Delphi

——Kylix 使用与开发

姜玉琴 谢松县 编著

北京 大学出版社

北 京

内 容 提 要

本书是一本 Kylix 入门教材,对 Kylix 的基础编程进行了详细的介绍。全书共分为 16 章,分别介绍 Linux 以及 Kylix 基础知识、文件管理应用程序的设计、对象 Pascal 语言、数据库应用程序的设计、网络应用程序设计基础和插口(Socket)的使用、Kylix 自定义部件的开发等内容;附录部分介绍了最常用的 Interbase 数据库。

本书不仅可以作为广大编程爱好者的指导书,也可以作为高等院校师生教学、自学的参考书。

图书在版编目(CIP)数据

Linux 下的 Delphi——Kylix 使用与开发/姜玉琴等编著. —北京:北京大学出版社,2002.5
ISBN 7-301-05199-9

I. L… II. ①姜… ②谢… III. Kylix——程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 053640 号

书 名: Linux 下的 Delphi——Kylix 使用与开发

著作责任者:姜玉琴 谢松县

责任编辑:王 妍

标准书号:ISBN 7-301-05199-9/TP·0596

出 版 者:北京大学出版社

地 址:北京市海淀区中关村北京大学校内 100871

电 话:编辑部(010)62765013 发行部(010)62750672

网 址: <http://cbs.pku.edu.cn>

电子信箱: xxjs@pup.pku.edu.cn

印 刷 者:河北省滦县滦兴书刊印刷厂

发 行 者:北京大学出版社

经 销 者:新华书店

787 毫米×1092 毫米 16 开本 19.75 印张 483 千字

2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

定 价:29 元

前 言

2001年1月31日, Borland公司正式发布了Kylix,它是世界上第一个运行在Linux平台上完整的RAD(快速应用程序)开发工具,其重要意义不亚于当年Borland推出的世界上第一个IDE(集成开发环境)——Turbo Pascal。Delphi是以Pascal编译器为基础的,它为Borland在全球赢得了众多开发者的支持,并成为Win 32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应用程序(尤其是商业和企业的数据库应用)的开发效率,使技术熟练的Win 32开发人员能够轻松转向Linux,而且非常多的Win 32商业应用将可能被移植到Linux平台上。

Kylix 1.0是一个功能强大的可视化编程环境,它为我们提供了一种方便、快捷的Linux应用程序开发工具。因为Kylix声称要支持Gnome,所以Kylix使用了图形界面的许多设计思想,采用了灵活的面向对象Pascal程序语言。Kylix使得Linux下的编程不再是枯燥无味的工作。对于广大的程序开发人员来讲,使用Kylix 1.0开发应用软件和数据库应用程序,会大大地提高编程效率。Kylix使得Windows上运行的许多程序可以移植到Linux上运行,与此相对应,用Kylix写成的Linux程序也可以在Windows上运行。

Kylix无疑是当前在Linux下开发应用程序最好的、最先进的编程环境。Kylix 1.0首先是一个应用程序框架。它的思想就是开发一个骨架应用程序,然后再加入一些短小有用的功能模块。Kylix 1.0已经为我们做好了一切基础工作,我们工作的主要精力就可以放在具体的功能实现上了,其每项功能都展示了一个实际程序设计环境的主流方法。此外,Kylix 1.0把编程的回调,句柄的处理等繁复过程都放在一个不可见的覆盖物下面,隐藏了编制Linux软件的复杂性,编程人员可以使用CLX部件更容易地制作选单、按钮和链接数据库以及设计出其他的特征。Kylix能让一般用户用上越来越多的优秀Linux应用软件。

Kylix 1.0采用了面向对象的开发软件的思想,在学习使用Kylix的过程之中,读者可以发现一些完善的软件工程技术,并可以在自己的程序中实际地运用它们。

“边学边做”是学习一种编程工具最有效的方法,读者在阅读本书的过程中应该上机实践。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,这将会使读者体验到进步和成功的乐趣。

本书由姜玉琴、谢松县主编。另外,阮小明、李志拥、吴中、毛兵兵、晏英、陈维伟、吴其、李向军、李燕杰、钱乾、狄荣、李林、谢志光、陈杰、王别芳、徐燕娥、戴凡等也参加了本书的编写工作。

由于时间仓促,书中错误、不当之处在所难免,希望读者批评指正。如需进一步交流,请发邮件至esky@tech.com.cn。

编 者

2002年4月

责任编辑: 王 妍

封面设计: 项 尧

Linux 下的 Delphi

——Kylix 使用与开发

本书是一本 Kylix 入门教材, 对 Kylix 的基础编程进行了详细的介绍。全书共分为 16 章, 分别介绍 Linux 以及 Kylix 基础知识、文件管理应用程序的设计、对象 Pascal 语言、数据库应用程序的设计、网络应用程序设计基础和插口 (Socket) 的使用、Kylix 自定义部件的开发等内容; 附录部分介绍了最常用的 Interbase 数据库。

本书不仅可以作为广大编程人员的指导书, 也可作为高等院校师生教学、自学参考书。

ISBN 7-301-05199-9



9 787301 051993 >

ISBN 7-301-05199-9/TP · 0596

定价: 29 元

北京大学出版社

地址: 北京市海淀区中关村北京大学校内

邮编: 100871

电话: (010)62765013 (编辑部)

(010)62750672 (发行部)

E-mail: xxjs@pup.pku.edu.cn

<http://cbs.pku.edu.cn>

目 录

第1章 Linux 操作系统与 Kylix 简介	1
1.1 Linux 操作系统简介	1
1.2 为什么使用 Linux 操作系统	2
1.3 Kylix 简介	3
1.3.1 Kylix 概述	3
1.3.2 Kylix 的主要特点	5
1.4 安装 Linux 之前的准备工作	6
1.4.1 了解 Linux 操作系统对硬件设备的要求	6
1.4.2 确定硬件设备是否正确	7
1.4.3 考虑安装 Kylix 的设备要求	7
1.5 本章小结	8
第2章 Linux 操作系统的安装和配置	9
2.1 Linux 的安装	9
2.1.1 安装准备	9
2.1.2 开始安装	13
2.1.3 安装 X 文件	15
2.1.4 关闭系统	15
2.2 Linux 系统的配置	16
2.2.1 配置鼠标	16
2.2.2 配置 X Window	16
2.2.3 配置 LILO	17
2.2.4 配置声卡	18
2.2.5 配置网络	18
2.2.6 配置时钟	18
2.2.7 配置打印机	18
2.3 本章小结	19
第3章 Linux 操作系统基本命令	20
3.1 使用 man 命令获取帮助	20
3.2 浏览及搜索文件系统	21
3.2.1 使用 cd 命令进入不同的目录	22
3.2.2 使用 pwd 命令查看当前所在的位置	22
3.2.3 使用 find 命令在子目录中搜索匹配的文件	22

3.2.4 使用 whereis 命令查找文件	24
3.2.5 使用 locate 命令定位文件	24
3.2.6 使用 whatis 和 apropos 命令获取命令简介	25
3.3 阅读子目录和文件	25
3.3.1 使用 ls (dir) 命令列出子目录的内容清单	26
3.3.2 使用 cat 命令查看文件或合并文件	27
3.3.3 使用 more 命令逐屏阅读文件	27
3.3.4 使用 less 命令浏览文件	29
3.3.5 使用 head 或 tail 命令阅读文件的开头和结尾	29
3.3.6 使用其他一些文件操作命令	30
3.4 在 Linux 系统中安装新软件	32
3.4.1 概述	32
3.4.2 寻找软件包	33
3.4.3 确定源代码是否可以在机器上通过编译并运行	33
3.4.4 替换制作文件	34
3.4.5 编译源代码	34
3.4.6 安装二进制代码	35
3.4.7 把软件包添加到用户的 PATH 语句中	35
3.4.8 识别可执行程序	35
3.5 本章小结	36
第 4 章 Kylix 的安装与启动	37
4.1 Kylix 入门	37
4.2 安装 Kylix 前的准备	37
4.2.1 Kylix 软件的获得	37
4.2.2 Kylix 的安装准备	37
4.2.3 安装前测试	38
4.2.4 安装 LIBJPEG	40
4.2.5 安装 GLIBC	40
4.3 Kylix 的安装	42
4.3.1 Kylix 要求	42
4.3.2 根安装与普通 (非根用户) 安装比较	43
4.3.3 Kylix 安装程序	43
4.3.4 安装过程	43
4.3.5 库相关性	44
4.3.6 标准 C++ 库问题	45
4.3.7 对象库	45
4.3.8 卸载 Kylix	45
4.4 启动 Kylix	46
4.5 本章小结	46

第5章 Kylix 1.0 集成开发环境	47
5.1 Kylix 1.0 集成开发环境的组成	47
5.2 菜单栏及其使用	48
5.2.1 File 菜单	48
5.2.2 Edit 菜单	49
5.2.3 Search 菜单	51
5.2.4 View 菜单	52
5.2.5 Project 菜单	53
5.2.6 Run 菜单	55
5.2.7 Component 菜单	56
5.2.8 Tools 菜单	56
5.2.9 Help 菜单	57
5.3 工具条及使用	58
5.4 部件板	59
5.5 窗体设计器	59
5.6 对象检查器及其使用	61
5.6.1 对象检查器的组成	61
5.6.2 使用对象检查器设置属性	61
5.6.3 为部件对象添加事件	62
5.6.4 使用对象检查器的上下文菜单	62
5.7 代码编辑器及其使用	63
5.7.1 代码编辑器的组成	63
5.7.2 代码洞察 (Code Insight)	64
5.7.3 代码浏览 (Code Browser)	65
5.7.4 类完善	65
5.7.5 使用书签	65
5.7.6 使用代码编辑器的上下文菜单	66
5.8 代码管理器 (Code Explorer) 及其使用	67
5.8.1 代码管理器的组成	67
5.8.2 代码管理器的使用	67
5.8.3 代码管理器的上下文菜单	68
5.9 本章小结	68
第6章 文件管理应用程序设计	70
6.1 文件管理设计基本思路	70
6.1.1 窗口设计	70
6.1.2 界面设计	70
6.1.3 子窗口的创建、布置和关闭	72
6.2 文件控件的联系	73
6.3 DriveTabSet 的自画风格显示	73

6.3.1 设置自画风格	73
6.3.2 向字符串链表添加图形对象	73
6.3.3 画出自画项目	74
6.4 文件管理基本功能的实现	75
6.4.1 文件打开	75
6.4.2 文件拷贝、移动、删除、更名	76
6.4.3 一致的界面	78
6.4.4 显示文件属性	79
6.5 其他文件管理功能的实现	81
6.5.1 文件查找	81
6.5.2 显示磁盘信息	82
6.5.3 改变显示文件的类型	83
6.5.4 目录管理功能的实现	83
6.5.5 创建目录	83
6.5.6 删除目录	84
6.6 为部件增添一个方法	84
6.7 一些问题的处理	87
6.7.1 子窗口的标题	87
6.7.2 状态条的显示	87
6.7.3 版本信息	88
6.7.4 菜单项的变灰与使能	88
6.7.5 可重用的文件处理模块	89
6.8 本章小结	89
第 7 章 对象 Pascal 语言基础知识	90
7.1 保留字和标识符	90
7.1.1 保留字和标准指令字	90
7.1.2 标识符	91
7.2 对象 Pascal 中的数据类型	92
7.2.1 简单数据类型	93
7.2.2 字符串类型	98
7.2.3 结构类型	99
7.2.4 指针类型	114
7.2.5 过程类型 (procedural)	116
7.2.6 可变类型 (variant)	118
7.2.7 其他类型	120
7.3 注释	120
7.4 运算符和表达式	121
7.4.1 运算符	121
7.4.2 表达式	123

7.5 语句	124
7.5.1 声明语句	124
7.5.2 赋值语句	124
7.5.3 goto 语句	125
7.5.4 复合语句	125
7.5.5 with 语句	125
7.5.6 if 语句	126
7.5.7 case 语句	126
7.5.8 repeat 语句	126
7.5.9 while 语句	127
7.5.10 for 语句	127
7.6 单元	127
7.6.1 对象 Pascal 的库单元 Unit	128
7.6.2 对象 Pascal 程序库单元的结构	128
7.7 类与对象	129
7.7.1 什么是对象	129
7.7.2 对象的封装性	129
7.7.3 对象的继承性	130
7.7.4 对象的多态性	130
7.8 本章小结	130
第 8 章 使用 Kylix 开发数据库应用程序	131
8.1 数据库系统概述	131
8.2 数据库应用程序的体系结构	132
8.3 Kylix (Server) 支持的数据库类型	133
8.3.1 Interbase	134
8.3.2 MySQL	135
8.3.3 DB2	135
8.3.4 Oracle	135
8.4 数据库访问控制部件及其应用	136
8.4.1 数据访问 (Data Access) 部件	136
8.4.2 数据库表达 (dbExpress) 部件	137
8.4.3 数据控制 (Data Control) 部件	138
8.5 Kylix 中的 SQL	138
8.5.1 SQL 语言简介	138
8.5.2 Kylix 的 SQL 语言的基本应用	139
8.6 本章小结	140
第 9 章 数据控制部件的使用	141
9.1 数据控制部件的基本特征	141
9.1.1 将数据控制部件和数据集关联起来	142

9.1.2 编辑更新数据	143
9.1.3 启动和关闭数据显示	144
9.1.4 刷新数据显示	145
9.1.5 启动鼠标、键盘和定时器事件	145
9.2 如何组织数据	145
9.2.1 显示单个记录	145
9.2.2 显示多个记录	150
9.3 使用 TDBGrid 显示、编辑数据	151
9.3.1 在缺省状态下使用网格控制	151
9.3.2 显示复合字段	152
9.3.3 设置 grid 选项	152
9.3.4 在网格内编辑	154
9.3.5 控制网格绘图属性	154
9.3.6 对运行时用户行为的响应	154
9.4 导航、操作记录	155
9.5 本章小结	156
第 10 章 使用无向数据集	157
10.1 指定数据显示	157
10.1.1 显示查询结果	158
10.1.2 在表中显示记录	160
10.1.3 显示存储过程结果	160
10.2 取数据	163
10.2.1 准备数据集	163
10.2.2 取多个数据集	164
10.3 执行无返回记录的命令	164
10.3.1 指定要执行的命令	164
10.3.2 执行命令	164
10.3.3 创建和修改服务器元数据	165
10.4 建立主/明细关系	166
10.4.1 建立 TSQLDataSet 或 TSQLQuery 的主/明细关系	166
10.4.2 建立 TSQLTable 的主/明细关系	167
10.5 访问主题信息	168
10.5.1 在使用数据集之后取数据	168
10.5.2 元数据数据集的结构	169
10.6 本章小结	171
第 11 章 链接到数据库	172
11.1 控制链接	172
11.1.1 描述服务器链接	172
11.1.2 打开和关闭服务器链接	174

11.2 控制服务器登录.....	175
11.3 管理事务.....	177
11.3.1 开始事务.....	177
11.3.2 终止事务.....	178
11.3.3 规定事务独立级别.....	178
11.4 访问服务器元数据.....	179
11.5 使用相关数据集.....	180
11.5.1 不断开与服务器的链接而关闭数据集.....	180
11.5.2 叠代穿过相关数据集.....	180
11.6 发送命令到服务器.....	181
11.7 调试数据库应用程序.....	182
11.7.1 使用 TSQLMonitor 来监视 SQL 命令.....	182
11.7.2 使用回叫来监视 SQL 命令.....	183
11.8 本章小结.....	184
第 12 章 使用客户数据集 (Client DataSets)	185
12.1 使用客户数据集处理数据.....	185
12.1.1 在客户数据集中导航数据.....	185
12.1.2 限制出现的记录.....	188
12.1.3 显示主/明细关系.....	191
12.1.4 约束数据值.....	194
12.1.5 将数据设置为只读的.....	194
12.1.6 编辑数据.....	195
12.1.7 排序和索引.....	196
12.1.8 表示计算值.....	199
12.1.9 使用维护聚合 (maintained aggregates)	199
12.1.10 给数据添加特定应用信息.....	201
12.2 从另一个数据集拷贝数据.....	201
12.2.1 直接赋值.....	202
12.2.2 克隆客户数据集光标.....	202
12.3 和提供商一起使用客户数据集.....	203
12.3.1 指定一个数据提供商.....	203
12.3.2 从数据源获取参数.....	204
12.3.3 传递参数到源数据集.....	204
12.3.4 指定在服务器上执行的命令.....	205
12.3.5 从源数据集请求数据.....	206
12.3.6 更新记录.....	207
12.3.7 刷新记录.....	209
12.3.8 使用客户事件与提供商部件通讯.....	209

12.4 使用 SQL 客户数据集	210
12.4.1 何时使用 TSQLClientDataSet	210
12.4.2 建立一 SQL 客户数据集	211
12.4.3 配置内部提供商部件	212
12.5 和基于文件的数据一起使用客户数据集	212
12.5.1 创建一个新数据集	213
12.5.2 从文件或流中载入数据	215
12.5.3 合并变化到数据	215
12.5.4 保存数据到文件或流	215
12.6 本章小结	216
第 13 章 使用提供商 (Provider) 部件	217
13.1 决定数据源	217
13.2 与客户数据集通讯	218
13.3 选择更新方法	219
13.4 控制数据包中包含哪些数据	219
13.4.1 指定数据包中有哪些字段	219
13.4.2 设置影响数据包的选项	219
13.4.3 向数据包中添加客户信息	220
13.5 对客户数据请求的响应	221
13.6 对客户数据更新的响应	222
13.6.1 在更新数据库之前编辑 delta 包	222
13.6.2 如何影响更新	223
13.6.3 显示单个更新	224
13.6.4 在提供商部件中解决更新错误问题	224
13.6.5 在不代表单个表格的数据集上应用更新	225
13.7 对客户引发事件的响应	225
13.8 本章小结	225
第 14 章 创建 Internet 服务器应用程序	226
14.1 术语与标准	226
14.1.1 URL (Uniform Resource Locator, 统一资源定位符) 的组成	226
14.1.2 HTTP 请求头信息	227
14.2 HTTP 服务器行为	227
14.2.1 形成客户请求	227
14.2.2 处理 CGI 请求	228
14.2.3 处理动态共享对象请求	228
14.3 Web 服务器应用程序	228
14.3.1 Web 服务器应用程序的类型	229
14.3.2 创建 Web 服务器应用程序	229
14.3.3 Web 模块	229

14.3.4 Web 应用程序对象	230
14.4 Web 服务器应用程序的结构	230
14.5 Web 调度程序	231
14.5.1 给调度程序添加行为	231
14.5.2 调度请求信息	231
14.6 行为项	232
14.6.1 确定触发行为的时机	232
14.6.2 用行为项响应请求消息	233
14.7 访问客户请求信息	234
14.7.1 包含请求头信息的属性	234
14.7.2 HTTP 请求消息的内容	235
14.8 创建 HTTP 响应消息	236
14.8.1 填写响应头	236
14.8.2 设置响应内容	237
14.8.3 发送响应	237
14.9 生成响应消息的内容	237
14.9.1 使用页面生成器 (page producer) 部件	238
14.10 在响应中使用数据库信息	241
14.10.1 在 Web 模块中加入会话	241
14.10.2 在 HTML 中标识数据库信息	241
14.11 调试服务器应用程序	243
14.11.1 调试 CGI 应用程序	243
14.11.2 调试 Apache DSO 应用程序	244
14.12 本章小结	245
第 15 章 使用插口 (Sockets)	246
15.1 实现的服务	246
15.2 Socket 链接的类型	247
15.2.1 客户链接	247
15.2.2 侦听链接	247
15.2.3 服务器链接	247
15.3 描述 Socket	247
15.3.1 描述主机	248
15.3.2 使用端口	248
15.4 使用 Socket 部件	249
15.4.1 使用客户 Socket	249
15.4.2 使用服务器 Socket	250
15.5 响应 Socket 事件	250
15.5.1 错误事件	251
15.5.2 客户事件	251

15.5.3 服务器事件	251
15.6 通过 Socket 链接读写	252
15.6.1 非阻塞链接	252
15.6.2 阻塞链接	253
15.7 本章小结	253
第 16 章 Kylix 自定义部件开发	254
16.1 建立部件概述	254
16.1.1 跨平台部件库 CLX	254
16.1.2 部件和类	254
16.1.3 建立部件过程概述	255
16.1.4 建立新部件的方法	256
16.1.5 测试未安装的部件	259
16.2 编写部件的面向对象技术	261
16.2.1 建立部件	261
16.2.2 控制部件的访问	262
16.2.3 派送方法	264
16.3 Kylix 部件编程	266
16.3.1 创建属性	266
16.3.2 创建事件	272
16.3.3 创建方法	277
16.4 使部件在设计时可用	278
16.4.1 注册部件	279
16.4.2 添加 Component Palette 上的位图	280
16.4.3 添加属性编辑器	280
16.5 部件编程实例	284
16.5.1 创建数据库相关的日历控制 - TDBCcalendar	284
16.5.2 创建图形部件	289
16.6 本章小结	297
附录 interbase 简介	298

第 1 章 Linux 操作系统与 Kylix 简介

1.1 Linux 操作系统简介

Linux 操作系统是可以运行在许多不同类型的计算机上的一种操作系统的“内核”。它是提供命令行或者程序与计算机硬件之间接口的软件的核心部分。Linux 操作系统内核管理以下事情：比如内存的分配使用及回收、采用什么方法以及在什么时候打开或者关闭文件、哪一个进程或者程序可以获得计算机的 CPU 等等。

Linux 操作系统可以说是 UNIX 操作系统的一个克隆体，它最初是在 1991 年 10 月 5 日由它的作者 Linus Torvalds 于赫尔辛基大学发布的，Linux 的发音与其作者的名字的发音相同，为 Li-nucks。Linux 操作系统支持所有现代操作系统应该具备的功能，它是作为版权软件发行的，由于它遵守 GNU 组织的一般公共许可协议（General Public License，简称 GPL），因此也是一个“copyleft”软件。基本上来说，这个许可协议允许我们修改或者分发 Linux 操作系统的拷贝，甚至允许销售它获利，但前提是必须承认原作者的版权并提供源代码。

Linux 操作系统继承了 UNIX 操作系统超过二十五年的经验、源代码以及技术支持。Linux 操作系统是 UNIX 操作系统的一个版本，它的源头是 BSD 和 System V 这两种久经考验的 UNIX 操作系统变体，它的核心其实就是一个面向命令行的操作系统。这使得 Linux 操作系统虽然是最新操作系统中的一员，却远比当今流行的商业化的操作系统成熟。

Linux 操作系统还有不同的版本适用于非 Intel 芯片的计算机，比如 Sun 微系统公司 Microsystems（SPARC 体系结构）或者那些使用其他类型中央处理单元如 Alpha 芯片的计算机。如果想了解更多关于用在非 x86 体系结构计算机上的 Linux 操作系统发行版本的资料，可以浏览网站：<http://metalab.unc.edu/LDP>。

Linux 操作系统不是共享软件，它不在公共域中。Linux 操作系统并不仅仅适用于黑客，而且更适用于高端网络服务器操作、多变的替代系统平台，同时也是一个适用于桌面计算机的操作系统。

现在 Linux 已经越来越受到人们的关注，其相关应用也渗透到桌面、服务器、无线设备等各个领域，且发展势头正日趋猛烈。但目前，Linux 面临的最大问题就是应用程序太少，不能满足各个领域使用者的需求。这是 Linux 很难战胜 Windows 的主要原因之一。虽然，随着许多公司和编程爱好者的加入，这种局面正在得到改观，但是由于 IT 业一日千里的发展速度，Linux 迫切需要一个高效的 IDE（集成开发环境）。Kylix 的产生，可以说正是这种要求的发展结果，也是本书要介绍的内容。

1.2 为什么使用 Linux 操作系统

Linux 操作系统经过这些年的发展之后, 已经逐渐走向完善, 可以说现在的 Linux 就是运行于 PC 上的 UNIX 操作系统。它不但继承了 UNIX 的许多功能和优点, 而且其更加短小的体积及几乎免费的低廉价格也是其他操作系统没法比拟的。具体说来, Linux 具有如下优点:

(1) 简单内核结构

Linux 的内核只保留了基本的内核功能, 通过高度安全的信息机制, 与各个模块化了的功能进行协调工作, 不但提高了系统的安全性, 而且使系统内核便于根据需要重新裁减, 具有很大的灵活性。

(2) 多用户

多用户是指在同一时间内可以有多个用户使用同一台机器。在这方面, Linux 的实现很方便, 和其他商业性的操作系统比起来, Linux 没有最多用户人数限制。对于很多用户来说, 这是很有用的, 不需要大量系统资金的投入就可以获得多用户的支持, 这对于构建非盈利性质的站点有着巨大的吸引力。

(3) 多任务

多任务是指同一台计算机在同一时间可以运行多个应用程序。用户可以同时做多件事情, 这不论对用户来说, 还是对最大限度地利用系统资源来说都是非常有好处的。这一点是从 UNIX 那里继承下来的。

(4) 多平台

Linux 最初是为在 x86 平台下运行而设计的, 但由于它在 Internet 上的成功引起了世人的注意, 所以很快被移植到 SPARC 和 Alpha 平台上了, 并且 RedHat 公司已经推出了这两种平台下的发行套件。到目前, 由于 Linux 的名气越来越大, 许多商业公司都开始支持 Linux, Linux 向其他平台的移植工作进行得更加迅速。

(5) 兼容 IEEE POSIX.1 标准

IEEE POSIX.1 标准是 UNIX 可移植编程接口标准, Linux 与之兼容, 使得许多运行于 UNIX 系统上的优秀软件, 可以不做修改或只做微小的修改就可以稳定地运行在 Linux 操作系统之上, 因而提高了 Linux 系统的跨平台性能, 也使得 Linux 有了丰富的软件资源。

(6) 按需存取

在 Linux 下, 一个应用程序执行时, 并不是它所有的代码段都要被加载到内存中。系统只把那些在应用程序运行时确实要用到的代码段加载到内存中, 这样做就减少了读写磁盘的时间, 加快了程序的运行。它相当于 DOS 中的 smartdrv 应用程序, 但是前者效率要比 smartdrv 高很多。

(7) 内存保护

在 Linux 下, 应用程序只能访问系统分配给它的内存区域, 而不能访问其他内存区域。这种相对独立的内存区域有个好处: 当一个应用程序出错时, 不会影响到操作系统的正常运行。基本不会出现而 Windows 3.1 和 Windows 9x 中由于一个应用软件错误导致系统死锁的现象。而且 Linux 还解决了 UNIX 下经常由于溢出而造成的系统安全漏洞问题。