

董 渊 朱亚平 倪 逸 等编著



Linux 系统 Motif/OpenGL 程序开发



机械
China

出版社
China Press

00010250

TP316.81

31



Linux 系统 Motif/OpenGL 程序开发

董渊 朱亚平 倪逸 等编著

32 62



机械工业出版社



C0487109

X-Window/Motif 是计算机工作站和大型机主要的图形用户界面系统,也是自由操作系统 Linux 的用户界面,目前已经成为应用最广泛的图形用户界面标准。采用 X-Window/Motif 技术开发的应用软件,具有很高的可移植性。

OpenGL是三维图形设计与制作的接口标准,广泛应用于可视化技术、实体造型、CAD/CAM、模拟仿真等诸多领域,是一套独立于操作系统和硬件环境的三维图形库,具有强大的图形功能和优秀的跨平台移植能力,已经得到IT工业界的广泛认可。

本书力求以通俗精炼的语言,介绍了在Linux Red Hat 6.0平台上开发Motif和OpenGL程序的技术,可供广大希望从事图形界面、图形设计制作方面软件开发的Linux爱好者参考,也可作为Linux下C语言开发的入门教材,同时对其他Unix系统下的开发人员也有一定借鉴作用。

图书在版编目(CIP)数据

Linux 系统 Motif/OpenGL 程序设计/董渊等编著. —北京:机械工业出版社, 2000.1

ISBN 7-111-07783-0

I. L… II. 董… III. 图形软件, Motif/OpenGL -程序设计
IV. TP391.41

中国版本图书馆 CIP 数据核字(1999)第 74730 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:张秀恩 封面设计:姚毅

责任印制:何全君

三河市宏达印刷厂印刷 • 新华书店北京发行所发行

2000 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16 28 印张 • 672 千字

0001—5000 册

定价:45.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本书购书热线电话(010)68993821、68326677-2527

Http://www.machineinfo.gov.cn/book/

序

Linux 是 Unix 操作系统的兼容产品，它具有 Unix 的所有特性，并且除了这些技术特性外，Linux 还有一个最吸引人的特性，就是 Linux 是一个自由软件，它是由世界各地的许多计算机爱好者们协作开发的，人们可以自由地得到 Linux 的所有源程序代码，这对于想学习和开发的朋友们是一个再好不过的福音了。

Linux 具有很强的实用性和可靠性，现在很多的科研机构、大专院校和商业公司都纷纷把 Linux 作为首选的操作系统平台，并且随着这场席卷全球的 Linux 热潮，很多的公司和科研机构纷纷推出支持 Linux 的产品，有自由软件也有商业软件。总之，随着时间的推移，这方面的东西会更多更好。

Linux 是高效和稳定的，借助于 Linux，我们可以把一台微机当作高档图形工作站来使用，甚至很多的 ISP 站点的服务器也是使用 Linux 加微机搭建的，本书就是在普通的 586 上由作者经过潜心研究学习而写成的。

这本 Motif/OpenGL 的书，不仅仅是 Motif/OpenGL 图形/图像方面的读物，由于作者把 Linux 下的基本编程写进去了，并且是循序渐进的，所以很适合学习使用 Linux 等自由软件的朋友们参考学习，另外，在自由软件界也有 Lesstif 和 Mesa，它们是 Motif 和 OpenGL 兼容的软件包，不同点在于 Lesstif 和 Mesa 是自由软件，可以直接免费下载使用。

看到目前在计算机界的各种报纸上炒作的什么什么等，有能这么仔细写 GNU、GPL 的稿子，循序渐进地介绍 Linux 的编程和调试，并且完整地介绍了如何使用 Motif/OpenGL 的书稿，真是太好了！但愿大家都能坐下来踏踏实实地写些、做些真东西。

freesoft.cei.gov.cn 系统管理员 张鹿

于 1999 年 10 月

前 言

Linux 是在日益普及的 Internet 上迅速形成和不断完善的操作系统，与 Windows 等商业操作系统不同，它完全是一个自由的操作系统。Linux 是一个 Unix 操作系统的克隆，可以免费使用，遵循 GPL (GNU General Public) 声明 (见附录 A)，可以自由修改和传播。最初是由芬兰籍的 Linus Benedict Torvalds 先生和经由 Internet 组织起来的开发小组编写的，其目标是与 POSIX (Portable Operating System Interface, 可移植操作系统界面) 兼容。Linux 包含了所有人们期望操作系统拥有的特性，不仅仅是 Unix 的，而且是任何一个操作系统的功能。它包括真正的多任务、虚拟内存、世界上最快的 TCP/IP 驱动程序、共享库和多用户支持。

图形用户界面 (Graphics User Interface, GUI) 提供了友好的人机交互环境，使得计算机的使用不再是枯燥的命令记忆，而变得更容易操作。现在，基于窗口系统的图形用户界面在整个计算机世界中已经基本普及。基于 X-Window 系统 (也简称 X 系统) 的 OSF (Open Software Foundation, 开放软件基金会) /Motif 就是最优秀的图形用户界面系统之一，这个系统是一个事实上的工业标准，几乎可以在所有的平台 (硬件、操作系统) 上运行。而使用 X-Window 和 Motif 所提供的函数库，可以开发出适用于各种 Unix 系统的软件。

Linux 系统采用的图形用户界面就是基于 X-Window 下的 Motif 系统，这里的 Motif 系统具有在一般 Unix 系统下的全部功能，而且在这样的界面下开发的程序可以不加修改或稍加修改就能在其它 Unix 系统下编译运行，因此，我们学习 Linux 下的 Motif 程序开发，开发的程序实际上几乎可以在任何一个 Unix 平台上运行。

OpenGL即开放的图形库 (Open Graphics Library)，是近几年发展起来的一个性能卓越的三维图形标准。它最初来自于SGI公司为图形工作站开发的IRIS GL。由于使用简单，该软件一经推出即受到广泛关注。目前，包括Microsoft、SGI、IBM、DEC、SUN、HP等在内的许多大公司都采用了OpenGL作为三维图形标准，许多软件厂商也纷纷以OpenGL为基础开发自己的产品，从而使OpenGL成为被IT工业界广泛接受的三维图形软件接口标准。

OpenGL的一个引人注目之处在于把图形绘制命令从具体窗口中独立出来，而把管理窗口的任务留给了扩展的函数库，如glx库专门用于OpenGL与X-Window系统的接口。作为一个开放共享的三维图形软件接口，OpenGL是独立于软硬件平台的，它可以采用FORTRAN、C、C++、Java等多种编程语言，以它为基础开发的应用程序可以十分方便地在各种平台 (如 Windows、Unix及Linux) 间移植。OpenGL的另一个显著特点是它的网络功能。在网络中，即使创建图形程序的计算机不是运行图形程序的计算机，只要它们遵守同样的通信协议，OpenGL就可正常运行。

在Linux系统下，Lesstif 是免费的Motif，它具有Motif的所有功能，同时遵循GPL声明，而Mesa则可作为OpenGL的代用品。本书力图以通俗的语言，介绍在Linux系统下Motif/OpenGL (也可以是Lesstif/Mesa) 的程序设计方法，使读者通过阅读本书就可以尽快开始在Linux这一优秀的操作系统下开发图形用户界面、处理图形。因此，本书从实用角度出发，采用实例剖析的方法来组织内容。

本书主要内容

本书共包含 15 章，按照内容可以划分为 4 个部分：

第 1 部分 Linux 程序开发基础知识，包括第 1、2、3 章。这部分内容是学习 Linux 系统下程序设计的基础。首先简单介绍 Linux 的基本常识和发展过程，尤其是它在中国的发展概况，接着介绍在 Linux 系统下进行程序开发的支撑工具，它们是：C 语言编译器 gcc、程序维护工具 make、程序调试工具 gdb 和 xgdb 以及程序版本维护工具 rcs。最后按照程序开发流程的顺序，学习编辑、编译连接、调试的工具和使用方法。特别重要的是在这个部分中，我们还使用大量篇幅介绍了获取帮助的方法，联机帮助是最基本和最容易取得的帮助类型，通常遇到问题之后首先应该查看这些帮助文档。另外，通过网络，我们有可能得到更多高手的亲自指点，可以获得更新、更具有针对性的资料，必将对程序设计的学习大有帮助。

第 2 部分 Motif 程序开发技术，包括第 4、5、6 章，系统地介绍 Motif 程序设计的知识。学习在 Linux 系统下开发 Motif 应用程序就是使用在 Linux 系统下由自由软件所提供的 Xlib、Xt Intrinsics (XFree86 提供) 和 Motif Toolkit 函数库 (OSF/Motif Release 或者 Lesstif 提供) 来编写用户应用程序，这样的应用程序是具有 Motif 风格的 X-Window 程序。

Motif 程序开发接口主要由三个部分组成，它们是 Xlib、X Toolkit Intrinsics 和 Motif Toolkit (有时也简称为 X 库、Xt 库和 Xm 库)，分别是由 X 系统和 Motif 提供。

X 系统提供的接口主要是 Xlib，它是一个 C 语言函数库，是 X 程序的基础，主要负责客户程序与 X 协议的联系。X Toolkit Intrinsics (Xt 库) 是建立在 X 库基础上的高层软件开发工具箱，由 X 联合会推出。Xt 库一般和 X 系统同时提供，包括 X 程序开发的许多实用函数和数据结构。Motif Toolkit (Xm 库) 是 OSF 提供的程序开发接口，包括组件集合和便利函数，其中组件包括按钮、标题、滚动条等等，它们都遵循 Motif 的风格规定，便利函数则可以代替多个 Xt 函数完成特定的任务。联合使用 X 库、Xt 库和 Xm 库，就可以更简便快速地开发出具有 Motif 风格的 X 程序。

UIL (User Interface Language, 用户界面语言) 是 Motif 四个组成部分之一，其中以 UIL 位于整个 Motif 接口层次结构的最高层，它提供了接近人类自然语言的界面描述语言。UIL 语言用于描述用户界面的初始状态，它提供了一种使用 Motif Tk 的简便方法。使用 UIL 语言，可以直接描述用户界面构成组件的属性和对应的回调函数，以及各组件对象之间的层次关系和相互位置关系。和 UIL 密切相关的是 MRM (Motif Resource Manager, Motif 资源管理器)，MRM 提供一系列专门的函数来管理界面文件，C 语言的 Motif 程序通过调用 MRM 函数来使用 UIL 程序，共同实现一个 Motif 程序。

第 4 章主要是 X-Window 与 Motif 入门，第 5 章介绍 Motif 程序开发基础，第 6 章是 UIL 开发基础。

第 3 部分介绍 OpenGL 程序开发技术，包括第 7~14 章。以 Mesa 为例，通过大量精心编制的程序示例由浅入深地介绍 Linux 系统下 OpenGL 的基本编程原理及编程技巧，给读者以清新的感受。本部分所涉及内容包括：

- 1) 基本几何对象：除了提供点、线、多边形等基本几何图元的绘制函数外，还提供了部分复杂的三维物体 (球、锥、多面体等)，以及复杂曲线和曲面 (如 Bezier 曲线及曲面、NURBS 等) 的绘制函数；

2) 颜色模式: 包括RGBA模式和颜色索引 (Color Index) 模式, 此外还规定了颜色过渡模式 (平直或光滑过渡);

3) 变换操作: 通过变换矩阵的存储状态控制实现取景-模型变换、投影变换、视口变换及视图裁剪等操作, 应当注意得是, 变换操作的顺序对于三维图形绘制来说意义至关重要;

4) 显示列表: 将一组需要反复执行的OpenGL命令存储并编译到显示列表中, 可以有效提高绘图性能;

5) 光照和材质: 以辐射光 (Emitted Light)、环境光 (Ambient Light)、漫反射光 (Diffuse Light) 和镜面光 (Specular Light) 模拟真实光照及材质属性, 场景中物体最终反映到人眼的颜色是光的RGB分量与材质RGB分量的某种组合;

6) 纹理映射 (Texture Mapping): 利用缠绕、滤波等手段实现纹理贴图, 可以十分逼真地模拟物体表面细节;

7) 位图和图像: 位图可用于绘制各种字符集, 而对像素的操作包括读写、拷贝、平移、缩放和存储、传输、映射等操作;

8) 雾化 (Fog)、融合 (Blending) 以及反走样 (Antialiasing) 等效果处理;

9) 双缓存 (Double Buffering): 后台缓存计算场景、生成画面, 前台缓存显示后台缓存已画好的画面;

10) 帧缓冲区: 包括颜色、深度、模板及累加缓冲区, 对这些缓冲区的操作可实现抖动 (Dithering)、深度暗示 (Depth Cue)、运动模糊 (Motion Blur)、镜像及影子等特殊效果, 并实现了消隐算法。

作为独立于硬件平台的三维图形编程接口, OpenGL不但为三维图形操作提供低层支持 (如: 基本几何图元、基本光照控制、坐标变换及帧缓存操作等), 还支持图像与帧缓存之间的传输等操作以及纹理映射等。总之, OpenGL简化了程序员在图形接口方面的诸多工作, 而且可以很方便地实现程序在系统平台之间的移植。

第4部介绍分综合使用Motif与OpenGL, 通过一个实例, 学习Motif与OpenGL的结合使用, 完成具有图形用户界面的图形处理功能。

本书面向读者

本书的读者对象主要是广大对图形用户界面和图形制作处理有兴趣的 Linux 爱好者, 也可以是在 Unix/ Linux 系统下的应用程序开发者, 同时也适用于需要学习 Motif/OpenGL 的学生和研究人员。读者应当具有初步的计算机程序设计基础, 对 C 语言有一定程度的掌握, 而最关键的是应该对程序设计、尤其是对图形用户界面和图形制作处理有一定兴趣, 因为, “兴趣是最好的老师”。

说明

Linux、X-Window、Motif 和 Mesa 都具有不同的版本, 并且都处于飞速发展过程中, 本书中使用的 Red Hat 6.0 Linux、X-Window 基本按照 Version11、Release6 (即 X11R6) 进行介绍, OSF/Motif 按照 Release 2.0 介绍, OpenGL 则使用 Mesa 2.6 版本, 编程语言采用标准 C。

本书给出的完整源程序,有些已经标明出处,其他均为作者自行开发,这些程序在 Red Hat 6.0 Linux 下使用 gcc 编译器编译通过并通过实际运行检测。源程序片段是为了说明语法规则而设计,没有经过正式编译和调试。作者郑重宣布,这些程序均遵循 GPL 声明(见本书附录 A),欢迎读者引用。

本书主要由董渊、朱亚平和倪逸编写,其中董渊完成第 1、2 部分,朱亚平完成第 3 部分,第 4 部分由董渊和倪逸共同完成。

另外,本书编写过程中,杨彬、陈旭、孟鸿利、张涵秋参加了资料查询整理,其他参加人员还有:王翔、宋健兰、李腾、金志强、覃红侠、金丽萍、朱雄志、张涵青、董喜明、徐昕、王军、张宏斌、张友生、孙卫涛、韩鹏、潘擎宇、文铭、宋康祖、杜启振、刘彬、裴民、刘宁晖、顾克勤、刘勇、陈永智、倪任翔、董道豫等人。

在本书编写过程中,除参考书后附录所列出的文献外,还参考了其他 Linux 文档及著作,特别是国内相关网站上的著作,限于篇幅不能一一列出,在此向各位作者表示歉意,并致以衷心感谢。

作者

1999 年 10 月于清华园

目 录

序
前 言

第 1 部分 Linux 程序开发基础知识

第 1 章 认识 Linux.....	1
1.1 Linux 是自由的操作系统	1
1.2 Linux 和自由软件	2
1.3 Linux 的发展简史	4
1.4 Linux 在中国	6
1.5 总结	9
第 2 章 Linux 系统 C 语言开发支撑工具	10
2.1 支撑工具简介.....	10
2.2 C 语言编译器——gcc	11
2.2.1 gcc 简介	11
2.2.2 gcc 的执行过程.....	11
2.2.3 gcc 的基本用法和选项.....	11
2.2.4 gcc 的错误类型及对策.....	13
2.3 程序维护工具——make	14
2.3.1 make 简介	14
2.3.2 基本原理和用法	14
2.3.3 宏(marco)和注释	16
2.4 程序调试工具.....	18
2.4.1 调试工具简介	18
2.4.2 调试使用场合	18
2.4.3 调试的功能	18
2.4.4 调试对编译的要求	19
2.4.5 gdb 的常用命令	19
2.4.6 xxgdb 介绍	23
2.5 程序版本维护	24
2.5.1 RCS 简介	24
2.5.2 程序版本结构	24
2.5.3 RCS 使用	24
2.6 总结	25

第3章 熟悉程序开发环境	26
3.1 程序开发流程	26
3.2 程序的编辑之一——使用 vi	27
3.2.1 vi 简介	27
3.2.2 快速掌握 vi	27
3.2.3 vi 的使用	27
3.2.4 进一步学习 vi	29
3.3 程序的编辑之二——使用 EMACS	29
3.3.1 Emacs 简介	29
3.3.2 快速掌握 Emacs	30
3.3.3 Emacs 使用	30
3.3.4 Emacs 的程序开发功能	33
3.3.5 使用 Emacs 编辑程序实例	34
3.4 程序的编译和调试	35
3.4.1 人工编译连接	35
3.4.2 使用 make 编译连接	35
3.4.3 在 Emacs 中编译程序	36
3.4.4 调试	37
3.5 获取帮助之一——联机帮助获取	39
3.5.1 联机帮助手册	39
3.5.2 系统帮助文档	42
3.5.3 Motif 帮助获取	43
3.6 获取帮助之二——网络帮助获取	44
3.6.1 LDP 文档	44
3.6.2 BBS(中文)	45
3.6.3 USENET 新闻组	45
3.6.4 小结	46
3.7 总结	46

第2部分 Motif 程序开发技术

第4章 X-Window 与 Motif 入门	48
4.1 图形用户界面	48
4.1.1 图形用户界面的特征	48
4.1.2 图形用户界面的结构	49
4.1.3 Linux 的图形用户界面	49
4.2 X-Window 系统	51
4.2.1 X-Window 简介	51
4.2.2 X-Window 发展历史	51

4.2.3	X 系统的特征	51
4.2.4	X 系统基本概念	52
4.3	Motif 介绍	54
4.3.1	Motif 简介	54
4.3.2	Motif 的组成	54
4.3.3	窗口管理	55
4.4	Motif 程序开发基础	56
4.4.1	Motif 程序开发简介	57
4.4.2	Motif 程序开发接口	57
4.4.4	Widgets 组件构成	59
4.4.5	基本原则	61
4.5	UIL 基础	61
4.5.1	UIL 与 MRM	61
4.5.2	UIL 的特征	62
4.6	总结	62
第 5 章	Motif 程序开发基础	64
5.1	开始 Motif 程序开发	64
5.1.1	第一个实例	64
5.1.2	程序分析	69
5.1.3	Motif 程序的结构	76
5.2	Motif 的资源	77
5.2.1	Motif 资源简介	77
5.2.2	Motif 资源的控制方式	77
5.2.3	Motif 资源文件	77
5.2.4	Motif 资源的使用	79
5.3	Motif 的组件	80
5.3.1	主窗口和菜单系统	81
5.3.2	对话框	88
5.3.3	标签和按钮	94
5.3.4	其他组件	98
5.3.5	小结	99
5.4	Motif 的事件处理	99
5.4.1	事件类型与处理方式	99
5.4.2	回调函数	101
5.4.3	动作函数	104
5.4.4	小结	110
5.5	X 系统的字体	110
5.5.1	复合字符串	111
5.5.2	字体	112

5.5.3 字体处理实例	113
5.6 总结	114
5.6.1 Motif 程序的结构	114
5.6.2 资源及管理	115
5.6.3 组件对象	115
5.6.4 事件	116
5.6.5 字体和复合字符串	117
5.6.6 实例	117
第 6 章 UIL 开发基础	118
6.1 UIL 简介	118
6.2 UIL 语言基础	119
6.2.1 字符集	119
6.2.2 名字	119
6.2.3 关键字	120
6.2.4 数据类型	121
6.2.5 表达式	122
6.2.6 取值函数	122
6.3 UIL 模块	123
6.3.1 UIL 模块内容	124
6.3.2 UIL 模块结构	124
6.3.3 UIL 模块书写规则	133
6.4 使用 UIL 的 Motif 程序实例	134
6.4.1 程序的功能	135
6.4.2 程序的组成	135
6.4.3 编译连接过程	138
6.5 使用 UIL 的 C 程序结构	140
6.5.1 初始化 Mrm	140
6.5.2 建立组件对象	140
6.6 使用 UIL 进行资源管理	144
6.6.1 颜色与字体管理	144
6.6.2 像图管理	147
6.7 总结	152

第 3 部分 OpenGL 程序开发技术

第 7 章 OpenGL 概述	154
7.1 OpenGL 的发展历程	154
7.2 OpenGL 特点及功能	155
7.2.1 OpenGL 基本功能	155

7.2.2 OpenGL1.2 的部分特性	156
7.3 Mesa 简介	157
7.4 OpenGL 函数库	158
7.4.1 gl 核心函数库	158
7.4.2 glu 实用库	160
7.4.3 glaux 辅助库	160
7.4.4 gltk 工具库及与 glaux 库的比较	161
7.4.5 其他扩展函数库	165
7.5 OpenGL 的语法规则及状态机制	165
7.5.1 语法规则	165
7.5.2 状态机制	166
7.6 OpenGL 的相关信息	167
7.6.1 OpenGL 技术评审委员会	167
7.6.2 Linux 下 OpenGL 的网址	167
7.6.3 使用或支持 OpenGL 的一些相关产品	167
7.7 总结	172
第 8 章 Mesa/OpenGL 编程要点	173
8.1 一个简单 Mesa 程序示例	173
8.1.1 迷宫程序代码	173
8.1.2 程序分析	178
8.1.3 Linux 及 Windows 系统下 Mesa/OpenGL 程序框架的比较	179
8.2 图形绘制的辅助技术	180
8.2.1 清除缓冲区	180
8.2.2 消隐	181
8.2.3 强制绘图完成	182
8.2.4 交换缓冲区	183
8.3 总结	184
第 9 章 绘制几何实体	185
9.1 基本几何图元	185
9.1.1 绘制开始与结束命令	185
9.1.2 点	186
9.1.3 线	193
9.1.4 多边形	200
9.1.5 法向量	211
9.2 颜色	212
9.2.1 颜色模式	212
9.2.2 颜色过渡模式	218
9.3 变换	225
9.3.1 模型—取景变换	226

9.3.2	投影变换	228
9.3.3	视口变换	230
9.3.4	附加裁剪平面	231
9.3.5	矩阵堆栈	232
9.3.6	程序示例	233
9.4	总结	245
第 10 章	显示列表	246
10.1	立即执行模式与显示列表	246
10.1.1	立即执行模式	246
10.1.2	显示列表方式	247
10.2	显示列表的创建与执行	247
10.2.1	创建显示列表	247
10.2.2	执行显示列表	248
10.2.3	显示列表嵌套	249
10.3	程序示例	249
10.4	总结	259
第 11 章	光照、材质与纹理	260
11.1	光照处理	260
11.1.1	光照基本概念	260
11.1.2	光源属性	261
11.1.3	光照模式	265
11.1.4	程序示例	267
11.2	材质	275
11.2.1	材质属性	275
11.2.2	材质程序示例一	277
11.2.3	材质程序示例二	285
11.3	纹理映射	296
11.3.1	纹理基本概念	296
11.3.2	纹理的定义	296
11.3.3	纹理坐标	301
11.3.4	程序示例	302
11.3.5	纹理缠绕方式	311
11.3.6	本章程序综合示例	312
11.4	总结	319
第 12 章	雾化、融合与反走样	320
12.1	雾化	320
12.1.1	雾化效果设置	320
12.1.2	程序示例	321
12.2	融合	329

12.2.1	融合效果设置	329
12.2.2	程序示例一	330
12.2.3	程序示例二	336
12.3	反走样	345
12.3.1	反走样效果设置	345
12.3.2	程序示例	345
12.4	总结	352
第 13 章	位图与图像	353
13.1	位图	353
13.1.1	位图绘制函数	353
13.1.2	位图程序示例	353
13.1.3	字体程序示例	361
13.2	图像	369
13.2.1	像素的读写、拷贝、缩放操作	370
13.2.2	像素的存贮、传输、映射操作	371
13.2.3	程序示例	374
13.3	总结	382
第 14 章	求值器与 NURBS	384
14.1	求值器	384
14.1.1	求值器的计算原理	384
14.1.2	求值器函数	384
14.1.3	程序示例	386
14.2	NURBS	395
14.2.1	NURBS 编程思路	395
14.2.2	NURBS 函数	395
14.2.3	程序示例	396
14.3	总结	402

第 4 部分 综合使用 Motif 与 OpenGL

第 15 章	Motif 与 OpenGL 联合编程	403
15.1	Motif 与 OpenGL 联合编程	403
15.2	实例介绍	404
15.2.1	程序功能与运行结果	404
15.2.2	程序源代码	405
15.2.3	编译连接	411
15.3	实例程序剖析	412
15.3.1	组件对象层次结构	412
15.3.2	便利函数与回调函数	413
15.3.3	OpenGL 处理	413

15.4 总结	413
附录 A GPL 声明	414
附录 B Xt 和 Motif 组件类族	427
附录 C X 系统设计 7 原则	428
附录 D 通用包含头文件	429
参 考 文 献	431

第 1 部分 Linux 程序开发基础知识

第 1 章 认识 Linux

Linux 是在日益普及的 Internet 上迅速形成和不断完善的操作系统，它的开发者和维护者一般都是职业编程专家、业余计算机好手和黑客，他们通过互联网传递信息、互相交流，仅仅几年时间，就取得了足以动摇微软 Windows 操作系统的伟大成果。在国内，Linux 的爱好者主要是一批有计算机专长的年轻人：高校在校学生、科研机构研究人员和 ISP(Internet Service Provider)技术人员，他们在各地组织俱乐部，通过各种形式的活动，宣传、普及、传播和发展利用 Linux 这个现代科技的结晶，形成了我国计算机行业发展的一个新热点。

Linux 到底是什么？它有那些优势？为什么引得众多的爱好者、厂家、商家如此关注？它对软件行业，特别是我国的软件行业，可能会带来什么样的影响？这正是我们这一章企图向大家阐明的主要内容。

1.1 Linux 是自由的操作系统

Linux 是一个 Unix 操作系统的克隆，可以免费使用，遵循 GPL(GNU General Public)声明(见附录 A)，可以自由修改和传播。最初是由芬兰籍的 Linus Torvalds 先生和经由 Internet 组织起来的开发小组编写的，其目标是与 POSIX(Portable Operating System Interface，可移植操作系统界面)兼容。Linux 包含了所有人们期望操作系统拥有的特性，不仅仅是 Unix 的，而且是任何一个操作系统的功能。包括真正的多任务、虚拟内存、世界上最快的 TCP/IP 驱动程序、共享库和理所当然的多用户支持（这意味着成百上千的人能在同一时刻，或者通过网络、Internet，或者通过连接在计算机串行口上的终端或膝上机/微机，使用同一台计算机）。因此，我们可以说，与 Windows 等商业操作系统不同，Linux 完全是一个自由的操作系统。

操作系统一般由内核和一些系统程序组成，同时，还有一些应用程序帮助用户完成特定任务。内核是操作系统的灵魂，它负责管理磁盘上的文件、内存，负责启动并运行程序，负责从网络上接收和发送数据包等等。总而言之，操作系统实际是抽象的资源操作到具体硬件操作细节之间的接口。

从程序员的角度来讲，操作系统提供了一个与计算机硬件等价的扩展或虚拟的计算平台。它抽象了许多硬件细节，程序可以以某种统一的方式进行数据处理，而程序员则可以避开许多硬件细节。从另一个角度讲，普通用户则把操作系统看成是一个资源管理者，在它的帮助下，用户可以以某种易于理解的方式组织自己的数据，完成自己的工作并和其他人共享资源。

Linux 有两种不同的含义。从严格的技术定义讲，Linux 指的是开放源代码的 Unix 类操作系统的内核。然而，目前大多数人用它来表示以 Linux 内核为基础的整个操作系统。从这