

海洋出版社

KEITH D. GREGORY

PROGRAMMING WITH
MOTIF™

编程指导与实例

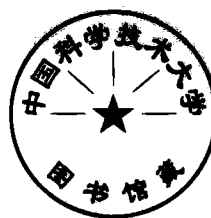


希望

SPRINGER-VERLAG

Programming With Motif
MOTIF 编程指导与实例

Keith D. Gregory 著
郭勇 郭莉等 译
沈翔 审校



海洋出版社

内 容 摘 要

本书详细介绍了如何利用 OSF/Motif™ 编制图形用户界面。它教你如何在复杂的程序中使用 Motif 所提供的各种界面元素。它并不需要读者具有在 X Window 环境下编程的经验。本书还提供了一些具有实用价值的 Motif 编程实例,通过这些实例读者可以立刻着手 Motif 编程。

需要本书的用户,请直接与北京海淀 8721 信箱书刊部联系,邮码:100080,电话:2562329。

版 权 声 明

本书英文版名为《Programming With Motif™》,由 Springer-Verlag 公司出版,版权归 Springer-Verlag 公司所有。本书中文版由 Springer-Verlag 授权出版。未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

JS198/25

(京)新登字第 087 号

MOTIF 编程指导与实例

Keith D. Gregory 著

郭勇 郭莉等译 沈翔 审校

*

海洋出版社出版发行(北京复兴门外大街 1 号)

双青印刷厂印刷

开本:787×1092 1/16 印张:33.5 字数:790 千字

1994 年 9 月第一版 1994 年 9 月第 1 次印刷

印数:1~2000

*

ISBN 7-5027-0676-3/TP·255 定价:56.00 元

目 录

导 读.....	(1)
第一章 关于 Motif 的介绍.....	(6)
1.1 “是机制而不是政策”	(6)
1.2 进入 Motif	(6)
1.3 X 层次结构	(7)
第二章 widget	(9)
2.1 什么是 widget	(9)
2.2 面向对象编程的介绍	(9)
2.3 作为对象的 widget	(10)
2.4 Motif 类树	(10)
2.5 widget 事例树.....	(12)
第三章 widget 资源	(14)
3.1 什么是资源.....	(14)
3.2 资源表.....	(14)
3.3 关于命名约定的注意事项.....	(15)
3.4 资源文件.....	(16)
3.5 widget 的命名.....	(16)
3.6 资源说明格式.....	(19)
3.7 资源文件样本.....	(19)
3.8 通过类设置资源.....	(21)
3.9 widget 名使用通配符后的解释.....	(21)
3.10 不加设置的资源	(22)
3.11 为资源文件加注释	(22)
3.12 程序如何使用资源	(22)
3.13 在命令行上设置资源	(24)
3.14 用编程方法进行资源设置	(24)
3.15 固联资源值	(26)
3.16 用程序获取资源值	(26)

第四章 编写一个 Motif 程序	(28)
4.1 widget 的数据类型	(28)
4.2 一个 Motif 程序的生命周期	(28)
4.3 程序模板	(32)
4.4 头文件	(33)
4.5 Motif 程序的编译	(34)
4.6 关于程序量大小的一点说明	(35)
4.7 什么是标准 I/O	(36)
第五章 XmLabel	(37)
5.1 概述	(37)
5.2 XmLabel 的继承性	(37)
5.3 XmLabel 资源	(37)
5.4 使用 XmLabel: “Hello, World!”	(40)
5.5 设置 Label 中的字体	(41)
5.6 XmString 数据类型	(44)
5.7 通过程序设置 LabelString	(47)
5.8 使用 XmLabel 显示图像	(49)
第六章 管理器	(51)
6.1 概述	(51)
6.2 继承	(51)
6.3 Constraint(约束): 概述	(52)
6.4 再谈管理器	(52)
6.5 度量: unitType 资源	(55)
6.6 XmBulletinBoard	(58)
6.7 XmBulletinBoard 实例	(61)
6.8 XmRowColumn	(65)
6.9 XmRowColumn 实例	(71)
6.10 XmPanedWindow	(77)
6.11 XmPanedWindow 实例: 三个标号和一个方格窗口	(80)
6.12 XmForm	(81)
6.13 XmForm 实例	(88)
第七章 按钮	(95)
7.1 概述	(95)
7.2 继承	(95)
7.3 XmPushButton	(96)
7.4 XmPushButton 实例: 按这儿	(99)

7.5	XmArrowButton	(101)
7.6	XmArrowButton 实例	(102)
7.7	XmToggleButton	(104)
7.8	XmToggleButton 实例	(109)
第八章	事件与回调	(112)
8.1	概述	(112)
8.2	处理 X 事件	(113)
8.3	事件句柄实例:鼠标跟踪器	(121)
8.4	回调	(127)
8.5	按钮回调	(129)
8.6	XmPushButton 回调实例:Press Here	(131)
8.7	动作	(132)
8.8	转译	(135)
8.9	动作/转译实例:多按钮下压按钮	(142)
8.10	XmPushButton 默认转译	(144)
第九章	键盘输入	(146)
9.1	概述	(146)
9.2	输入焦点	(146)
9.3	键盘事件	(147)
9.4	遍历	(147)
9.5	遍历实例:三个 Tab 组	(149)
9.6	XmText	(152)
9.7	XmText 资源	(153)
9.8	XmText 回调	(158)
9.9	XmText 默认转译	(160)
9.10	XmText 便利函数	(164)
9.11	XmText 实例:便笺	(167)
9.12	XmText 实例:入口域	(169)
9.13	滚动正文	(173)
9.14	创建滚动正文“widget”	(174)
9.15	滚动正文资源	(174)
9.16	滚动正文实例:文件浏览器	(176)
第十章	滚动条	(180)
10.1	概论	(180)
10.2	滚动条成份及术语	(180)
10.3	XmScrolledWindow	(181)

10.4	XmScrolledWindow 资源	(182)
10.5	创建和初始化滚动窗口	(185)
10.6	XmScrolledWindow 实例:滚动的行—列	(186)
10.7	XmScrollBar	(189)
10.8	XmScrollBar 的资源	(189)
10.9	XmScrollBar 回调	(192)
10.10	由程序来得到或设置滚动条的值	(193)
10.11	XmScrollBar 示例	(194)
10.12	XmScale	(197)
10.13	XmScale 资源	(197)
10.14	XmScale 的回调	(200)
10.15	由程序得到如设置标尺的值	(200)
10.16	XmScale 示例	(201)
第十一章	Motif 程序的外观	(204)
11.1	概述	(204)
11.2	Motif 客户的成分	(204)
11.3	XmMainWindow	(205)
11.4	XmMainWindow 资源	(205)
11.5	XmMainWindow 的便利函数	(207)
11.6	XmSeparator	(208)
11.7	XmSeparator 资源	(208)
11.8	XmSeparator 示例	(211)
11.9	XmFrame	(213)
11.10	XmFrame 资源	(213)
11.11	XmFrame 示例:在框架中显示 label	(215)
11.12	新的标准程序模板	(217)
11.13	一简单应用程序:文本编辑器	(221)
第十二章	菜 单	(227)
12.1	概述	(227)
12.2	XmCascadeButton	(227)
12.3	XmCascadeButton 资源	(227)
12.4	XmCascadeButton 的回调	(229)
12.5	XmRowColumn 菜单资源	(229)
12.6	XmRowColumn 回调	(231)
12.7	菜单条	(232)
12.8	标准 Motif 菜单条	(234)
12.9	菜单条示例程序:给编辑器增加一菜单条	(236)

12.10	XmMenuShell 与一菜单条的事例树	(239)
12.11	下拉菜单	(240)
12.12	下拉菜单示例:为编辑器添加下拉菜单。	(241)
12.13	助记符	(245)
12.14	菜单加速键	(247)
12.15	级联下拉菜单	(249)
12.16	级联下拉菜单示例	(250)
12.17	弹出菜单	(251)
12.18	弹出菜单示例:色彩选择菜单	(251)
12.19	选项菜单	(255)
12.20	针对选项菜单而作的资源 xmRowColumn 设置	(255)
12.21	选项菜单的例子程序:字体族与大小	(257)
12.22	总结	(260)
第十三章	对 话	(263)
13.1	概述	(263)
13.2	对话的模态性	(264)
13.3	对话设计与成员	(264)
13.4	XmDialogShell	(264)
13.5	XmDialogShell 资源	(265)
13.6	对话的管理与放弃	(267)
13.7	XmBulletinBoard 对话框资源	(267)
13.8	建立并使用一对话	(270)
13.9	通告栏对话的示例程序	(271)
13.10	消息框	(280)
13.11	XmMessageBox	(281)
13.12	XmMessageBox 资源	(281)
13.13	XmMessage 便利函数	(284)
13.14	XmMessageBox 示例:“Can't Find”警告	(285)
13.15	获取与分配输入焦点	(287)
第十四章	列 表	(289)
14.1	概述	(289)
14.2	XmList	(289)
14.3	XmList 资源	(291)
14.4	XmList 回调	(293)
14.5	XmList 便利函数	(295)
14.6	XmList 示例:E-Mail 地址选择	(297)
14.7	滚动列表	(302)

14.8	滚动列表资源	(302)
14.9	滚动列表示例:滚动的 E-Mail 地址	(303)
14.10	XmSelectionBox	(304)
14.11	XmSelectionBox 资源	(305)
14.12	XmSelectionBox 回调	(309)
14.13	创建与使用—选择框	(309)
14.14	访问选择框的孩子	(310)
14.15	XmSelectionBox 示例:E-Mail 地址	(311)
14.16	XmFileSelectionBox	(313)
14.17	XmSelectionBox 资源	(314)
14.18	创建与使用 XmFileSeletionBox	(315)
14.19	XmFileSelectionBox 示例:Open/Save As... 选项	(317)
14.20	XmCommand	(326)
14.21	XmCommand 资源	(327)
14.22	XmCommand 便利函数	(329)
14.23	XmCommand 示例	(330)
第十五章	通过 Motif 使用 Xlib	(332)
15.1	概述	(332)
15.2	X 服务器	(332)
15.3	用数字表示的颜色	(333)
15.4	标准 Xlib 参数	(335)
15.5	图形现场	(336)
15.6	XmDrawingArea	(343)
15.7	XmDrawingArea 资源	(343)
15.8	XmDrawingArea 回调	(344)
15.9	XmDrawingArea 举例:Doodle	(346)
第十六章	资源的进一步介绍	(364)
16.1	概述	(364)
16.2	程序事例名	(364)
16.3	命令行选项	(368)
16.4	资源转换程序	(371)
16.5	编写一个转换程序	(379)
16.6	quark	(386)
16.7	在资源管理器外使用转换程序	(389)
16.8	转换 Cache	(392)
16.9	应用程序资源	(393)

第十七章 客户间通信	(398)
17.1 概述	(398)
17.2 文件 I/O	(398)
17.3 时限	(404)
17.4 后台处理:Workproc	(408)
17.5 在客户间发送事件	(411)
17.6 原子	(413)
17.7 窗口特性	(415)
17.8 窗口管理器协议	(420)
17.9 选择	(427)
17.10 裁剪板	(437)
17.11 示例:把剪切、拷贝、和粘帖功能加入正文编辑器	(443)
17.12 XmText 的直接裁剪板接口	(450)
 附录 A ISO Latin 1 字符集	(452)
 附录 B Widget 类总汇	(453)
 附录 C X 事件参考	(477)
C.1 概述	(477)
C.2 事件类型	(477)
C.3 事件屏蔽码	(478)
C.4 XEvent	(480)
C.5 XAnyEvent	(483)
C.6 ButtonPress, ButtonRelease	(484)
C.7 CirculateNotify	(485)
C.8 CirculateRequest	(486)
C.9 ClientMessage	(487)
C.10 ColormapNotify	(488)
C.11 ConfigureNotify	(489)
C.12 ConfigureRequest	(491)
C.13 CreateNotify	(493)
C.14 DestroyNotify	(494)
C.15 EnterNotify, LeaveNotify	(495)
C.16 Expose	(497)
C.17 FocusIn, FocusOut	(498)
C.18 GraphicsExpose, NoExpose	(499)
C.19 GravityNotify	(501)
C.20 KeymapNotify	(502)

C. 21	KeyPress,KeyRelease	(503)
C. 22	MapNotify,UnmapNotify	(505)
C. 23	MappingNotify	(506)
C. 24	MapRequest	(507)
C. 25	MotionNotify	(508)
C. 26	PropertyNotify	(510)
C. 27	ReparentNotify	(511)
C. 28	ResizeRequest	(512)
C. 29	SelectionClear	(513)
C. 30	SelectionNotify	(514)
C. 31	SelectionRequest	(515)
C. 32	VisibilityNotify	(516)
附录 D	X 和 Motif 的变动	(518)
D. 1	概述	(518)
D. 2	从 Motif 1.0 到 1.1	(518)
D. 3	从 X11 Release 3 到 Release 4	(520)
D. 4	从 X11 Release 4 到 Release 5	(524)

导 读

关于本书

本书详细介绍了如何利用 OSF/Motif™ 编制图形用户界面。说它是一本介绍性质的书是由于它并不需要读者具有在 X Window 环境下编程的经验；另一方面，说它内容详尽是由于它教你如何在复杂的程序中使用 Motif 所提供的各种界面元素。虽然本书包含了大量内容，但它仍旧不是一本权威性的参考资料。OSF/Motif Programmer's Reference 才是一本真正权威性的书，它总共包含了 900 多页的内容，但只有它用户很难掌握 Motif 编程。而本书主要提供了一些具有实用价值的参考内容，它教你如何着手工作，并且使你能更有效地使用 Programmer's Reference 一书。

本书针对的读者是那些在 unix 操作系统下有经验的 C 语言程序员和 X Window 系统的用户。读者应当熟悉 unix 所提供的编译与调试工具。本书并不涉及如何编译 Motif 程序，它假设读者已经熟悉诸如“指点设备”、“显示”这类“X”术语。

本书是如何产生的

作为一名学习 Motif 的程序人员，我发现许多关于 Motif 的书都假设读者已经了解 X 工具箱（这些内容对了解 Motif 的功能以及编写 Widget 是必须的）。而有关 X 工具箱的书对 Motif 又仅仅一带而过，不太考虑那些编写 Widget 的读者。我以前用 Xlib 编写了一些小型软件，现在希望转到 Motif 上来，我并不知道 Widget 是什么东西，就更别谈编一个 Widget 了。

我写这本书是为了那些与我陷入同样困境的人，我认为大多数学习 Motif 的程序员都会遇到上述难题。过去一些时候，一个编程者通过 Xlib 或 Xt 进入 X 环境，其目的是为了编写一个新 Widget 集，而现在，我认为大部分使用 X 环境的程序员是为了寻找解决商业事务问题的方案，一开始便要接触 Motif 或 Open Look。

基于这种情况，本书开始时谈论利用工具箱编程的基本问题，最后讨论 Widget 的实现。在整本书中我都试图侧重于那些对于实际应用有价值的内容，而不去叙述那些如何充分利用环境的细节。

编写过程中，我参考了 X 11 所发布的文档与源代码以及开放软件基金会 (OSF) 出版的 Motif 文档。此外，我还从 comp. windows. x 和 comp. windows. x. motif newsgroup 的编程人员那里得到了大量素材。

推荐的览读计划

刚看到这本书时，你或许会被它包含如此之多的内容吓一跳。如果我要你从头一直读到

尾,你也许更没有读下去的勇气。Motif 是一个十分复杂的环境,我所写的后一章总是基于前一章的内容的。下面我会向你介绍这本书的结构。

然而,如果你只是查找你所关心的内容的话,这里有一个简略的介绍材料,你仅需一个来小时便可看完它。

章节介绍

第一章描述了 Motif:如什么是 Motif? 为什么它会存在? 它给用户与编程人员提供了什么便利? 如果你还不熟悉 X 和 Motif(XLib、工具箱和 Motif 库)的目的与组织方式,我建议你读一下这一章。否则,你可跳过去。

第二章描述 Widget。它是一个抽象的图形对象,是 Motif 编程的核心。我们从两方面对其进行描述,一是从面向对象编程的抽象观点,另一方面是从“Widget 到底做些什么”的编程观点。这一章中阐述了最为核心的两个概念:派生关系(在已有 Widget 的基础上建立一个新的 Widget)与(父子关系——程序中 Widget 的层次结构)。这一章是本书其它部分的基础,每个人都需仔细览读。

第三章描述资源。使用它可以定制一个 Widget 的外观与交互方式。这一章内容颇多,叙述了资源的各种特性。这些都是至关重要的,你应该仔细览读。但是最好能联系一个实际的程序来看,可是目前你还没见到过任何一个程序,因此我建议你首先阅读介绍性章节(在“资源文件示例”中),然后在编写第一个程序时再回头看这一章。

第四章描述一个 Motif 程序的格式并告诉你当按下 Return 键时将会发生什么情况。这些内容也十分重要,需要仔细览读。但你可看完第五章后再看它。然而,我建议你至少先扫一眼这一章,仔细看看“编译一个 Motif 程序”这一节。

第五章介绍第一个 Widget: XmLabel。它用来显示一段静态文本。它还引入了第一个程序,如同大部分书一样,显示一行“Hello, World”。我建议你此时看看以前那些被略掉的内容。

第六章描述管理器 Widget。它被用来定位其它 Widget,因此,它也就控制了一个程序的界面。在这里叙述了四个最常用的 Motif 管理器: XmBulletinBoard, XmRowColumn, XmPaned Window, 以及 XmForm, 并给出了如何使用它们的实例。

第七章描述了标准的 Motif 按钮 Widget: XmPushButton, XmArrowButton 以及 XmToggleButton。在本章中它们被看作用户界面对象,如何链接到程序中将在第八章讨论。

第八章描述 X 事件机制。通过它程序可获得输入。开始我们讨论最低层的实际事件,叙述什么是事件以及程序如何使用它们。然后,介绍回调机制,这是程序与事件间的高层联系。最后讨论动作与转译,这是事件过程中级别更高的内容。

第九章描述键盘输入的方法。一开始介绍键盘事件,然后引出 XmText Widget,它是 Motif 中键盘输入最基本的方法。本章包含了三个关于 XmText 使用的较为复杂的实例:作为一个入口域、作为一个单屏记板以及作为 more 工具程序的替代方法。

第十章描述滚动条。它是一个用来控制程序中数据显示的位置与显示范围的用户界面对象。本章先讨论 XmScrollBar Widget 和它的界面,然后介绍 XmScrolledWindow。它是一个管理 Widget,它能透明地操纵所包含内容的滚动。最后,我们描述一个与此相关的 Widget:

XmScale Widget, 可以用它输入以及显示数据值。

第十一章完整地描述了一个程序。从两方面进行描述:其一是从程序的外观,其二是从那些帮助程序员实现有特定外观的程序的 Widget。本章包含了一个用 Motif Widget 实现的编辑器的程序的开始部分。

第十二章描述菜单。它是一个 Motif 程序控制操作的最基本的方法。开始我们讨论菜单条、它的下拉菜单以及在文本编辑器中如何添加菜单项。本章还描述了能提供现场敏感输入的弹出与选项菜单。

第十三章描述对话框。它为程序提供了用于用户交互的辅助窗口。例子中包含了一个为文本编辑器提供查询功能的用户对话框以及 Motif“封闭”消息对话的使用。

第十四章描述了列表以及用户与它们交互的方法。一开始给出由 XmList 实现的基本列表,紧随其后的是围绕列表所建立的 Widget: XmSelectionBox, XmFileselctionBox 和 XmCommand。本章还包括一些说明列表用法的例子,有 e-mail 地址选择对话和文本编辑器中的文件选择模块。

第十五章介绍通过 Motif 使用 Xlib。Xlib 是低层 X 绘制函数集。本章描述了 Xlib 调用产生的格式以及 XmDrawingArea Widget,它的作用形象地讲就如同为那些调用提供了一块“画布”。这一章内容很有限,讲述 XLib 本身就可以编写一本书。

第十六章包含了一些高级的资源主题。从联接资源说明和程序调用名的方法到建立并使用用于将一种资源数据格式转换成另一种格式的函数。本章描述了 X 实现命令行选项的途径以及使用资源文件修改程序变量的方法。

第十七章即最后一章描述了客户间通信。这个主题既包含了 X 指定的客户间通信方法也包含了传统的通信方式,例如使用管道。这一章内容相当长,介绍了多种通信方法,这在其它一些书中是没有的。

附录是为了读者参考而设立的。附录 A 仅是一张字符表,它包含了 ISO Latin I 字符集,这是 X 和许多现代系统所使用的扩展 ASCII 集。附录 B 是一个关于 Widget 信息的摘要——一个“编程人员微型参考手册”。附录 C 提供了 X 事件的详细描述,这是有关 X 的书必不可少的。附录 D 是关于 X 与 Motif 不同之处的摘要。

源代码约定

如同大部分关于编程的书一样,这本书中有许多源代码。我认为理解一个程序关键在于理解它的命名约定。我在这本书中基本上遵从 MIT X 约定,如下所示:

- 函数名包含连在一起的一个或多个词,每个词的首字母需大写(MyFunction)。
- 变量名包含一个或多个词,每个词均小写。如果变量名包含了多个词,则用下划线将它们分开,如果名字是由几个词的缩略格式构成的,则不需下划线(如 app-shell 代表 'application shell')。

在为 Widget 的变量命名时,本书的约定与 X 约定不一致。本书不区分这些变量与其它程序变量。而 X 标准指定 Widget 名包含一个或多个连写的词且每个词的首字母为大写。

- 宏名与常量包含一个或多个词,并用下划线连起来,同时词中字母大字(如 A-CONSTANT)。这与 X 约定是一致的。X 的“动作”宏可以使用大小写混合的名字。如 XtSetArg。
- 以“X”开头的名字是保留给 X 库的,不能用作变量与函数名。名字前缀指定如下:“X”指低层名(与 Xlib 相关联),“Xt”指派给 X 工具箱(也称为 Intrinsics),“Xm”指派给 Motif。
- 本书中,并不区分全局与局部变量,所有全局变量在一处声明后在程序中保持一致。我倾向于只用全局变量来放置应用数据(对于 Motif 程序,它也用来代表一些普遍使用的 Widget)。我认为无命名约定的必要。

本书中源代码结构规则如下:

- 除了函数体外,同级大括号排列一致(函数体均排在第一列)。大括号独占一行。我不采用“K & R”风格,在那里,大括号放在它所包括的程序块的最后一列。
- 函数参数声明采用老式风格。参数名在头一行指定,而参数声明处于头一行与函体间。之所以这样,主要由于 ANSI C 编译器很晚才引入 unix 的原因。
- 当一个新函数引入时,它由上下文进行原型定义。只有在这种情况下,本书把参数与其名联在一起,本书认为这样做比标准的内缩方式更清晰。

文本约定

本书中使用两种引用符:文本引用符与字符引用符。文本引用符用来强调一些习惯术语以及真正的引用部分;单引号(‘和’)用于计算机界的一些行话,其他引用使用双引号(“和”)。文中双引号用于界定字符串(如“mystring”),单引号(‘)用在单字符字符串中(如‘\n’),这个串表示 C 语言中的 ESC 符。

最后请注意本书中的注文。我用注文为一些主题提供详细的解释。我认为大量使用注文是必要的,因为它可以使你透彻地理解一些抽象的概念。但如果你觉得读注文使你分心的话,你可略过不看或是以后再回头看。

还要说明的一点是,本书中的“父”用于事例树中的 widget:某个 widget 的“父”是指事例树的较高层中紧接该 widget 的 widget(更接近于根);本书中的“孩子”也用于事例树中的 widget:某个 widget 的“孩子”是指事例树的较低层中紧接该 widget 的 widget(离根更远)。

声明

首先我声明这本书是针对 Motif 1.0 与 X11 Release 3 (X11R3)而写的,它们都是相当老的版本。大量调查表明许多人都在使用它们,因此我围绕它们写了这本书。

在写书过程中,Motif 1.1 也正式公布了,但在这本书发行前,Motif1.1 还不十分普遍。Motif1.1 相对于 Motif1.0 版本提供了许多更好的技术,我在正文和注文中都有所涉及。这对于介绍 Motif1.1 的大多数新技术是可行的方法,但对于裁剪板这一技术是不适合的,原

因在于它在 1.0 版中非常不完善,只能针对 1.1 版来进行讨论。

在我写这本书时,X11 Release 4 是随处可见的,在本书最后一稿前,X11Release 5 正式发布。然而,据我调查,许多人仍旧在使用 R3 版本。再者,针对本书所谈内容,R3 与 R4 版本并无多大区别,最大的不同在于类型转换,这在第十七章中谈到了。

另一点声明是虽然 X 与 Motif 在其他一些平台上也可得到,但本书是针对 unix 平台环境而编写的。这是由于我本人熟悉 unix 系统,而无法检测在其他平台下本书是否也可使用。还有一点就是本书篇幅有限,如果再讨论 VMS 和其他系统的内容,篇幅就会过于庞大。我认为本书所包含的一些内容对大部分使用者都是有用的。为了适应于您的具体情况,您可对其中的例子作相应的改动。

第三点,我要声明每个供应商提供的系统都存在着细微差别。文件所处位置甚至文件名与你的实际情况也许不同。在这种情况下,我力图使用标准文件名。在某些情况下,我给出多种名称以供选择。

第一章 关于 Motif 的介绍

1.1 “是机制而不是政策”

这个短语表明了 X Window 系统的设计目标。它的意思是指, X 为图形用户界面提供了一种实现机制, 而并不关心使用这些界面的程序的具体内容。“是机制而不是政策”, 这句话同样适用于 PostScript: 它提供了一种格式化打印页面的机制, 而不指定页的具体内容。

通过避免政策性问题, X Consortium 能够简化 X 的设计, 避免了“由委员会设计”的系统。这种规范有两个重要的结果。第一个是, 一个 X 程序是可移植的, 即一个运行于某一厂商生产的系统中的 X 程序也能够另外的系统下显示输出或接受输入。第二个是, 各厂商能凭自己的“观点和考虑”进行设计, X 并不埋没创造力。

这种做法的缺点是, 它会导致混乱。在为 IBM PC 编写的软件中就能看到这种情况。直到最近, 每个软件还都有其独特的界面, 这样一个使用五种软件的典型用户就至少要记住五种退出程序的方法。

而使用 X 便不会出现这种情况。原因在于, 首先在 X 出现前图形用户界面的概念已很完善; 再者, 编写一个 X 程序不再是件繁琐的任务。每个程序与服务器在低层进行交互, 必要时还要使用为 X 协议所编写的汇编语句。

因此, 以前大部分的软件产品都是使用 X Consortium 编写而不使用早期版本的 X 编写。直到能使编写 X 程序变得简单的 X Intrinsics(Xt) 出现时才改变了这种情况。这是由于 Xt 引入了 widget: 一种提供用户界面功能的独立的对象。

与此同时, X Consortium 发布了 Athena widget 集。虽然 Athena 只被看作一些简单代码, 但它提供了用户界面的大部分元素: 菜单, 文本框等等。从而简化了 X 程序的编写。

1.2 进入 Motif

随着 intrinsics 的出现, 厂商们可以编写他们自己的 widget 集。但不幸的是这些 widget 集都是相互独立的。没有一个供应商愿意为其它供应商的产品提供 widget 集。这种情况依旧可能导致混乱状况, 但实际上却并没有。

原因在于八十年代末出现了两个工业标准制定组织: unix International(UI) 和 the Opensoftware Foundation(OSF)。它们的任务都是制定标准化的操作系统。^① 由于它们双方的合作, 每个产品都有着基于 X 的界面: UI 的界面是 Open Look 而 OSF 的是 Motif。

图 1.1 表示了一个简单的 Motif 程序, 最为引人注目的特征是它的三维外观, 这是由 HP Widget 集提供的。另一与 MS—windows 和 OS/2 Presentation Manager 一致的特征是使用醒目字符作为菜单选项, 并且在菜单条右侧有一 Help 菜单项。此外, 一个 Motif 程序还具

^① UI 的产品是 Unix system V; 而 OSF 的产品是 OSF/1。