



北京希望电脑公司 X Window System 系列丛书之五

X Toolkit Intrinsics

编程手册

(11.4版)

(共九册)

海洋出版社

北京希望电脑公司 X Window System 系列丛书之五

X Toolkit Intrinsic 编程手册

叶欣 罗华平 薛梅 编

海 洋 出 版 社

一九九一年五月·北京

内 容 提 要

X Window System(X 窗口系统)是 80 年代末推出的可移植标准,是软件开发和运行的必备支持系统。

本书是《X Window System 系列丛书》的第五册,它与第六册配套使用。全书包括十三章和五个附录,全面系统地介绍了 X Toolkit 的基本概念和知识,阐明了应用目标基的建立方法和一些典型目标基的使用。

欲购本书的用户可直接与北京 8721 信箱联系,电话 2562329 邮政编码 100080。

X Window System 系列丛书之五 X Toolkit Intrinsics 编程手册 (11.4 版)

叶欣 罗华平 薛梅 编

审校:刘莉蕾

责任编辑:阎世尊

海洋出版社出版发行(北京市复兴门外大街 1 号)

双青印刷厂印刷

开本: 787×1092 1/16 印张: 20.4 字数 495846

1991 年 5 月第一版 1991 年 5 月第一次印刷

印数: 1—3000 册

ISBN 7-5027-1909-1/TP·25

定价: 14.00

序

近几年来,国际计算机界出现了一股“窗口”热,从微型机到小型机到大型机,从工作站到网络,窗口系统已成为软件开发和运行的必备支持系统。

由美国 Massachusetts 大学、IBM 公司、DEC 公司、AT&T 公司、Microsoft 公司等十几家团体联合开发的 X 窗口系统(X Window System)更成为窗口中热点,自从它在八十年代末推出以来,得到了国际软件界的高度评价和肯定。

X Window System 运行于工作站上,以其极其方便的用户接口、强大的编程功能和高质量的代码,改变了整个工作站的世界;X Window System 运行于 386、486 等微机上,更能充分发挥这些微机的特点,利用它们开发更好更完美的软件。

X Window System 推出以后,经过不断的更新和完善,已达到 11.4 版,本版系统在原来的基础上,性能方面又有了很大的提高。

X Window System 主要有核心系统(协议)和建立在其上的应用程序环境工具(Xt、XView、Xlib 等)组成,为了使国内广大计算机用户了解并掌握 X Window System 的基础知识和高级程序设计技术,我们在近几年使用它们的基础上,给合国外最新软件和资料,特编辑了 X Window System 系列丛书,包括:

第一册:《X 协议参考手册》

第二册:《Xlib 编程手册》

第三册:《Xlib 参考手册》

第四册:《X 教程》

第五册:《X Toolkit Intrinsics 编程手册》

第六册:《X Toolkit Intrinsics 参考手册》

第七册:《X Window 编程指南》

第八册:《X View 编程手册》

第九册:《X Window 高级编程指南》

《X Toolkit Intrinsics 编程手册》和《X Toolkit Intrinsics 参考手册》配套使用,主要介绍 X Window 软件包中提供的 Xt 工具,其中第五册描述了如何使用 Xt Intrinsics 库和目标基集合来编写应用程序,并提供了如何用现有的目标基编程以及如何编写自己的目标基的完整例子。

本丛书编辑过程中,得到了国内许多专家、学者的帮助和支持,特别是石清教授,在百忙之中审阅了全书,提出了许多修改意见,刘京同志、薛梅同志、王丽同志、张阳同志为丛本书的完成作出了许多努力,编者在此向他们表示感谢。

由于 X Window System 资料非常新颖,国内又没有可供借鉴的中文参考资料,所以本丛书在编辑过程中,难免会存在一些不足之处,希望各位同仁批评指正。

本丛书出版过程中,得到了中国科学院希望高级电脑技术公司资料部秦人华经理、杨淑欣老师的大力帮助和支持,编者在此表示衷心的感谢。

编者

一九九一年五月

目 录

简介	1
第一章 X Window System 介绍	4
1.1 服务程序和客户程序	6
1.2 软件层次	8
1.3 事件驱动编程	9
1.4 窗口管理程序	10
1.5 X 的扩展	10
第二章 X Toolkit 介绍	11
2.1 用目标基编程	11
2.1.1 目标基的内容	13
2.1.2 目标基类和实例	13
2.1.3 使用资源的目标基可配置性	17
2.1.4 目标基的独立性	18
2.1.5 目标基与应用程序之间的相互作用	18
2.1.6 Xt 和面向对象编程	20
2.2 X Toolkit 应用程序的结构	23
2.3 一个简单的 X Toolkit 应用程序	23
2.3.1 代码	24
2.3.2 编译应用程序	26
2.3.3 应用程序缺省文件	27
2.3.4 硬码(hardcode)和非硬码	28
2.4 连接目标基和应用程序代码	29
2.4.1 反调用	30
2.4.2 动作	32
2.5 资源的进一步讨论	38
2.5.1 在应用程序中设置和获得资源	38
2.5.2 Core 资源	42
2.5.3 其它继承的资源	43
第三章 使用目标基的更多技术	47
3.1 使用复合目标基	47
3.1.1 为一个实例层次设置资源	49
3.1.2 几何管理实践	50
3.2 使用限制目标基	51
3.3 使用弹出	54
3.4 反调用的进一步讨论	60
3.4.1 把数据传递给反调用	60

3.4.2 反调用表	62
3.5 应用程序资源	63
3.5.1 应用程序数据结构	63
3.5.2 资源表	64
3.5.3 获得资源	66
3.6 命令行选项	68
3.6.1 标准命令行选项	68
3.6.2 定义用户自己的命令行选项	68
3.7 防止用户定制目标基资源	72
3.7.1 使用 vararg 接口	72
3.7.2 使用 argList 接口	73
第四章 一个应用程序例子	77
4.1 xbitmap1: 使用 BitmapEdit 目标基的位图编辑器	77
4.1.1 目标基的公共函数	80
4.1.2 应用程序缺省文件	80
4.2 xbitmap2: 增加滚卷条以处理大型位图	80
4.2.1 覆盖翻译	84
4.2.2 翻译表中的动作参数	84
4.2.3 resize_thumbs 动作	84
4.2.4 滚卷条的反调用	85
4.3 xbitamp3: 增加显示位图的图形	87
4.3.1 应用程序的图形	89
4.3.2 写位图文件	92
4.4 xbitmap4: 不使用 BitmapEdit 目标基的位图编辑器	93
第五章 目标基内部	95
5.1 目标基源文件组织	95
5.2 私有头文件——BitmapEdip.h	96
5.2.1 部分结构和记录	96
5.2.2 类部分和类记录	97
5.2.3 实例部分结构和实例记录	98
5.3 目标基实现文件——BitmapEdit.c	99
5.3.1 约束包含文件	100
5.3.2 定义资源表	101
5.3.3 翻译表和动作表	103
5.3.4 声明方法	104
5.3.5 初始化类记录	105
5.3.6 Core 方法介绍	109
5.3.7 为应用程序使用而封装类记录	110
5.3.8 一个方法例子	110
5.4 公共头文件——BitmapEdit.h	112

5.5 编写目标基的过程	114
5.6 约定总结	115
第六章 基本的目标基方法	117
6.1 目标基内部的图形模型	117
6.2 方法 initialize	118
6.2.1 创建 GC	119
6.3 expose 方法	121
6.4 set_values 方法	124
6.5 resize 方法	126
6.6 query_geometry 方法	128
6.7 destroy 方法	130
6.8 目标基框架中的动作	131
第七章 事件、翻译和加速器	134
7.1 翻译表语法	134
7.1.1 指令	135
7.1.2 选择翻译的事件	135
7.1.3 键盘事件的细节	138
7.1.4 其它事件类型的细节	139
7.1.5 修饰词	140
7.1.6 事件序列	144
7.1.7 翻译之间的相互作用	146
7.2 加速器	147
7.2.1 事件传播	149
7.2.2 在多个目标基中安装加速器	151
7.2.3 在代码中定义加速器	151
7.2.4 display_accelerators 方法	152
第八章 更多的输入技术	153
8.1 事件处理程序	153
8.1.1 增加事件处理程序	155
8.1.2 增加不可屏蔽事件处理程序	156
8.1.3 删除事件处理程序	157
8.1.4 增加纯事件处理程序	157
8.2 编写使用专门的事件数据的例程	158
8.2.1 事件类型和结构名称	159
8.3 文件、管道和插座输入	161
8.3.1 获得文件输入	161
8.3.2 获得管道输入	163
8.4 超时(Timeouts)	163
8.4.1 可见性问题	166
8.5 工作过程	167

8.6	事件队列的低级管理	169
8.6.1	XtPending 和 XtPeekEvent	169
8.6.2	事件过滤程序	170
8.6.3	输入敏感性	171
第九章	资源管理和类型转换	172
9.1	资源基础介绍	172
9.2	资源数据库如何工作	175
9.2.1	资源规格说明的格式	175
9.2.2	资源文件的合并	176
9.2.3	语言串	177
9.2.4	备份资源	179
9.2.5	资源匹配算法	180
9.2.6	资源的优先规则	182
9.3	类型转换	185
9.3.1	XtRString 的转换	185
9.3.2	其它已有的类型转换	186
9.3.3	不用转换的特殊的缺省资源	187
9.3.4	类型转换程序的登录	189
9.3.5	显式地激活转换程序	193
9.3.6	编写类型转换程序	194
9.4	子部份和子资源	196
9.4.1	中继方法	197
9.4.2	子资源的管理	197
第十章	客户程序之间通讯	199
10.1	窗口管理程序交互	199
10.1.1	Shell 子类	200
10.1.2	Shell 资源的设置	200
10.1.3	屏幕空间	202
10.1.4	输入方式	203
10.1.5	颜色表	204
10.1.6	图符	206
10.1.7	窗口管理程序的装饰	207
10.2	目标基之间通讯的选择机制	207
10.2.1	原子选择如何工作	208
10.2.2	高亮显示选取的数据(属主方面)	210
10.2.3	用 XtOwnSelection 作出选择(属主方面)	215
10.2.4	对选择的请求(请求者方面)	216
10.2.5	选择的转换(属主方面)	218
10.2.6	选择的最终粘贴过程(请求者方面)	220
10.2.7	选择放弃时(属主方面)	222

10.2.8	选择传送完成时(属主方面).....	222
10.2.9	ICCCM 的一致性.....	223
10.2.10	增量选择如何工作.....	226
10.2.11	其它的选择程序.....	227
第十一章 几何管理.....		228
11.1	复合管理如何工作.....	229
11.1.1	初始几何的协商.....	230
11.1.2	复合目标基中基本的 Core 方法.....	233
11.1.3	孩子目标基的布局.....	235
11.1.4	change_managed 方法.....	237
11.1.5	XtQueryGeometry 和 query_geometry 方法.....	237
11.1.6	XtMakeGeometryRequest 和 geometry_manager 方法.....	238
11.1.7	insert_child 和 delete_child 方法.....	239
11.2	限制目标基如何工作.....	239
11.3	限制目标基的编写.....	240
11.3.1	Core 资源表.....	240
11.3.2	限制资源表.....	240
11.3.3	类结构的初始化.....	242
11.3.4	Constraint 的 initialize 方法.....	244
11.3.5	class_part_init 方法.....	245
11.3.6	geometry_manager 方法.....	245
11.3.7	resize 方法.....	249
11.3.8	Core 和 Constraint 的 set_values 方法.....	251
11.3.9	change_managed 方法.....	251
11.3.10	query_geometry 方法.....	252
11.3.11	重新安排几何的延迟处理.....	252
11.4	组合目标基.....	253
11.5	栈顺序.....	253
第十二章 菜单、辅助目标基和阶式弹出.....		255
12.1	菜单方式及其实现.....	256
12.1.1	如何弹出菜单.....	257
12.1.2	菜单窗格.....	258
12.2	创建和使用菜单的几种方法.....	259
12.2.1	呈现式菜单: 指示器的获取.....	259
12.2.2	下落式菜单.....	266
12.2.3	阶式菜单.....	268
12.2.4	使用第 4 版的 SimpleMenu 目标基.....	272
12.2.5	弹出创建的延迟.....	276
12.3	关于对话框.....	277
12.4	辅助目标基.....	277

12.4.1 辅助目标基内部	280
12.4.2 专用头文件	280
12.4.3 辅助目标基的源文件	281
12.4.4 公共头文件	282
12.4.5 辅助目标基父母	282
第十三章 其它 ToolKit 编程技术	286
13.1 出错和警告	286
13.2 对象	288
13.3 取得信息的宏	289
13.4 键盘聚焦和 accept_focus 方法	290
13.5 键盘翻译	291
13.6 存储分配	292
13.7 动作中继站和动作的直接调用	293
13.8 Xt 的获取函数	293
13.9 文件的查找和国际化	294
13.10 应用上下文	294
13.10.1 多重应用上下文	295
13.10.2 为多重应用上下文重写 XtAppMainLoop	295
13.10.3 多重应用上下文中使用的函数	296
13.11 多重顶层 Shell	296
13.12 多重服务程序之间的连接	296
13.13 类扩展结构	297
附录 A OPEN LOOK 和 Motif	299
附录 B 字体和颜色的描述	319
附录 C 命名习惯	330
附录 D 版本注释	331
附录 E xbitmap 应用程序	352

简介

Xt Intrinsics 库由建立和使用目标基的例程组成。一个目标基(widget)是一个预先建立好的用户接口成分。应用程序是用 Xt Intrinsics 库和一个目标基集合库编写的,以简化 X Window System 应用程序的开发。Xt 和一个目标基集合统称为 X Toolkit。本书描述了如何使用 Xt Intrinsics 库和目标基集合来编写应用程序,并提供了如何用现有的目标基编程以及如何编写自己的目标基的完整例子。

系统销售商提供了一些目标基集合以实现他们特定风格的用户接口,使用最广泛的两个目标基集是 OSF 的 Motif 和 AT&T 的 OPEN LOOK,它们提供了能够编写简单应用程序的一小部分目标基。Xaw 是由 MIT 的 Project Athena 开发的,缩写的 Xaw 代表“Athena Widgets”。Xaw 设计了为一个简单的演示和测试 Intrinsics 的目标基的集合,并不是用来编写实用程序的完整的集合。

本书使用 Athena 目标基来演示如何使用现有的目标基,这等价于实际应用,并为基于 Xt 目标基集合的编程提供了一个很好的介绍。本书基于 Intrinsics 和 Athena 目标基的第四版。

Xt 是用 Xlib 编写的,Xlib 是 X Window System 的最低级语言接口。Xt Intrinsics 和 Xlib 都可由允许用 C 编写的 X 应用程序运行的系统上的 X 标准(由 X Consortium 建立)获得。

内容摘要

XToolkit 的讲座分成两册,是 X Window System 系列丛书的第五册和第六册。

本书是第五册,《X Toolkit Intrinsics 编程手册》。它提供了 X Toolkit 的解释,包括教学材料和大量的编程例子,按任务或主题排列。每章都包括一组 Xt 函数,描述了这些函数基于的基本概念,并通过例子说明了最常用法。本手册既可作为教学时参考也可作为科研时参考。

第六册《X Toolkit Intrinsics 参考手册》包括每个 Xt 函数的参考页以及由 Xt 定义的目标基类。整个手册按字母顺序排列并有一个循环置换为索引表,以及大量的附录。

这两手册应一起使用。为了获得第五册中大量例子,读者需要第六册中每个函数的准确调用顺序。为了完全理解第六册中每个函数的如何使用,除了最有经验的好手外其它人都需要第五册中的解释和例子。

这两本书都包括了 MIT 提供的原始 Toolkit 文档中的材料,虽然在第五册中此材料大部分局限于附录。为了体现 MIT 文档中所有有用的信息,我们尽了最大努力来重新组织并表示成更有用的形式,补充了有关概念、教材、参考、资料和例子。换句话说,本手册不仅是 MIT 文档的一个替代物而且是一个扩充。

前提

本书对读者的面向对象编程和 X Window System 没作任何要求,但读者应该精通 C 语言编程,因为本书中的例子使用 X Toolkit 编程时涉及到 C 语言的特性。另外,熟悉光

删图形原理也有帮助。

即使 Toolkit 可以隐藏 Xlib 提供的低级 X 接口,但有时在编写应用程序和目标基时必须使用 Xlib 函数,因为不存在具有同样功能的 Xt 函数。本书描述了 Xlib 的最常用法,但没有为每个函数提供参考。关于 Xlib 的其它文档,诸如第二册《Xlib 编程手册》和第三册《Xlib 参考手册》也是不可缺少的。

如何使用本手册

第五册解释了使用目标基编写应用程序和编写目标基(设计和编码新的目标基)的知识。

本书前四章把目标基作为“黑盒子”即适当地考虑 Toolkit 的面向对象的原理。这些章节也提供了 X Toolkit 许多成分的概述,因此适合所有类型的读者。

第一章“X Window System”介绍:本章讨论了 X 程序的运行环境。能够用 Xlib 熟练编程的程序员可以跳过这章。

第二章“X Toolkit 介绍”:本章介绍了 Toolkit 编程的基本概念,表明了如何使用现有的目标基编写简单的程序。本章还介绍了 Toolkit 编程的基本概念,诸如资源、翻译、管理、反调用和动作。

第三章“使用目标基的更多技术”:本章介绍了如何使用复杂的目标基,包括复合目标基、限制目标基和弹出,还介绍了如何定义应用程序资源和命令行参数,以及创建一个目标基时如何硬码目标基资源的值,最后介绍了如何创建多个高级窗口,以及如何使用应用程序正文创建更易于移植的应用程序。

第四章“一个应用程序例子”:本章分几个阶段介绍了一个完整例子。首先,介绍了一个位图编辑器程序的简单版本,它假定已存在一个 BitmapEdit 的目标基(实际上在第五章中开发)。然后,开发了两个细化版本,每个版本都演示了不同的 toolkit 编辑技术。最后,介绍了如果位图编辑器在一个应用程序窗口实现而不是用 BitmapEdit 目标基实现时,该应用程序的实现。

下面两章介绍了目标基的内部实现和创建新的目标基的过程。当此信息对所有应用程序编程者不必要时,许多应用程序需要一个常规的目标基来实现它们的特殊的图形能力。

第五章“目标基内部”:本章介绍了目标基内部的代码,所有目标基的大部分代码都相同,因而可看作 Xt 用来实现一个目标基特性的框架。在阅读完本章后,读者应该懂得利用此框架创建自己的目标基的过程。

第六章“基本的目标基方法”:本章介绍了目标基的 Initialize, expose, set_values, destroy, resize 和 query_geometry 方法。(一个目标基的方法是由 Xt 从应用程序中传递给目标基一个无关联,然后自动调用的内部例程。)本章解释了什么时候 Xt 调用哪种方法,并详细介绍了每种方法中应该有什么。在其它情况下,这些方法为目标基中出现的图形作准备并绘出。本章介绍了 Toolkit 增加什么内容到 Xlib 提供的图形模型中,但没有详细介绍如何用 Xlib 绘图,这方面的内容在第二册《Xlib 编程手册》中的第五、六、七章中作了介绍。

后面的几章由应用程序编程者和目标基编程者根据兴趣选读。这些主题有一些在前面的章节中已经介绍了,在后面的章节中作了更完整的讨论。

第七章“事件、翻译和加速表”:本章介绍了翻译表的完整语法,允许用户配置事件

序列到目标基动作的映象。还介绍了加速表,即把一个目标基的事件映象到另一个目标基的动作的机制。

第八章“更多的输入技术”:本章介绍了怎样用事件处理程序处理事件,以及如何使用事件处理程序和动作例程中的事件结构中的信息。然后又介绍了如何获得文件。管道输入,在一个延迟或一定的时间间隔后如何使用超时来调用函数,以及如何使用工作过程作后台处理。本章最后讨论了 Xt 和事件队列直接相互作用的一些低级特性。

第九章“资源管理和类型转换”:本章比较完整地讨论了资源如何工作和它们应如何被使用。本章详细介绍了资源、文件格式和管理资源设置次序的规则,然后介绍了如何增加用户自己的转换程序,以使用户能通过资源来设置应用程序或特定的目标基的数据。最后介绍了子资源以及如何使用它们。

第十章“客户之间的通讯”:本章讨论了客户程序和窗口管理程序之间以及客户程序之间通过 X 服务程序的通讯。客户程序和窗口管理程序的通讯是由 Shell 目标基中的代码执行。客户程序之间的通讯通常是由一个称为 Selection 的进程完成的;这种通讯形式已经在大多数显示正文的目标基中实现了,但用户也可以在自己习惯的目标基中实现。Selection 也可以传递其它类型的数据,如图形。

第十一章“几何管理”:本章讨论了复合和限制目标基如何管理目标基的布局,以及如何编写用户自己的简单的复合和限制目标基。

第十二章“菜单、辅助目标基和阶式弹出”:本章介绍了菜单如何工作以及创建菜单目标基的几种方法;其中一种方法是采用无窗口的目标基或辅助目标基。还介绍了如何使用 Xt 弹出机制的更高级特性,包括典型的阶流式布置来实现阶式弹出菜单。

第十三章“其它的 Toolkit 编程技术”:本章介绍了本书其它地方没有出现的 Xt 函数,这些函数包括错误和警告处理、情况转换等等。

附录 A“OPEN LOOK 和 Motif”:本附录介绍了 AT&T 的 OPEN LOOK 目标基集和 OSF 的 Motif 中利用的目标基的概貌。并将这些目标基和 Athena 目标基集合相比较。

附录 B“字体和颜色描述”:本附录给出了当把字体和颜色作为资源描述时,使用的值的信息。

附录 C“命名习惯”:本附录介绍了命名目标基和目标基代码成分的习惯集合。

附录 D“版本注释”:本附录介绍了版本 3 和版本 4,并注释了对于那些熟悉版本 3 的用户来说重要的变化。

附录 E“bitmap 应用程序”。本附录列出了在第 4、5 章介绍的 BitmapEdit 目标基的完整代码。

第一章 X Window System 介绍

X Window System(简称 X)是一个与硬件和操作系统无关的窗口系统。它是由 MIT 和 DEC 联合开发的,已经被计算机工业界采纳为图形应用一个标准。

X 控制一个位图显示,位图中的每个象素都可单独控制,这就允许应用程序绘制图形以及正文。直到最近,屏幕像素的单独控制才广泛地应用于个人计算机和价格昂贵的技术工作站,大多数通用计算机仅局限于正文终端输出。X 个人计算机和更强大的计算机带来了一致的图形输出。图 1-1 比较了一个 X 应用程序和在传统正文终端上运行的应用程序。

与其它窗口系统类似,X 把屏幕分成多个称为窗口的输入输出区域。使用终端模拟器,窗口就象一个虚拟终端运行普通基于正文的应用程序。但是如图 1-1 所示,窗口中也能运行利用位图的图形能力设计的应用程序。

X 从指示器接收用户的输入;指示器通常是鼠标,也可以是跟踪球或输入板。指示器允许用户通过指向屏幕上的目标如菜单或命令按钮而不用键盘来控制程序,这种使用程序的方法比传统的键盘控制方法容易学习,因为这更加直观了。图 1-2 列举了一个三键鼠标选择菜单项的应用程序。

当然 X 也可以处理键盘输入,指示器操纵从窗口到窗口的键盘输入,每次只有一个窗口能接收键盘输入。

在 X 中,与在许多其它窗口系统中一样,每个应用程序不必只由一个窗口组成;应用程序的任何一部分都可有自己独立的子窗口,这使得应用程序代码中简化了输入输出管

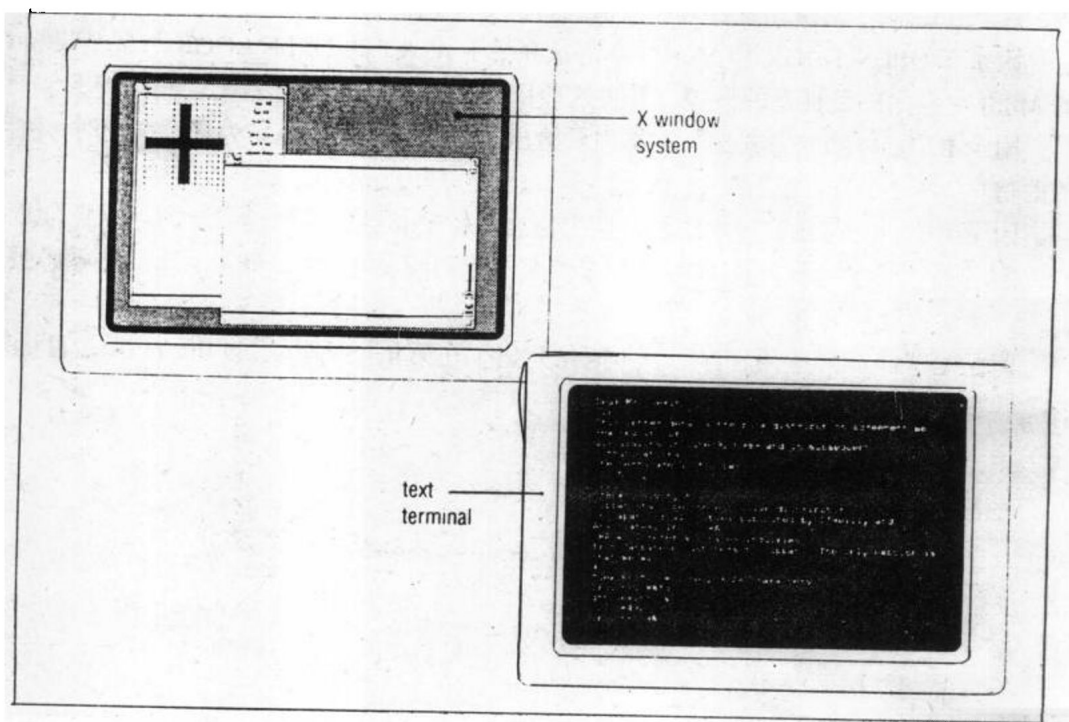


图 1-1 X 应用程序和传统正文终端上的应用程序

理。子窗口只有在它们的父窗口中才可见。

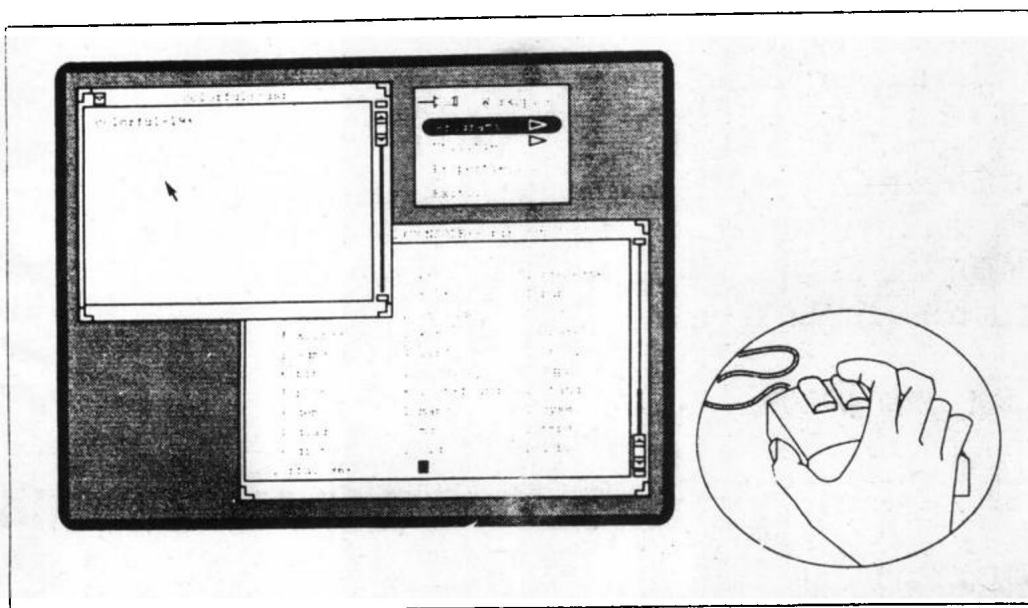


图 1-2 三键鼠标指示指示器选择一个菜单项

窗口是一个矩形，四周有显示边界。每个窗口都有自己的坐标系，坐标原点在窗口内部边界的左上角。应用程序和用户 can 改变窗口的大小。图 1-3 列举了多个虚拟终端运行

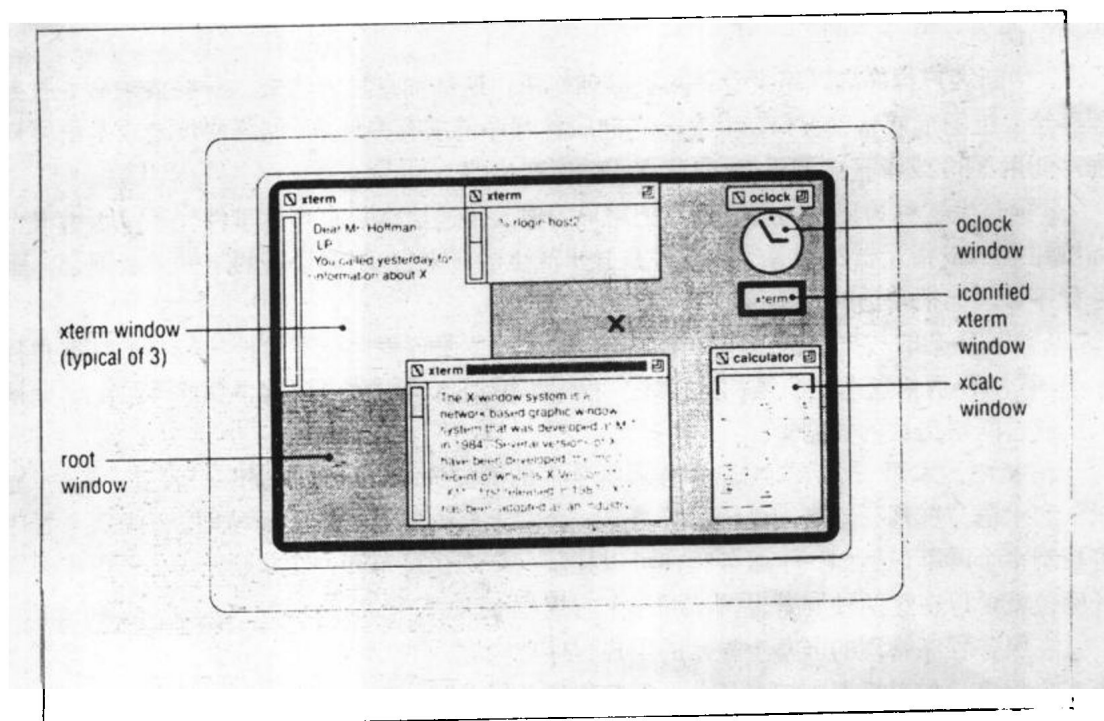


图 1-3 典型 X Window System 的屏幕布置

的典型屏幕；这个屏幕也表示一些运行在各自的窗口内的应用程序，如 `xmh`，`xclock`，`xload`。

X 支持彩色和黑白两种显示。

上述的许多特性在其它的窗口系统中也具有。X 的独特之处在于它基于一个网络协议而不是特定的系统过程或系统调用。这个网络协议使得 X 适用于不同的计算机或操作系统，它允许程序运行在一种结构的计算机或操作系统之上，而在另外一种计算机或操作系统显示。由于这种独特的设计，X 可以使不同的计算机联成一个网络，例如一个计算密集型应用程序在主机上运行，它可以从局域网上的工作站接收输入或显示输出。对用户来说，应用程序只是在工作站上运行。

1.1 服务程序和客户程序

为了允许程序在一台机器上运行而另一台机器上显示，X 被设计成两类进程之间的一个网络协议(预先定义好请求和回答集合)，其中应用程序称为客户程序，其它控制显示硬件、键盘和指示器的程序称为服务程序。

用户坐在计算机前运行服务程序。首先，服务程序这个术语的用法似乎有点奇怪，因为文件和打印服务程序通常是远程计算机。但是，这两个用法是一致的。局网显示服务程序可通过网络访问其它系统，因为这些系统的 X 服务程序象其他类型的服务程序一样工作。

X 服务程序是用户程序和局部系统资源如键盘和屏幕之间的媒介；它包括了所有与设备有关的代码，这使应用程序与显示硬件无关。X 服务程序执行下面的任务：

- 允许多个客户程序访问同一显示，服务程序可能拒绝来自在某一机器上运行的客户程序的访问。

- 中断客户程序的网络消息并对其施加作用。这些消息称为请求。一些请求命令服务程序绘二维图形或移动窗口，其它一些向服务程序请求有关信息。网络协议请求是由客户程序调用 `Xlib` 或者直接调用 `Xt` 或其它函数库产生的。

- 通过发送称为事件的网络消息把用户的输入传递给客户程序。事件代表键盘键或鼠标按钮按下或指示器移动等等。事件是异步产生的，来自不同设备的事件可能混合，服务程序必须能把合适的事件传递给相应的客户程序。

- 维护复杂的数据结构，包括窗口和字体，以便服务程序能有效地执行任务。客户程序是通过 ID 引用这些抽象结构的，服务程序维护抽象结构减少了客户程序维护的数据量以及通过网络传送的数据量。

在 X 中，术语“显示”常常用作服务程序的同义词即显示服务程序。但是，术语“显示”和术语“屏幕”不是同义词，屏幕是一个在上面绘制图形的实际硬件。一个服务程序可控制多个屏幕，例如一个服务程序可以控制一个彩色屏幕和一个单色屏幕。用户不用离开座位就可以在这两种屏幕上调试同一个应用程序。

在服务程序管理的屏幕上显示的用户程序称为它的客户程序。一个服务程序可连接多个客户程序。如果机器支持多任务，客户程序和服务程序就可运行在同一台机器上，客户程序也可运行在网络中的其它机器上。在任何一种情况下客户程序都使用 X 协议来发送绘图或向服务程序查询信息的请求，服务程序使用 X 协议来发送用户输入或用户程序请求的

应答信息。从客户程序到服务程序和从服务程序到客户程序的所有通讯都用 X 协议进行。客户程序和服务程序之间的一条通讯路径称为一个连接。

用户在网络中不同的主机上运行程序是很普通的，并且所有调用和显示窗口都在一个屏幕上(见图 1-4)。远程运行的客户程序可以通过网络应用程序 `rlogin` 或 `rsh` 从远程计算机或局域计算机上启动。

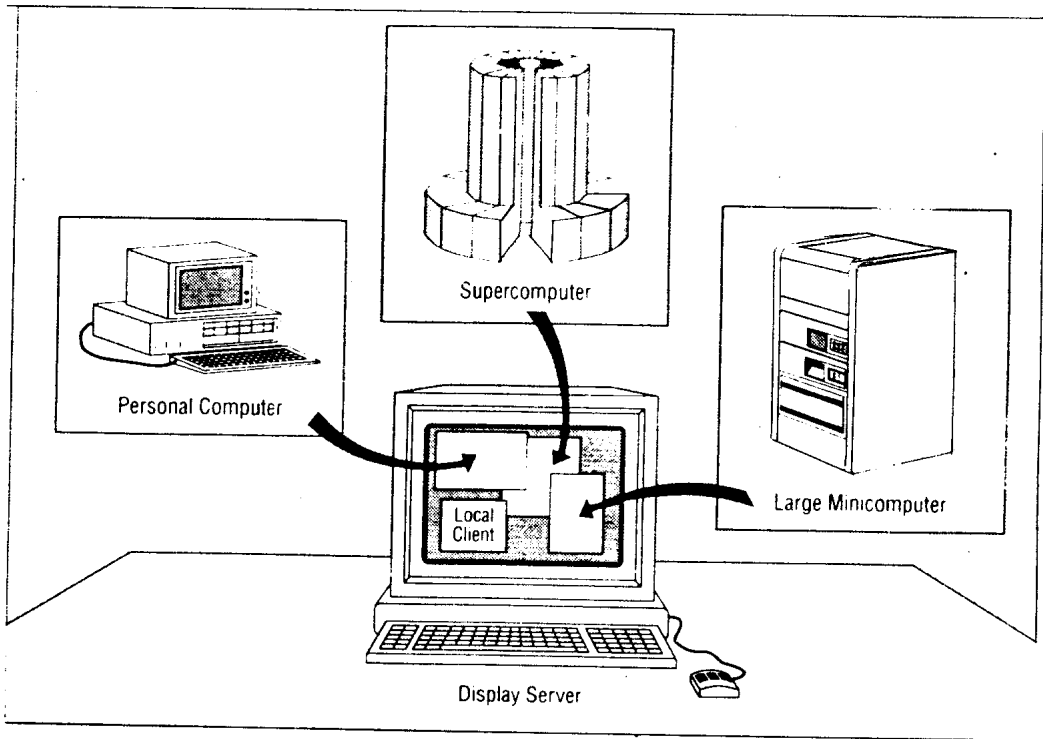


图 1-4 应用程序通过网络运行在任何系统上

这种风格的使用称为分布式处理，这个概念最重要的应用是为自己没有图形软件包的强大系统提供图形输出。分布式处理也可帮助解决系统负载不平衡的问题，当一个主机过载时，在此机器上运行客户程序的用户可以把一些客户程序安排到其它主机上运行。将来可能会有自动平衡负载的应用程序，但是当前这样的远程操作都是手工进行的。一点也不奇怪，看见 X 环境中的用户用多个 `xload` 负载来监控应用程序在网络上的不同系统上运行，并在他们的屏幕上显示，以使用户能看见网络负载的平衡状况。

在结束服务程序和客户程序讨论之前，我们应该提及 PC 服务器和 X 终端。现在已经有软件允许各种类型的 PC 服务器象 X 服务程序一样运行。X 终端是用来只运行 X 服务程序的专用设备，并通过局域网连接远程系统。PC 服务器和 X 终端为用户提供 X 屏幕最便宜的方法。因为大多数 PC 服务器使用单任务操作系统，它们不能象 X 服务程序那样同时运行多个客户程序，因此它们也需要一个网络适配器连接到客户程序运行的那个系统上。

X 终端和 PC 服务器都显示了 X 客户程序和服务程序模型的强大。即使 PC 服务器和终端本身不能执行多任务，它们也给用户提供了多任务工作站的效果，因为它们能够同时与运行在远程多任务系统上的多个用户程序相互作用。