



北京希望电脑公司 X Window System 系列丛书之四

X 教程

(11.4版)

(共九册)

海洋出版社

中国科学院希望高级工程技术人员 Windows 系列丛书之四

Windows

(11.4 版)

叶欣 唐晓菲 吴新根 编

海 洋 出 版 社

1991 年 5 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》中的一部基础书。它包括十二章和七个附录,全面系统地介绍了 X 的基本概念和术语;阐明了窗口建立、操作和破坏方法以及如何使用 X;讨论了事件处理的高级技术问题及应用程序之间的通讯技术等。

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 系列丛书之四

X 教程

(11.4 版)

叶欣 唐晓菲 吴新根 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: 24.5 字数: 596232

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: 16.00

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 80 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,特编译了 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 教程》是一本有关 X 的基础书,本书全面介绍 X 的基本概念和术语,并说明了 X 中事件处理的方法和应用程序之间的通信技术。

本丛书编译过程中,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、张阳同志为本丛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,所以本丛书在编译过程中,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

1991 年 5 月

目 录

前言	1
第一章 简介	2
1.1 X 环境	2
1.2 X 窗口系统的目标	3
1.2.1 厂家和模型独立	4
1.2.2 输出功能	5
1.2.3 输入功能	5
1.2.4 信息共享能力	5
1.2.5 并发性	6
1.3 使用 X	6
1.3.1 窗口管理器	7
1.4 用 X 进行程序设计	7
1.5 协议说明	8
1.6 工具箱	8
1.7 本书的使用	9
1.7.1 要求	9
1.7.2 各章概述	9
1.7.3 调用	9
1.7.4 事件	10
1.7.5 术语	10
第二章 Hello, World	11
2.1 helloworld.c 做什么	12
2.2 helloworld.c 描述	12
2.3 helloworld.c 分析	12
2.3.1 头文件	12
2.3.2 声明	13
2.3.3 初始化	13
2.3.4 输入事件请求	14
2.3.5 窗口映射	15
2.3.6 读事件主循环	15
2.3.7 建立和运行 helloworld.c	17
2.3.8 X 窗口系统应用结构化	17
第三章 X 概念	19
3.1 X 网络协议	19
3.1.1 单道协议请求信息	20
3.1.2 事件信息	21

3.1.3	错误事件消息	21
3.2	Xlib 过程接口	21
3.2.1	头文件	22
3.2.2	请求和方便函数	22
3.2.3	服务函数	22
3.2.4	信息宏和函数	22
3.3	显示连接	22
3.3.1	显示名	23
3.3.2	打开显示连接	25
3.3.3	关闭显示连接	26
3.4	资源	26
3.4.1	控制资源生存期	28
3.4.2	显示结构信息	29
3.5	事件	31
3.5.1	事件和窗口	32
3.5.2	事件类型汇总	32
3.5.3	请求事件	34
3.5.4	异步事件生成	35
3.5.5	从事件队列接收事件	35
3.5.6	请求缓冲区的显示控制	39
3.6	错误处理	40
3.6.1	错误代码	41
3.6.2	缺省 Xlib 错误事件处理器	41
3.6.3	应用定义错误事件处理	41
3.6.4	致命 I/O 处理	44
3.7	小结	44
第四章 窗口		47
4.1	窗口及台式模型	47
4.1.1	窗口级和压栈	48
4.1.2	窗口几何	51
4.1.3	窗口状态和生存周期	51
4.2	简化窗口	54
4.2.1	窗口建立	55
4.2.2	请求窗口事件	57
4.2.3	窗口映射	57
4.2.4	背景和边界	58
4.2.5	撤销窗口映射	59
4.2.6	破坏窗口	60
4.3	窗口结构	61
4.3.1	操作窗口几何	61

4.3.2	操作窗口压栈顺序	66
4.3.3	XConfigureWindow	68
4.4	属性和特性	70
4.4.1	窗口特性	70
4.4.2	窗口属性	71
4.5	高级窗口操作	78
4.5.1	XSetWindowAttributes 结构和屏蔽	78
4.5.2	XCreateWindow	79
4.5.3	XChangeWindowAttributes	81
4.5.4	XSetWindowBackgroundPixmap	81
4.5.5	XSetWindowBorderPixmap	82
4.5.6	XGetWindowAttributes	82
4.5.7	XQueryTree	83
4.5.8	XReparentWindow	84
4.6	窗口管理器交互	85
4.6.1	窗口管理器的干涉	85
4.6.2	无管理（瞬时）窗口	85
4.6.3	可管理窗口	86
4.7	窗口定尺寸方法	94
4.8	通知事件	96
4.8.1	请求窗口通知事件	97
4.8.2	接收和处理窗口通知事件	97
4.9	小结	104
第五章	图形	106
5.1	图形流水线和 GC	107
5.1.1	像素选择阶段	108
5.1.2	模式化阶段	109
5.1.3	GC 剪裁阶段	109
5.1.4	窗口剪裁阶段	109
5.1.5	光栅输出阶段	109
5.2	操作 GC 资源	110
5.2.1	通用 GC 操作请求	111
5.2.2	方便函数	114
5.3	画图	117
5.3.1	点	119
5.3.2	线	120
5.3.3	弧、圆弧和椭圆弧	126
5.3.4	矩形	129
5.3.5	多边形	130
5.4	Expose 事件	133

5.4.1	引起 Expose 事件的原因	133
5.4.2	窗口的局部暴露	134
5.4.3	请求 Expose 事件	135
5.4.4	接收和处理 Expose 事件	135
5.5	清除窗口	136
5.5.1	XClearWindow	137
5.5.2	XClearArea	137
5.6	拷贝区域	138
5.6.1	XCopyArea	139
5.6.2	XCopyPlane	139
5.6.3	GraphicsExpose 事件和 NoExpose 事件	140
5.6.4	处理 GraphicsExpose 事件和 NoExpose 事件	142
5.7	高级画图技术	143
5.7.1	填充形式	144
5.7.2	剪裁	148
5.8	工作站性能	152
5.9	图形尺寸	153
5.10	小结	154
第六章	文本	157
6.1	字体	157
6.2	简单字体选择	159
6.2.1	装入字体	160
6.2.2	设置 GC 中 fond 属性	160
6.2.3	卸载字体	160
6.3	绘制字符串	161
6.3.1	XDrawString	162
6.3.2	XDrawImageString	163
6.3.3	XDrawText	164
6.4	字体结构	165
6.4.1	装入字体结构	166
6.4.2	字体结构与 GC 字体属性	167
6.4.3	释放字体结构	167
6.4.4	计算字符串的宽	167
6.4.5	计算字符的维数	168
6.4.6	字体特性	169
6.5	搜索字体	172
6.5.1	列举可用字体	172
6.5.2	工作站的字体搜索路径	174
6.6	16 位字符串	174
6.6.1	绘制 16 位串	175

6.6.2	计算 16 位串维度	176
6.7	小结	177
第七章	颜色	179
7.1	颜色概念	179
7.1.1	平面与像素值	179
7.1.2	使用像素值	180
7.1.3	红色、绿色、蓝色	181
7.1.4	可见类	182
7.1.5	颜色图	183
7.2	工作站能力	186
7.2.1	信息函数小结	187
7.2.2	工作站与屏幕缺省信息	188
7.2.3	缺省黑色与白色像素值	188
7.2.4	缺省颜色图信息	188
7.2.5	缺省可见集信息	189
7.2.6	缺省深度信息	190
7.2.7	选择可见集结构	191
7.3	像素值与颜色元	192
7.3.1	策略：共享颜色元	192
7.3.2	策略：标准颜色图	196
7.3.3	策略：独享颜色元	200
7.4	服务函数	209
7.4.1	为指定颜色寻找 RGB 值	209
7.4.2	释放颜色元	211
7.4.3	查询颜色元 RGB 的内容	212
7.5	单色与灰度级	212
7.5.1	单平面单色	213
7.5.2	多平面灰度级	214
7.6	颜色图操纵	214
7.6.1	建立颜色图	215
7.6.2	把颜色图和窗口联系起来	215
7.6.3	拷贝颜色图	216
7.6.4	释放颜色图	216
7.6.5	装载与拆卸颜色图	217
7.6.6	处理 ColormapNotify 事件	217
7.7	小结	218
第八章	像图、位图与图像	221
8.1	像图资源	221
8.1.1	使用像图	223
8.1.2	建立像图	223

8.1.3	确定像图大小	224
8.1.4	像图删除	225
8.2	位图	225
8.2.1	位图文件格式	225
8.2.2	读位图文件	226
8.2.3	写位图文件	228
8.2.4	从数据中建立位图资源	228
8.3	图像	230
8.3.1	图像建立	231
8.3.2	子图像	233
8.3.3	像素存取	234
8.3.4	发送图像到工作站	235
8.3.5	从工作站检索图像	236
8.3.6	图像删除	238
8.3.7	图像数据结构	238
8.4	小结	243
第九章	鼠标与指针	244
9.1	鼠标使用策略	244
9.2	指针控制	245
9.2.1	读当前指针位置	246
9.2.2	控制指针动作特性	248
9.2.3	移动——扭曲——指针	249
9.3	光标	251
9.3.1	建立、改变和破坏光标	252
9.3.2	显示窗口内光标	254
9.4	鼠标事件	255
9.4.1	请求 ButtonPress 和 ButtonRelease 事件	255
9.4.2	请求 EnterNotify 和 LeaveNotify 事件	257
9.4.3	请求 MotionNotify 事件	257
9.4.5	如何处理指针动作暗示	261
9.4.6	EnterNotify 和 LeaveNotify 事件——细节	263
9.4.7	在嵌套窗口内的事件	266
9.4.8	InputOnly 窗口	266
9.5	小结	267
第十章	键盘	269
10.1	键盘事件	269
10.1.1	控制键盘事件的发送	270
10.1.2	接收和处理键盘事件	271
10.2	键码、键图、键符号和文本	273
10.2.1	基础	274

10.2.2	键转换服务函数	275
10.2.3	键符号分类宏	278
10.3	键盘聚焦	278
10.3.1	XGetInputFocus	279
10.3.2	XSetInputFocus	279
10.3.3	聚焦模型	280
10.3.4	请求 FocusOut 和 FocusIn 事件	281
10.3.5	FocusIn 和 FocusOut 事件——细节	281
10.3.6	请求 KeymapNotify 事件	287
10.3.7	接收和处理 KeymapNotify 事件	287
10.4	控制键盘	288
10.4.1	轮询键盘	288
10.4.2	鸣键盘上的铃	289
10.4.3	键盘控制设置	289
10.5	小结	292
第十一章	高级事件处理	294
11.1	轮询队列	294
11.1.1	在事件队列中往前看几个	295
11.1.2	根据窗口及事件类型读事件	295
11.1.3	用布尔函数选择事件	297
11.2	事件压缩	298
11.2.1	压缩 MotionNotify 事件	299
11.2.2	压缩 EnterNotify 和 LeaveNotify 事件对	299
11.3	多个显示器连接	301
11.4	放回和发送事件	303
11.5	获取指针	305
11.5.1	开始一个指针获取	306
11.5.2	修改一个指针获取	308
11.5.3	结束一个指针获取	308
11.5.4	指针获取与 EnterNotify/LeaveNotify 事件	308
11.5.5	在 ButtonPress 和 ButtonRelease 之间	309
11.6	获取鼠标按钮	310
11.7	获取键盘	312
11.7.1	开始一个键盘获取	312
11.7.2	结束一个键盘获取	314
11.7.3	键盘获取和聚焦变化事件	314
11.8	获取特别键	315
11.9	同时发送被获取的事件	316
11.10	被动获取的活动	319
11.11	小结	319

第十二章 应用程序之间的通信	321
12.1 截取缓冲区(Cut Buffer)	321
12.1.1 在截取缓冲区内存储数据	322
12.1.2 在截取缓冲区中检查数据	323
12.1.3 在截取缓冲区中交换数据	323
12.2 特性	324
12.2.1 特性的生存期	325
12.2.2 原子和特性名	326
12.2.3 特性数据格式和类型	327
12.2.4 建立和存储数据到特性中	328
12.2.5 从特性中检索数据	329
12.2.6 交换特性数据	332
12.2.7 为一个窗口列表特性	333
12.2.8 删除特性	333
12.2.9 PropertyNotify 事件	334
12.2.10 使用特性传送信息	335
12.3 选择	336
12.3.1 声明选择占有权	337
12.3.2 失去选择占有权	339
12.3.3 请求选择的数据	340
12.3.4 为选择响应请求	345
12.4 小结	348
附录 A helloworld.c	350
附录 B X 协议请求代码	352
附录 C LATIN-1 和标准键符号	354
附录 D 字体	359
附录 E 颜色名字	366
附录 F 标准光标形状与符号	368
附录 G 预定义原子	371

前 言

X Window 系统是在美国马萨诸塞大学(MIT)的 **Athena** 工程上开发的一个可移植软件标准。它能控制工程工作站的显示并提供给应用软件如页格式编辑器和计算机辅助设计软件包一个标准环境,用 **X** 来进行工作站显示操作的应用,可以很容易地在不同厂家的各种工作stations上运行。

X Window 系统能够激发工作stations上优秀应用软件的开发。本书面向使用 **X11** 版本来开发软件的用户。

第一章 简介

在早期的交互式计算处理中，用户在打字机似的键盘上穿孔输入命令。随着时间的推移，视频显示终端的大量出现使用户节省了纸张，但人们仍然使用字母和数字与计算机进行通信。直到计算机工作站的引入，计算机图形技术、位图视频显示、键盘输入和指示器输入才被集大成于一体。

现已证明集成计算机工作站是很成功的：用户们要求计算机既有高功能又易于使用，而工作站正好能满足这个要求。但工作站的出现增加了应用软件开发者的负担，提供一个基于文本的简单问答对话已不够了：用户要求能够充分利用工作站图形能力的交互式软件。

在努力使向完全交互程序的转换变得容易的过程中，工作站硬件的制造商提供了一些子例程以显示他们硬件的功能和特征(如彩色图形和鼠标操作)。这些已说明是很成功的：集成工作站的销售正在增加，在开始的七年中，从无到有地增加到每年总量为 15 美亿。

但工作站应用软件的开发者仍面临为每个工作站硬件产品开发各种图形接口的问题和费用。各种工作站的两个大用户是美国“*Athena 工程*”和“*计算机科学实验室*”，它们都属 MIT。这两个组织负责开发和使用大量的工作站应用程序。为了避免为每种工作站都设计一种应用，便产生了 X Window 系统。X Window 系统为 MIT 用工作站硬件的公共和可移植接口设计工作站应用程序提供了一个途径。*Athena 工程*由 IBM 和 DEC 赞助。这两个公司看到了 X 的价值，并且具有远见地及慷慨地为 X 的开发提供了物质资源。

1987 年 1 月，十多家销售商声称要联合进行 X 的标准化及改进工作。到现在为止，几乎所有工作站都接受 X 窗口系统是其工作站硬件的标准接口。更进一步说，多数厂家都承认了 X 的未来：它们努力保证其 X 产品的速度和可靠性。

1.1 X 环境

X 窗口系统是工程工作站的软件环境。它为应用程序员和用户提供了丰富和复杂的环境。X 的基础是 base window system(基本窗口系统)。整个 X 环境由基本窗口系统之上的几个层次组成。

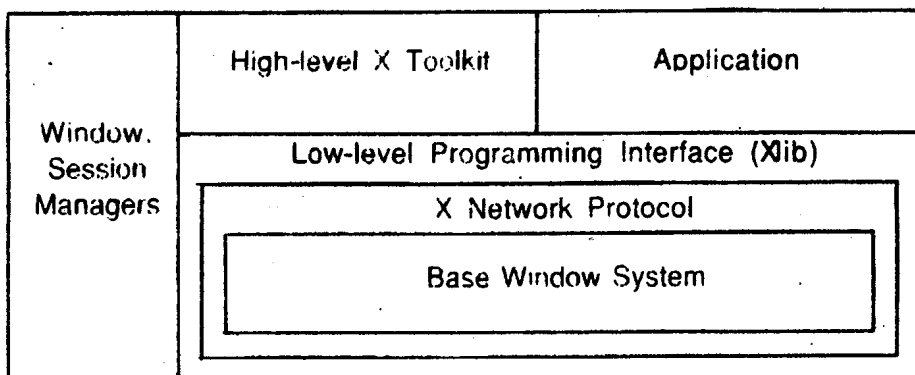


图 1-1 X 结构

基本窗口系统用 X 网络协议与外界接口。网络协议接口的设计既能工作于单中央处理机(CPU)中,也能工作于双 CPU 中。网络协议的存在给予了设备/厂家以在 X 上的独立性和网络透明性。如图 1-1 所示,协议是到基本窗口系统的唯一接口。X 不为有特权的软件提供任何特殊的“后门”接口。这就意味着象对话管理器和窗口管理器这样的软件被 X 当作应用软件而不是特权系统软件。

X 应用程序通常不直接使用网络协议。而是通过程序设计接口进行工作。X 窗口系统提供了称作 Xlib 的 C 语言子程序软件包,其目的在于让应用与网络协议相接口从而与基本窗口系统相连。多数应用程序应使用高级 X 工具箱来屏蔽网络协议的某些复杂性。

用户与工作站的接口是由基本窗口系统、X 工具箱、窗口管理器、对话管理器和应用软件提供的功能的组合体。X 的中心原则是基本窗口系统提供的是机制而不是策略。基本窗口系统提供机制,如输出画图功能。用户界面策略留作别的一部分处理。

例如,X 允许在屏幕上的窗口相互覆盖,并允许输出部分地使窗口模糊:这是机制。应用程序和高级工具箱可能使用弹出菜单(含有选项的窗口,它弹出并部分地模糊其它窗口,参见图 4-4)作为人机界面的一个部分:这是策略。

有些用户喜欢策略,这样窗口被精确地一个一个拼接不是覆盖。在 X 中拼接是可能的:控制屏幕上窗口组织的策略是由一个窗口管理器应用来实施的,因而希望拼接的用户应使用一个拼接窗口管理器。若显示窗口的机制要求拼接窗口,则要注意,要实现一个弹出菜单策略是很困难的。X 的基本窗口系统的一个优点在于它提供了足够的机制来支持各种不同的人机接口策略,把这些策略的实现和实施留给其它软件。

本书,如同 X 的基本窗口系统,是关于机制的:它说明如何通过 Xlib 接口使用 X 基本窗口系统。Xlib 象本书一样,有几个月的稳定性,并可预其具有三到五年的生命力。在你读这本时候,要记住与 X 接口的高级工具箱是在不断发展的。许多应用程序将能直接使用工具箱,而几乎不用 Xlib。但许多应用程序,尤其是使用交互图形技术的应用程序将要直接使用 Xlib。本书的目的是帮助工作站程序员理解从 Xlib 接口看 X 基本窗口系统的开发原因。

1.2 X 窗口系统的目标

X 窗口系统的目的是为工作站软件提供一个网络透明和厂家独立的操作环境。操作环境支持在各种彩色和单色工作站上的多窗口覆盖。

网络透明性是指在一种 CPU 上运行的 X 应用程序可用连接到同样的 CPU 或某种别的 CPU 显示来说明其输出。X 的许多能力都是由其到当地网络环境的集成导出的:该环境允许各种计算模式,从当地运行应用独立工作站到把不同工作站当作终端使用的共享时间主机。所有这些模式都可与 X 环境共存。X 的网络透明性不偏向任何一种计算模式,而是把它们综合起来形成一个适于小至办公室大至一个学校校园的各种地方使用的灵活网络计算环境。有关 X 的经验表明,在局域网环境下,与本地(单 CPU)应用比较,网络 X 应用要很少或没有性能损失。

网络透明性隐含着应用可以在任何 CPU 上运行。例如,一个要求大规模共享数据库的应用程序可以在含有数据库的 CPU 上运行,并利用 X 通过网络连接用户工作。类似地,要求进行各种计算的应用可在一个连网超级计算机上运行。当然,许多应用程序(如文本编辑器、终端仿真程序和窗口管理器)都局域地运行在与工作站显示硬件相同的 CPU 上。

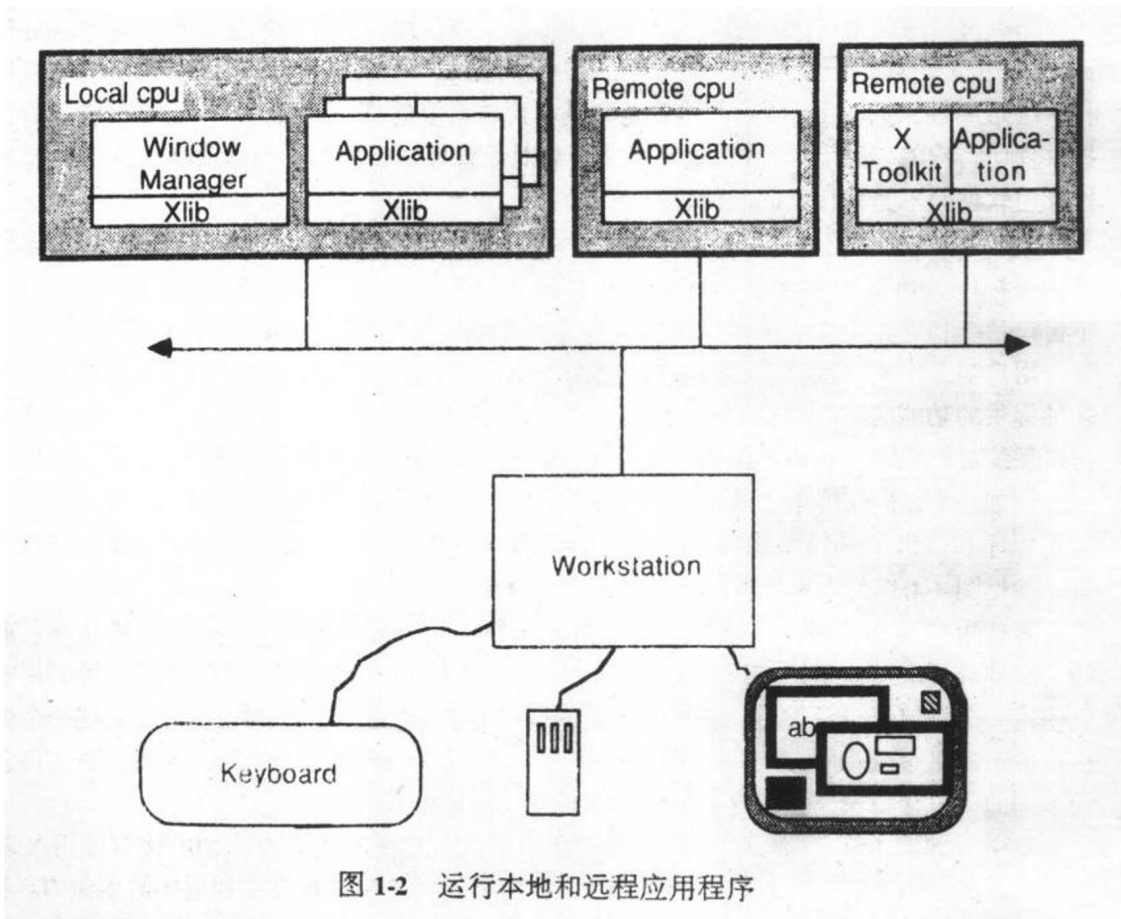


图 1-2 运行本地和远程应用程序

没有必要用某种方法使 X 程序通过网络连到一个远程工作站上。

同样地，单个 X 应用程序可同时把输出窗口显示到许多不同的工作站上。第十一章将描述一个应用程序如何与多显示连接以一起工作。

1.2.1 厂家和模型独立

X 应用是可移植的。它们与 X 打交道，因而它们不知道特定工作站显示硬件的细节。只要一个 X 应用可以建立一个到工作站的连接，它便可以使用该工作站上基本窗口系统所有的功能。由于工作站硬件被协议隐蔽，运行在来自一个厂家的 CPU 上的 X 应用可使用任何工作站模型，它可以来自厂家，也可以来自别的厂家。没有必要一个应用进行重编译或重连接以使之能够在各种 X 工作站上使用。

象其它程序一样，X 应用必须对应其将在上面运行的特定 CPU 和操作系统环境进行编译和连接。例如，当应用从一个 Hewlett-Packard 移到一个 Apollo 环境，或从一个 DEC VAX/Vlrix 环境移到一个 DEC VAX/VMSX 环境时，有必要重新编译和重新连接。X 工作站厂家提供了 Xlib 的预编译版本以及可能与应用连接的其它子程序库。

厂家独立性的一个问题是字节顺序问题。有些厂家的计算机(如 DEC 公司的 VAX 计算机)是“小结尾”的：即它们首先存放多字节最低字节。其它厂家的(被几个工作站厂家使用的 Motorola MC68000 系列)是“大结尾”的：它们首先存放最高有效字节。X 通过使工作站跟踪其是否应该重排列在每个应用结束的协议中的字节，从而彻底地解决了该问题。

改变字节顺序的任务由工作站完成，不由应用完成，而一个“小结尾”的工作站只重排来自一个“大结尾”应用字节，反之也一样。

1.2.2 输出功能

X 的基本窗口系统提供了一组各种特征的位图图形操作。应用和高级工具箱使用 X 的图形操作在工作站屏幕上画出信息以使用户能看见。X 提供了下列图形能力：

- X 把显示屏幕组织成覆盖窗口层次。每个应用可以使用任意多个窗口，重置窗口尺寸和把窗口压栈等。本书第四章将讨论应用使用 X 的多窗口功能的方法。

- X 提供了画图功能。X 的图形操作是立即的而不是面向显示表的：工作站不保留图形操作系列，而是立即画出。

X 的画图操作是面向位图的：应用用窗口中整型像素地址来指定所有的操作。所有 X 图形操作被指定到某一特定窗口，因而应用可在某窗口画图而不考虑它们的窗口在屏幕上的什么地方。第五章将描述如何画点、线、矩形、和多边形。

- X 能让应用画出高质量的文本。X 的文本支持许多种应用，从视频显示终端仿真程序到多语言字处理程序。第六章介绍如何画文本。

- 对于各种彩色工作站以及黑白模型来说，X 是很有用的。第七章将描述如何使用 X 中的颜色，以及如何书写可在彩色工作站以及黑白工作站上工作的应用。

- 对于映象的显示、操作和捕获，X 是很有用的。第十一章将描述如何使用 X 的映象功能。

1.2.3 输入功能

X 提供了一种把用户输入分配到应用程序中的综合机制。X 的输入分布部分有下列功能：

- 用户通过在工作站键盘上按键或操作工作站的鼠标产生输入。X 的基本窗口系统把该输入作为事件形式分配到应用中。第三章将描述基本窗口系统如何把事件放到一个队列中，应用可以从该队列读取事件并加以处理。

- 许多事件允许应用弄清楚有关窗口层次的变化。第四、第五和第七章分别描述与窗口、图形和颜色有关的事件。

- 来自工作站指示器或鼠标的输出以事件形式出现。基本窗口系统对这些指示器事件进行安排以把它们送往适当的应用。即使是在许多应用同时在工作站上运行一样。第九章描述应用如何处理指针输入事件。

- 支持各种各样的打字键盘。第十章描述基本窗口系统的键盘支持和如何使用。

1.2.4 信息共享能力

当几个应用程序在工作站上运行时，要相互协调工作(例如允许剪切和剪贴操作)。X 为应用提供了通信手段：

- 应用可使用简单的剪切和剪贴功能以共享简单的非结构数据，如字符串。

- 应用可通过 X 特性共享结构数据。特性是可与窗口连接的数据项。每个特性有一个名字、一个数据类型和一个数据格式。

- 选择为应用提供了应用间的交互数据结束的一种复杂方式。