

Windows 动态数据交换程序设计

——用 Borland C++ 和
Turbo C++ for Windows

程铁皋 程文斌 等编

北京航空航天大学出版社

Windows

386173

Windows 动态数据交换程序设计
——用 Borland C++ 和 Turbo C++ for Windows

程铁皋 程文斌等 编



TP312
C771-2

北京航空航天大学出版社

(京)新登字 166 号

图书在版编目(CIP)数据

Windows 动态数据交换程序设计：用 Borland C++ 和 Turbo C++ for Windows/程铁阜等编著. —北京：北京航空航天大学出版社，1995.6
ISBN 7-81012-572-9

I. W... I. 程... III. 数据交换-操作系统, Windows IV. TP312

中国版本图书馆 CIP 数据核字 (95) 第 04668 号

内 容 简 介

本书介绍利用 Turbo C++ for Windows 和 Borland C++ 进行 Windows 动态数据交换程序设计的技术和参考信息。首先介绍一般 Windows 程序设计技术和 Windows 动态数据交换概念及如何进行 DDEML 会话管理、DDEML 事务管理和设计 DDEML 监视器应用程序；接着讨论对象链接与嵌入概念，介绍如何设计 OLE 服务器应用程序、OLE 客户应用程序和对象处理程序；然后阐述剪贴板的使用，如何编写动态链接函数库；最后给了 OLE 和 DDEML 使用动态数据交换、对象链接与嵌入和剪贴板函数和消息的参考信息。适合程序员及大学高年级学生使用。

● 书 名: Windows 动态数据交换程序设计
——用 Borland C++ 和 Turbo C++ for Windows
Windows DONGTAI SHUJU JIAOHUAN CHENGXU SHEJI
——YONG Borland C++ HE Turbo C++ for Windows

- 编 著 者: 程铁阜 程文斌等 编
- 责任编辑: 王小青
- 出 版 者: 北京航空航天大学出版社 发行科电话: 2015720
- 地 址: 北京市海淀区学院路 37 号(100083)
- 印 刷 者: 朝阳科普印刷厂印刷
- 发 行: 新华书店总店科技发行所
- 经 售: 全国各地书店
- 开 本: 787×1092 1/16
- 印 张: 50
- 字 数: 1278 千字
- 印 数: 5000 册
- 版 次: 1995 年 6 月第 1 版
- 印 次: 1995 年 6 月第 1 次印刷
- 书 号: ISBN 7-81012-572-9/TP·160
- 定 价: 61.00 元



动态数据交换(DDE)是进程间通信的方式。进程间通信(IPC)包括进程之间的数据传递和事件之间的同步;DDE用共享存储器在应用程序之间交换数据,用协议来同步数据的传递;Windows还使用剪贴板、动态链接库(DLL)和对象链接与嵌入(OLE)作为进程间通信的方法。

OLE是使用DDE协议的协议;从构成来看,OLE协议是DDE Execute命令集。

剪贴板包括用全局内存在应用程序间传递数据的函数。全局内存不同于剪贴板程序,剪贴板程序是显示全局内存中数据的应用程序。典型的,剪贴板以一遍为基础在应用程序中传递数据。

DLL是包含代码、数据和多个Windows程序可以存取的Windows源程序的模块。多个程序可以共享一个DLL实例,然而同一时间内只能有一个DLL实例存在。DLL和应用程序使用全局内存的共享数据。为了完成这一功能,应用程序调用DLL的函数来分配全局内存块,然后应用程序在DLL终止调用函数时,全局内存块被释放。全局内存块的生命期和应用程序调用DLL的生命期一样长,因此,两个应用程序不能存取同一个DLL分配的全局内存块。

本书讨论用Borland C++和Turbo C++ for Windows设计Windows动态数据交换程序。从Borland C++ 4.0开始,Borland C++的集成环境Borland IDE运行于Windows,是一个很好用的Windows应用程序开发环境;要特别指出的是,在新的IDE中可在Windows中直接调试程序。

在Borland C++ 3.1~4.0,配备了Turbo Debugger for Windows用来调试用Borland C++设计的Windows应用程序,增强了资源包括图标、光标、位图、对话框、菜单、热键和字体等的设计,用来设计Windows应用程序的界面。Borland C++在OWL中封装了Windows API,包括函数、数据结构和消息等,4.0版支持在Windows 3.1和Windows NT 3.5操作系统下开发应用程序,方便C/C++读者开发Windows应用程序。

本书适合于用Borland C++以及各种演化版本编写Windows动态数据交换程序。

本书中的例子针对国内程序员的需要,全部采用中文的菜单、对话框、提示界面等,每个程序都有很好的注释和解说,方便理解和学习。

目 录

简 介	1
0.1 概述	1
0.2 所需的编程工具	2

第一部分 一般 Windows 编程

第一章 启动 Borland C++	5
1.1 运行安装程序	5
1.2 集成开发环境(IDE)的使用	10
1.2.1 启动 IDE	10
1.2.2 使用 IDE 的菜单和窗口	10
1.3 配置 Borland C++	14
1.4 编写第一个 C 语言程序	16
1.4.1 理解 C 程序的结构	17
1.4.2 使用编辑器编写程序	21
1.4.3 编译和运行程序	23
1.4.4 程序是如何工作的	23
1.5 介绍库函数	24
1.5.1 使用一些基本的输入函数	25
1.5.2 使用一些基本的输出函数	27
1.5.3 使用一些格式转换函数	29
1.6 总结	31
第二章 Windows 程序设计	32
2.1 关于 Windows 编程的一些说法	32
2.2 为什么要为 Windows 编程	33
2.3 开发者的有利之处	33
2.4 对 Windows 编程环境的理解	34
2.4.1 Windows 是多任务环境	34
2.4.2 Windows 是面向对象的环境	35
2.5 Windows 资源	42
2.5.1 Windows 资源的类型	42

2.5.2 资源文件类型.....	44
2.6 用资源工作.....	44
2.7 fcwin.c 例子程序所需的资源	44
2.8 定制 FCWIN 资源	49
2.9 总结.....	56
第三章 Windows 编程工具	58
3.1 WinSpector 应用	58
3.1.1 启动 WinSpector 应用	58
3.1.2 Preferences 对话框.....	58
3.1.3 Directory 选项	59
3.1.4 Viewer 选项	59
3.1.5 Append New Reports 选项	59
3.1.6 Overwrite Previous Reports 选项	60
3.1.7 System Information 选项	60
3.1.8 Summary To AUX 选项	60
3.1.9 Post - Mortem Dump 选项	60
3.1.10 Stack Frame Data 选项	60
3.1.11 User Comments 选项	60
3.1.12 解释日志文件	60
3.2 WinSight 应用	73
3.2.1 启动.....	74
3.2.2 选择察看.....	74
3.2.3 使用窗口树.....	75
3.2.4 窗口树显示格式.....	75
3.2.5 扩展及关闭窗口树.....	75
3.2.6 定位一个窗口.....	75
3.2.7 记录窗口.....	76
3.2.8 窗口类的工作.....	77
3.2.9 选择被跟踪消息.....	77
3.3 Turbo Debugger for Windows	79
3.3.1 调试准备.....	79
3.3.2 使用 Turbo Debugger	82
3.3.3 使用高级调试特征.....	88
3.4 总结.....	89
第四章 设计 Windows 应用程序的方法	91
4.1 建立 Windows 应用程序环境	91
4.1.1 建立你自己的 Windows Stub 可执行文件	91

4.1.2	理解 Windows 目录的使用	93
4.1.3	示例设计 fcwin.c	94
4.2	为 Windows 应用程序建立源文件	123
4.2.1	熟悉 Windows 编程环境	124
4.2.2	建立一个模块定义文件	124
4.2.3	设计程序的头文件	125
4.2.4	为 FCWIN 构造工程文件	127
4.3	总结	128
第五章	编写 Windows 应用程序	129
5.1	设计 Windows 接口	129
5.1.1	登录窗口类	129
5.1.2	建立主消息循环	134
5.1.3	写 WndProc() 函数	135
5.1.4	为对话框设置回调函数	136
5.1.5	建立对话框函数	137
5.1.6	控制对话框	138
5.1.7	使用 MessageBox() 函数来建立弹出帮助和错误信息帮助	139
5.1.8	假脱机硬拷贝到 Windows 打印管理器	141
5.2	使用动态链接库(DLL)	148
5.2.1	理解 DLLs	149
5.2.2	编写一个 DLL 应用程序	150
5.3	总结	154

第二部分 动态数据交换

第六章	动态数据交换概念	157
6.1	DDE 协议	157
6.1.1	会话的同步	158
6.1.2	应用程序名、标题名和项目名	158
6.1.3	会话初始化	158
6.1.4	会话中的交换	159
6.1.5	会话终止	160
6.1.6	DDE 消息	160
6.2	动态数据交换管理库——DDEML	174
6.2.1	回调函数	174
6.2.2	DDEML 应用程序	175
6.2.3	服务名、标题名和项目名	175

6.2.4	DDEML 事务	176
6.3	总结	178
第七章	DDEML 会话管理	179
7.1	DDEML 的初始化过程	179
7.1.1	初始化应用程序——DdeInitialize()	179
7.1.2	登记服务名	182
7.1.3	错误捕获——DdeGetLastError()	183
7.1.4	释放 DDEML 资源——DdeUninitialize()	185
7.2	会话管理	185
7.2.1	运用标题名	185
7.2.2	建立一个会话——DdeConnect()	186
7.2.3	建立多个会话——DdeConnectList()和 DdeQueryNextServer	188
7.2.4	获取会话信息——DdeQueryConvInfo()	189
7.2.5	结束一个会话——DdeDisconnect()和 DdeDisconnectList()	190
7.2.6	重建一个终止了的会话——DdeReconnect	190
7.3	DDEML 客户机和服务器应用程序举例	190
7.3.1	DDEML 服务器应用程序举例	191
7.3.2	DDEML 客户机应用程序举例	206
7.4	总结	225
第八章	DDEML 事务管理	226
8.1	DDEML 字符串管理	226
8.1.1	创建字符串句柄	227
8.1.2	递增和递减参考计数	227
8.1.3	获得和比较字符串	227
8.2	DDEML 数据管理	229
8.2.1	创建全局内存对象	229
8.2.2	存取全局内存对象	229
8.2.3	释放全局内存对象	230
8.3	事务管理	231
8.3.1	DDEML 事务——DdeClientTransaction()	232
8.3.2	事务控制	239
8.4	程序例子——TICKER 和 QUOTE	241
8.4.1	应用程序设计	242
8.4.2	一个 DDEML 客户应用程序——TICKER	242
8.4.3	一个 DDEML 服务器应用程序——QUOTE	271
8.5	总结	297

第九章 DDEML 监视器应用程序	298
9.1 监视器应用程序的初始化	298
9.2 监视程序结构	299
9.3 DDEMON —— DDE 监视程序举例	302
9.4 总结	355

第三部分 对象链接与嵌入

第十章 对象链接与嵌入概念	359
10.1 对象和文档	359
10.2 链接与嵌入	360
10.3 打包	360
10.4 登记数据库	360
10.5 剪贴板	361
10.6 OLE 应用程序	362
10.6.1 客户应用程序	363
10.6.2 服务器应用程序	367
10.7 总结	363
第十一章 OLE 服务器应用程序	370
11.1 POLY.EXE: 一个 OLE 服务器应用程序	370
11.1.1 头文件编译举例	371
11.1.2 WinMain: 应用程序初始化和终止	378
11.1.3 主窗口处理模块	388
11.1.4 多边形窗口处理	397
11.1.5 服务器分类	400
11.1.6 文档分类	404
11.1.7 文档管理	408
11.1.8 对象方法	413
11.1.9 对象管理	420
11.1.10 文件输入输出	432
11.1.11 菜单和杂类处理	443
11.1.12 对话框过程	449
11.1.13 支持模块	451
11.2 总结	459

第十二章 OLE 客户应用程序	460
12.1 OLECLNT.EXE: 一个 OLE 客户应用程序	460
12.1.1 程序例子头文件	461
12.1.2 WinMain: 应用程序初始化和终止	468
12.1.3 主窗口处理模块	474
12.1.4 拖 放	478
12.1.5 对象窗口处理	480
12.1.6 客户回调和流方法	485
12.1.7 文档管理	488
12.1.8 对象管理	507
12.1.9 会话和对象插入	523
12.1.10 剪贴板处理	532
12.1.11 链接管理	535
12.1.12 登记数据库存取	548
12.1.13 菜单处理	553
12.1.14 支持模块	557
12.2 总结	570
第十三章 对象处理程序	571
13.1 动态链接库的概念	571
13.1.1 输入库	571
13.1.2 动态链接库的组成成员	572
13.1.3 动态链接库的优点	573
13.2 POLYHAND.DLL: 动态链接库对象句柄	573
13.2.1 对象句柄登记	573
13.2.2 对象句柄的输入和输出函数	574
13.2.3 对象句柄函数管理	574
13.2.4 对象创建函数与 OLE API 的不同点	576
13.2.5 实例程序	577
13.3 总结	596
第四部分 剪贴板和动态链接库	
第十四章 剪贴板的使用方法	599
14.1 剪贴板使用的数据格式	600
14.2 文本数据与剪贴板	600
14.2.1 剪切或复制文本数据到剪贴板	601

14.2.2	粘贴文本数据到应用程序	602
14.3	图形数据与剪贴板	611
14.4	虚拟文件图与剪贴板	619
14.4.1	剪切或复制虚拟文件图到剪贴板	620
14.4.2	粘贴虚拟文件图到应用程序	621
14.5	剪贴板的高层使用技巧	629
14.5.1	数据格式多样化	629
14.5.2	延迟提供数据给剪贴板的时间	630
14.5.3	使用自定义的剪贴板格式	631
14.5.4	剪贴板内容显示程序	633
第十五章	动态链接函数库	641
15.1	什么是动态链接函数库	642
15.1.1	静态链接与动态链接	642
15.1.2	动态链接函数库与一般应用程序间的差异	644
15.1.3	目标函数库与导入函数库	645
15.1.4	Windows 如何找到动态链接函数库	645
15.1.5	使用动态链接函数库的优点	645
15.1.6	查看动态链接函数库的内容	646
15.2	自行设计动态链接函数库	647
15.2.1	动态链接函数库的入口	648
15.2.2	LibMain()函数	649
15.2.3	WEP()函数	650
15.2.4	动态链接函数库的模块定义文件	651
15.3	动态链接函数库的调用方式	657
15.3.1	输入动态链接函数库中的函数的方法	658
15.3.2	远程函数调用	659
15.3.3	DS! =SS 的衍生问题	660
15.3.4	动态链接函数库的重要限制	662
15.4	动态链接的高层技巧	675
15.4.1	回调函数的运用	675
15.4.2	应用程序指定链接的方法	677
15.4.3	善用输入函数库	677
15.5	动态链接函数库的使用扩展	693
15.5.1	无需事先输入函数的动态链接方式	693
15.5.2	仅含资源的动态链接函数库	694
15.5.3	内存对象的归属权	694
15.6	其他需要运用动态链接函数库的时机	700
15.6.1	挂接与动态链接函数库	700

15.6.2 驱动程序与动态链接函数库.....	701
15.6.3 用户自定义的子控制.....	702

第五部分 附 录

附录 A OLE 和 DDEML 使用	717
附录 B 动态数据交换函数和消息	718
附录 C 剪贴板函数和消息	742
附录 D 对象链接与嵌入函数和消息	755
附录 E 安装、启动、退出中文 Windows	781



Microsoft Windows 是广泛应用的应用程序开发平台之一。自从 1990 年 5 月发行 Windows 3.0 以来,一些软件开发的公司已提供了上千种的 Windows 应用程序。与此同时,Microsoft 公司已通过使 Windows 更稳定和扩展的 Windows 编程 API 来增强 Windows 平台。

Windows 3.1 在 Windows 系列产品中对数据处理表现突出。本书讨论了 Windows API 扩展的四个方面:动态数据交换管理库(Dynamic Data Exchange Management Library (DDEML))、对象链接与嵌入(Object Linking and Embedding(OLE))、剪贴板(Clipboard)和动态链接库(Dynamic Link Library)。

0.1 概述

本书是为想学习 DDEML 和 OLE API 的 Windows 程序员而编写的。为了能让初学 Windows 编程也能尝试进行 DDEML 和 OLE 程序设计,在本书的第一部分先介绍了一般的 Windows 程序设计技术,第二部分以后的例子集中讨论了 DDEML 和 OLE 编程技术。

该书分为五部分,第一部分包含五章,介绍了一般的 Windows 程序设计技术;第二部分和第三部分各包含四章,第四部分包含二章;第五部分包含五个附录,提供编程的参考信息。

第一部分介绍 Windows 程序设计技术。第一章介绍如何启动 Borland C++。第二章讨论 Windows 程序设计的概念。第三章介绍 Windows 的编程工具。第四章阐述设计 Windows 应用程序的方法。第五章具体介绍如何编写 Windows 应用程序。

第二部分讲 DDEML API。第六章为 DDE 和 OLE 的概念,复习基于消息的 DDE 和 DDE 协议。第七章到第九章通过使用应用程序例子来详述 DDEML API。第七章以使用 DDEML 服务器应用程序的例子来作为讨论的焦点。第八章通过给出的 DDEML 客户应用程序来论述 DDEML 程序的客户/服务器循环。第九章讲述扩展 DDEML API 作为监视应用程序。

第三部分讨论 OLE。第十章讲概念,介绍了开发 OLE 应用程序的基本结构和过程。第十一章列举了 OLE 服务器应用程序的例子。第十二章是 OLE 客户应用程序的例子。第十三章介绍开发 OLE 对象的处理程序。

第四部分介绍动态数据交换的另外二种方法:剪贴板和动态链接库。第十四章介绍剪贴板的程序设计。第十五章讨论了动态链接库的设计。

第五部分是附录。附录 A 给出了 OLE 和 DDEML 的使用。附录 B、附录 C 和附录 D 列出 Windows API 涉及 DDE、剪贴板和 OLE 中的函数和消息。附录 E 介绍中文 Microsoft Windows 3.1 的安装、启动和退出,以便读者设计中文的应用程序界面。

本书提供程序例子以全面展示 DDEML 和 OLE 的应用程序。每章的最后总有连续好几页的大程序,这种情况对于 OLE 应用程序更是常见。从读者的观点出发,更愿意阅读有众多注释的代码;但从实用的观点出发,不能篇幅太大,从而只能省略大量注释;因此,本书的代码明显地比注释多。注释和程序例子的结合将提供给用户编写 OLE 和 DDEML 应用程序的细节要

求。

该书中的程序例子使用中模式,这些程序是读者实际应用的起点。在不清楚实际的应用程序将有多大时,用中模式更合适,因为用中模式编写大程序比用小模式编写大程序更容易。

0.2 所需的编程工具

用户需要包括有能开发 Windows 3.1 应用程序的库和头文件的 C 或 C++ 编译器,或 Microsoft Windows 3.1 Software Development Kit (SDK)。该书可使用下述开发工具以编写程序例子:

Windows 3.1

Turbo C++ for Windows, Borland C++ 3.1~4. X

Microsoft Windows 3.1 Software Development Kit

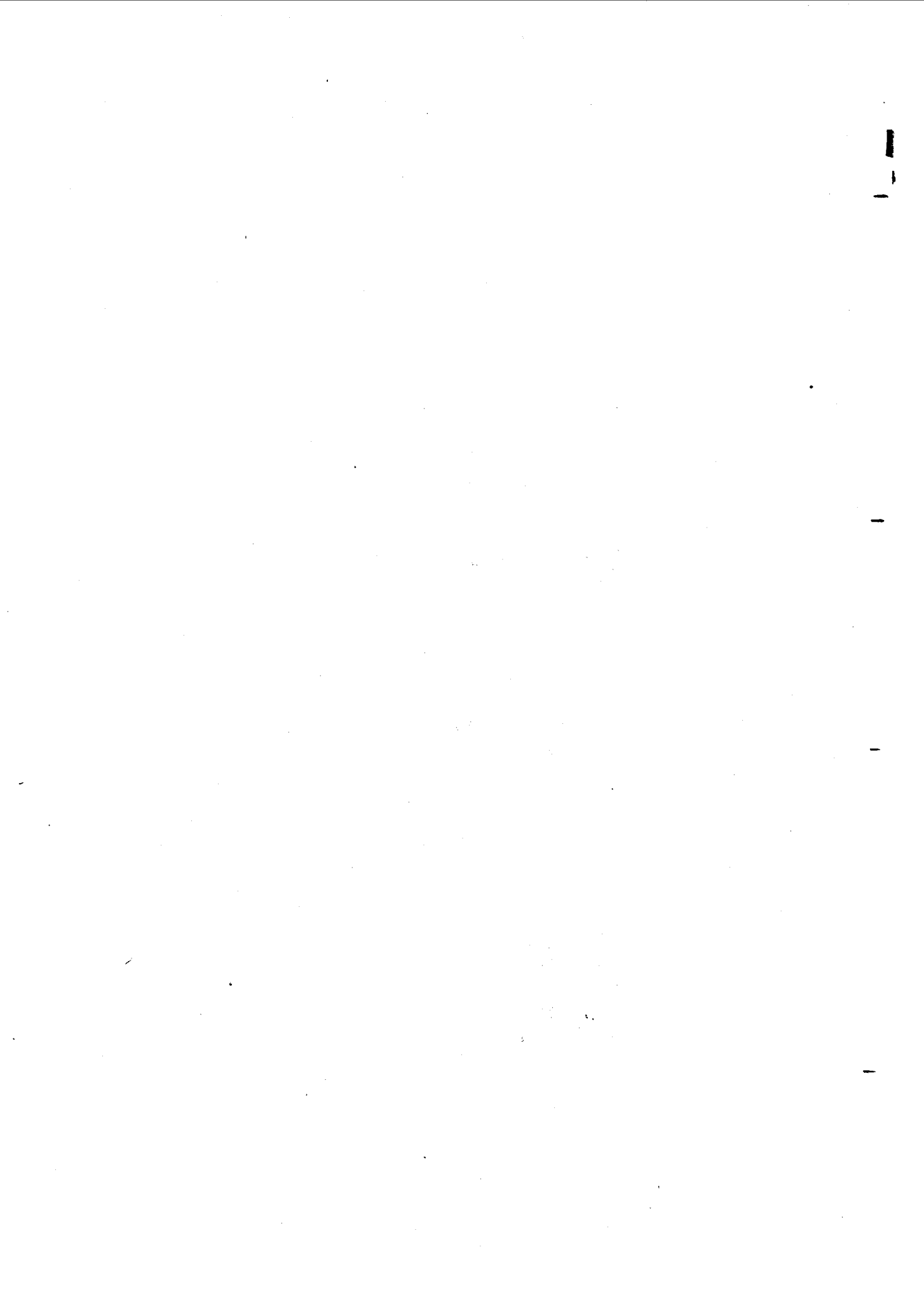
开发 Windows 3.1 应用程序所需的最小硬件配置和运行 Windows 3.1 相同,即具有 640K 常规内存的 80286(或更高级的)PC,插有 256K 扩展内存,至少有 6M 剩余空间的硬盘。最好具有 80386 和 80486 的机器,4M 内存,200M 硬盘,VGA 显示器和鼠标。

使用单色监视器作为调试程序,这是在 Windows 3.1 用 Turbo Debugger for Windows 调试程序使用双屏幕调试所必需的。

本书中的程序例子用 Turbo C++ for Windows 和 Borland C++ 3.1 与 4.0 编译过,如果使用上述编译器进行编译,则不用对源程序进行修改。

第一部分

一般 Windows 编程



第一章 启动 Borland C++

本章帮助用户学会如何在机器上安装 Borland C++，介绍如何使用 Borland C++ 的特色。首先学会如何安装 Borland C++ 和根据不同的需要对 Borland C++ 进行配置。其次探索新的集成开发环境 (IDE) 中的一些最有用的特色。在熟悉集成开发环境之后，再学习如何编制简单的 C 程序。

1.1 运行安装程序

Borland 公司有一个很优秀的安装实用程序，该程序简化了安装大的软件包时可能有的繁琐步骤，所以安装 Borland C++ 是一件容易的事。用户所要做的事基本上就是告诉 INSTALL (安装) 程序想要哪一个选择项，然后，当系统提示时，装入相应的每一张软盘。其余的工作由程序替用户完成。

另外，Borland C++ 有光盘版本，用光盘安装则更加简单。在整个安装过程中，你不必再插入每一张软盘 (因为一张光盘的容量可以容得下所有要安装的内容)。用户所要做的事无外乎是确保光盘已插在驱动器里。本章的屏幕解释就是针对光盘安装程序的。软盘版本的安装过程与此类似。

关于 Borland C++ 安装问题

在安装 Borland C++ 之前，需要检查硬盘上可用空间的大小。Borland C++ 是一个大型软件包：当所有的选择项都安装时，它几乎要用掉 128M 空间。安装过程开始时，把光盘插入光盘驱动器 (或把标有 INSTALLATION DISK 的软盘插入软盘驱动器)，然后运用文件管理程序 (File Manager) 或其它类似的 Shell 软件，在装有安装盘的驱动器中找到 INSTALL. EXT 文件，双击此文件 (用鼠标) 即可。你也可以在程序管理程序 (Program Manager) 的文件 (File) 菜单中选择运行 (Run) 选项，输入：

```
n:install
```

其中 n: 是指插有 INSTALLATION 盘的那个软盘驱动器号或者是装有 Borland C++ 光盘的光盘驱动器号。

这时见到的第一个对话框如图 1.1 所示，图中显示出“欢迎使用安装程序”字样，并且告诉用户全部安装要用多大空间以及现在硬盘还有多少可用空间。对话框显示的内容还有：Borland C++ 4.0 子目录及 Windows 子目录的路径以及 (安装) 源路径。在此对话框的下半部分，有四个选择框。第一个框告诉用户可以只安装配置文件 (此项只适用于光盘安装)，运行时再在光盘驱动器中插入光盘，以便节省可观的硬盘空间。第二个框可让用户选择是否安装 Win32S；如果你打算生成 32-位程序，你必须选择该项。另外二个选择框可以让用户选择是否创建一个 Borland C++ 的群组 and 选用局域网 Windows 等两项配置。对话框的右边，还有四个按钮：Install, Exit, Reset Default 和 Help。第一个按钮 (Install) 帮助用户安装 Borland C++。第二个按钮 (Exit) 允许用户退出安装对话框。当用户对已经修改的选择项不满意时，第三个按钮 (Reset