

Microsoft Windows 3.0 手册

Windows 3.0

软件开发指南(一)

应用程序设计

熊桂喜 钟 宁 编译
熊桂喜 魏 彬 审校

清华大学出版社

1.52

X/1-1

Microsoft Windows 3.0 手册

Windows 3.0 软件开发指南 (一) ——应用程序设计

熊桂喜 钟 宁 编译
熊桂喜 魏 彬 审校

清华大学出版社

内 容 简 介

本书介绍了如何在 Windows 3.0 的环境下进行 Windows 应用程序的设计。与一般程序不同, Windows 应用程序有许多新的特点。它使用窗口、对话框、菜单和滚动条,支持统一的图形界面、多任务、资源共享等特性。因此,除了一般的编程方法外,进行 Windows 应用程序设计还需掌握许多 Windows 特有的编程技巧,使用好 Windows 提供的资源和工具。

全书分二十二章,每章一个专题,详细地介绍了 Windows 3.0 应用程序设计的环境、程序特点、各种资源的使用方法等。

本书是从事 Windows 应用程序设计的必备参考书,也可作为一般编程人员了解应用程序软件动向和掌握窗口技术的参考。

(京)新登字 158 号

Microsoft Windows 3.0 手册
Windows 3.0 软件开发指南 (一)
——应用程序设计

熊桂喜 钟 宁 编译
熊桂喜 魏 彬 审校

☆

清华大学出版社出版
北京 清华园
清华大学印刷厂印刷
新华书店总店科技发行所发行

☆

开本: 787×1092 1/16 印张: 23.25 字数: 580 千字

1991 年 5 月第 1 版 1992 年 3 月第 2 次印刷

印数: 10001—25000

ISBN 7-302-00857-4/TP·309

定价: 12.00 元

前 言

Windows 3.0 的推出,使得操作计算机(特别是 PC 机)的方法和软件开发过程发生了革命性的变化。

Windows 3.0 是一个图形窗口操作环境软件,它使得 PC 机的面目焕然一新。它提供了一种不同于以往的命令式操作手段,计算机的操作是通过诸如“对话”、“肖像”、“菜单”等图形画面和符号的操作来完成的。与 DOS 操作系统相比,Windows 3.0 是一个功能更强的图形操作环境,具有如下的功能和特点:

全新的、漂亮的图形操作界面,使 PC 机易于掌握,易于使用;

多任务运行,各任务之间既易于转换,又可方便地交换信息;

突破 DOS 内存 640KB 限制,提供了实模式、标准模式、386 增强模式等操作模式,提供了虚存管理能力,内存最大可达 16M;

提供了程序管理器、文件管理器、打印管理器、控制面板等功能强大、操作方便的管理工具,可完成任务、文件、输出设备等的并行管理工作;

提供了多个方便的功能强大的应用程序:字处理器(Write),画图软件(Paintbrush),终端通讯软件(Terminal)等;

提供了一套完整的桌面办公用具:计算器(Calculator)、日历(Calendar)、卡片文件(Cardfile)、时钟(Clock)、便笺(Notepad)、记录器(Recorder)等。

Windows 3.0 应用软件开发工具包 SDK,为在 Windows 3.0 环境下开发出具有窗口特点和功能的应用软件提供了各类工具、资源、函数库和数据结构。在 Windows 3.0 下使用 SDK 进行应用程序开发与以往在 DOS 下进行的常规软件开发过程不一样,它可使用 SDK 提供的各种编辑、管理、编译、连接、调试、观察、帮助等工具,使软件开发工作不仅可以在源程序级进行调试并使用各类开发工具,而且可使用 Windows 3.0 提供的菜单、对话框、肖像、控制、Help 等资源,并可自己进行设计,最终开发出的软件也具有 Windows 风格,方便好用、功能强大、运行效率高。不仅如此,在 Windows 下开发应用程序可以充分地利用机器资源,例如,调试过程可在 80386 保护方式下的源程序级以窗口方式进行。

因此,使用好 Windows 3.0 及其开发软件包 SDK,对使用好 PC 机,提高其使用效率,对缩短软件开发周期,减少软件出错,开发出高质量的、用户界面良好的应用软件,都有十分重要的意义。

我们在清华大学出版社的大力支持下,编译了这套 Windows 3.0 手册,介绍 Windows 3.0 的使用以及如何 Windows 3.0 下使用软件开发工具包 SDK 进行应用程序设计。全套手册包含如下内容的四本书:

《Windows 3.0 用户指南》:详细介绍如何安装、设置、使用 Windows 3.0;在 Windows 3.0 下使用各类管理工具和实用软件;在 Windows 3.0 下运行各类应用程序。

《Windows 3.0 软件开发指南(一)——应用程序设计》:介绍了如何使用 Windows 3.0

应用程序设计接口(API)进行应用程序设计。内容包括如何使用 Windows 函数、资源、设备、控制、消息和数据结构,设计出具有 Windows 风格的、高性能的、可靠的应用程序。特别是涉及了一些高级程序设计内容和多语言、内存管理和动态数据交换、打印以及其它高级设置等内容。

《Windows 3.0 软件开发指南(二)——开发环境及工具》:介绍在 Windows 3.0 下使用 SDK 进行应用程序设计时的开发环境及可用的程序及资源编译工具;资源编辑及管理工具;程序调试、优化,信息、消息管理及监视工具以及 Help 工具。

《Windows 3.0 软件开发指南(三)——库函数及数据结构》,详细介绍了在 Windows 应用程序开发时使用到的程序接口(API)的众多函数、消息和数据结构的完整内容。

全套手册深入细致地介绍了 Windows 3.0 的全部使用特点及技巧,以及在 Windows 3.0 下开发应用软件的方法、工具,需查阅的全部信息。内容详尽,图文并茂。

在这套工具书中,“用户指南”一书适用于各类 PC 机的操作、开发、系统维护人员,通过本书,可以更好地、更方便地、更有效地使用 PC 机及其软件。“软件开发指南”的三本书是在 PC 机上通过 Windows 3.0 进行应用程序开发的程序员必备的参考资料,也是其它软件设计人员了解新的软件设计思想和方法的技术参考书。

全套丛书由熊桂喜、赵海等从事 Windows 应用软件开发人员编译而成。由熊璋、田子钧、王先之等从事应用软件研究的人员主审。刘源对本书的出版也给予了热情支持。本书在编译过程中,还得到北航计算机系其它众多老师的热情支持和帮助。在此一并致谢。

由于编译者水平有限,加上时间仓促,书中难免有缺点和错误,欢迎广大读者给予批评和指正。

编译者

1991年3月于北京航空航天大学计算机系

简介

本书介绍了在 Windows 3.0 环境下如何进行应用程序的设计。与一般的应用程序不同,在 Windows 3.0 环境下运行的应用程序具有许多 Windows 的独特性能,如多道程序同时运行,共享内存资源,使用漂亮统一的图形界面,使用窗口、菜单、控制、对话框、鼠标等操作手段以及与设备无关等特性。这些特点赋予应用程序以全新的面貌,方便了用户,但也使编制应用程序的工作更复杂。软件开发人员需要学习和掌握新的技巧,使用好资源,掌握好 Windows 工具,熟悉 Windows 环境。

一、预备知识

在使用本书之前,读者需要具备下列预备知识:

- 对 Windows 有一定使用经验,对 Windows 用户界面有较好的理解。

在开始开发 Windows 应用程序之前,必须在计算机上安装 Windows 3.0 并学会使用它,确信已掌握了 Windows 应用程序的各个部件的名称、用途与操作方法(如窗口、对话框、菜单、控制和滚动条等),因为将要开发的 Windows 应用程序也将具备这些特征。

- 对 Windows 用户界面风格有较好的理解。

Windows 的目标之一是为所有应用程序提供一个通用的用户界面,这有助于减少用户为学习 Windows 应用程序的界面所花费的精力,也有助于程序开发人员在设计用户界面时有一个正确的选择。为了达到这个目标,应用程序的用户界面必须遵从 Windows 应用程序界面的统一标准。

- 编写 C 语言程序和使用标准 C 语言函数的经验。

C 语言是推荐的 Windows 应用程序的开发语言,Windows 的许多编程特性都充分考虑了 C 语言的特点。(Windows 应用程序也可以用 Pascal 语言和汇编语言来实现,不过,与 C 语言相比,使用这些语言来进行 Windows 应用程序设计将会给编程工作带来一些困难。)

二、本书内容的组织

本书旨在帮助有经验的 C 程序员采用 Windows 3.0 的应用程序界面开发应用程序。书中解释了如何使用 Windows 函数、消息和数据结构来实现大多数应用程序所应具有的功能,并用丰富的应用程序示例加以说明,这些示例均可在 Windows 3.0 中编译和运行。

本书由三个部分组成。

第一部分:Windows 应用程序设计,先简单介绍 Windows 的编程环境,接着深入考察一个 Windows 应用程序示例。本部分包括两章:

第一章,Windows 环境概貌。对 Windows 环境作扼要的介绍,描述 Windows 程序设计模型及开发过程。

第二章,Windows 应用程序 Generic。介绍创建名为 Generic 的应用程序的过程。在后面的章节中将要使用此应用程序做为分析的例子。

第二部分:Windows 应用程序设计。介绍 Windows 程序设计的基本任务,如建立菜单,打印和使用剪接板等。各章分别涉及一个专题,并举一个例子加以说明。本部分包括以下各章:

第三章,窗口输入。介绍图形设备接口(GDI)以及如何使用 GDI 工具建立自己的输出。

第四章,键盘输入与鼠标输入。介绍处理鼠标输入与键盘输入的过程。

第五章,肖像。说明如何为应用程序建立和显示肖像。

第六章,光标、鼠标和键盘。介绍光标、鼠标和键盘的用途以及如何在程序中运用它们。

第七章,菜单。介绍如何建立菜单和处理菜单输入。

第八章,控制。介绍如何创建和使用控制,如按钮和列表框等。

第九章,对话框。介绍如何创建和使用对话框以及在对话框中使用控制的方法。

第十章,文件输入与输出。介绍函数 OpenFile 的用法以及磁盘操作的要领。

第十一章,点位图。介绍如何创建并显示点位图。

第十二章,打印。介绍如何使用打印机。

第十三章,剪接板。介绍在应用程序中使用剪接板的方法。

第三部分:高级程序设计专题。讲述程序设计的几个高级专题,如内存管理和动态数据交换等。每章论述一个专题。本部分包括以下各章:

第十四章,C 语言与汇编语言。给出用 C 语言和汇编语言编写 Windows 应用程序的指导。

第十五章,内存管理。介绍如何分配全局内存与局部内存。

第十六章,关于内存管理的进一步说明。深入细致地分析应用程序有效使用内存的方法,还介绍 Windows 如何管理在不同配置下的内存。

第十七章,打印设置。介绍如何对打印机进行设置以满足应用程序的要求。

第十八章,字库。说明创建和装入字库,以及使用函数 TextOut 输出各种字体的方法。

第十九章,彩色调色板。介绍 Windows 如何利用彩色调色板,使应用程序最有效地使用颜色。

第二十章,动态连接库。介绍如何创建和使用 Windows 的动态连接库。

第二十一章,多文档界面。介绍应用程序如何创建和使用 Windows 的多文档用户界面(MDI),以便让用户能同时对多个文档进行操作。

第二十二章,动态数据交换。介绍如何利用基于消息的动态数据交换协议来实现应用程序之间的动态数据交换。

三、开发应用程序所需工具简介

大多数 Windows 3.0 应用程序的建立需要用到下列工具:

- Microsoft C 优化编译器:CL
- Microsoft 的可执行代码段连接器:LINK
- Microsoft Windows 资源编译器:RC
- Microsoft Windows SDKPaint;SDKPAINT

- Microsoft Windows 对话编辑器: DIALOG

为建立 Windows 的函数库和字库资源文件, 需要用到下列工具:

- Microsoft 宏汇编程序: MASM
- Microsoft Windows 字库编辑器: FONTEDIT

在建立和调试 Windows 应用程序时, 可以利用下列工具:

- Microsoft 程序维护工具: MAKE
- Microsoft 符号调试器: SYMDEB
- Microsoft CodeView 的 Windows 版本: CVW
- Microsoft Windows 剖析器: PROFILER
- Microsoft Windows 对换分析器: SWAP
- Microsoft Windows 堆使用跟踪器: HEAPWALK
- Microsoft Windows 消息监视器: SPY

Microsoft Windows 软件开发工具包 SDK 3.0 提供了以上大部分工具, 但未提供 C 编译器、连接器、宏汇编程序和程序维护工具, 在《Windows 3.0 软件开发指南(二)——开发环境及工具》一书中对这些内容有详细描述。

关于 Windows 3.0 对软件和硬件的需求, 参见《Windows 3.0 用户指南》。

四、示例应用程序的用法

本书中的示例应用程序均用 C 语言编写而成, 其用户界面风格符合 Microsoft Windows 应用程序所采用的规范。

SDK 示例源代码磁盘提供了所有示例应用程序的源代码文件。我们建议在阅读本书时参考这些源代码文件。为了阅读方便, 本书列出了部分源代码文件, 它们可以作为 Windows 应用程序设计的基础。

五、书中符号约定

在本书中, 除非特别注明之外, 术语“DOS”系指 MS-DOS 和 PC-DOS。

本书使用的符号约定说明如下:

- () 在语句中, 圆括号括住一个或多个传给函数的参数。
- ... 独立的省略号行表示程序中省略了一段程序。
- ... 行内省略号表示某项后可出现多个形式相同的项。例如, 下例中的省略号表示可为 g 命令指定多个断点地址:
g[=
<开始地址>][<中断地址>]...
- < > 尖括号里的内容为非终结符。
- [] 方括号括住命令行或文法语句中的任选项或参数。在下例中, <任选项>和<可执行文件>为 RC 命令的任选参数: RC[<任选项>]<文件名>[<可执行文件>]
- | 竖杠表示选择“|”左面或右面其中的一个项。下述命令行语法说明了竖杠的用法:
DB[<地址>|<范围>]
竖杠表示在 DB 命令之后, 可指定一个地址或一个范围。
- “ ” 引号用来分开在正文中定义的项。
- { } 大括号表示一个必须括起的项。

目 录

前 言

第一章 Windows 环境概述	1
1.1 Microsoft Windows 与 DOS 的比较	1
1.1.1 用户界面	1
1.1.2 队列式输入	2
1.1.3 与设备无关的图形	2
1.1.4 多任务	3
1.2 Windows 程序设计模式	3
1.2.1 窗口	3
1.2.2 菜单	4
1.2.3 对话框	4
1.2.4 消息循环	5
1.3 Windows 函数库	6
1.4 建立 Windows 应用程序	7
1.5 软件开发工具	8
1.5.1 C 编译器	8
1.5.2 连接器	8
1.5.3 SDK 资源编辑器	9
1.5.4 资源编译器	9
1.5.5 调试和优化工具	10
1.5.6 程序维护工具	10
1.6 编写 Windows 应用程序的要领	11
1.7 小结	12

第二章 Windows 应用程序 Generic

2.1 Generic 应用程序	13
2.2 Windows 应用程序	13
2.3 WinMain 函数	14
2.3.1 Windows 的数据类型和 数据结构	14
2.3.2 句柄	15
2.3.3 实例	15
2.3.4 注册窗口类	16
2.3.5 建立窗口	18
2.3.6 显示和修改窗口	19
2.3.7 建立消息循环	20
2.3.8 提供控制	21

2.3.9 终止应用程序	21
2.3.10 初始化函数	22
2.3.11 应用程序的命令行参数	23
2.4 窗口函数	23
2.5 建立 About 对话框	24
2.5.1 建立对话框模板	25
2.5.2 建立包含文件	26
2.5.3 建立对话函数	26
2.5.4 用 About 命令定义菜单	27
2.5.5 处理 WM_COMMAND 消息	28
2.6 建立模块定义文件	29
2.7 装配 Generic	31
2.7.1 建立 C 语言源文件	31
2.7.2 建立头文件	35
2.7.3 建立资源描述文件	35
2.7.4 建立模块定义文件	35
2.7.5 建立 make 文件	36
2.7.6 运行 MAKE 程序	37
2.8 把 Generic 作为模板使用	37
2.9 小结	38

第三章 窗口输出

3.1 显示描述表	39
3.1.1 使用 GetDC 函数	40
3.1.2 WM_PAINT 消息	40
3.1.3 使用用户区域无效	41
3.1.4 显示描述表与设备描述表	41
3.1.5 坐标系	42
3.2 建立、选择和删除绘图工具	42
3.3 绘图与书写	43
3.4 应用程序示例 Output	44
3.4.1 增加新变量	44
3.4.2 增加 WM_CREATE 分支 语句	45
3.4.3 增加 WM_PAINT 消息	45
3.4.4 修改 WM_DESTROY 分支	

语句	48	5.1 什么是肖像	61
3.4.5 编译与连接	48	5.1.1 使用固有肖像	62
3.5 小结	49	5.2 使用自己的肖像	62
第四章 键盘和鼠标器输入	50	5.2.1 建立肖像文件	62
4.1 Windows 输入消息	50	5.2.2 定义肖像资源	62
4.1.1 消息的格式	50	5.2.3 载入肖像资源	63
4.1.2 键盘输入	51	5.3 指定类肖像	63
4.1.3 字符输入	51	5.4 显示自己的肖像	63
4.1.4 鼠标器输入	51	5.5 在对话框内显示肖像	64
4.1.5 定时器输入	52	5.6 应用程序示例 Icon	65
4.1.6 滚动条输入	53	5.6.1 增加 ICON 语句	65
4.1.7 菜单输入	54	5.6.2 增加 ICON 控制语句	65
4.2 应用程序示例 Input	54	5.6.3 设置类肖像	66
4.2.1 Input 应用程序怎样显示		5.6.4 把 MYICON.ICO 加入到	
输出	55	make 文件里	66
4.2.2 增加新变量	55	5.6.5 编译与连接	66
4.2.3 设置窗口类风格	56	5.7 小结	66
4.2.4 修改 CreateWindow 函数	56	第六章 光标、鼠标器和键盘	67
4.2.5 设置文本矩形	56	6.1 控制光标的形状	67
4.2.6 增加 WM_CREATE 分支		6.1.1 使用固有光标形状	67
语句	57	6.1.2 使用自定义的光标形状	68
4.2.7 修改 WM_DESTROY 分支		6.2 显示光标	68
语句	57	6.2.1 指定类光标	68
4.2.8 增加 WM_KEYUP 和		6.2.2 显式设置光标形状	69
WM_KEYDOWN 分支语句	57	6.2.3 在过长操作中显示砂漏状	
4.2.9 增加 WM_CHAR 分支语句	57	光标的例子	69
4.2.10 增加 WM_MOUSEMOVE 分支		6.3 让用户用鼠标器选择信息	70
语句	57	6.3.1 开始图形选择	71
4.2.11 增加 WM_LBUTTONDOWN 和		6.3.2 显示选择	72
WM_LBUTTONDOWN 分支		6.3.3 结束选择	73
语句	58	6.4 用键盘移动光标	74
4.2.12 增加 WM_LBUTTONDOWNBLCLK		6.4.1 使用键盘移动光标	74
分支语句	58	6.4.2 没有鼠标器时如何使用	
4.2.13 增加 WM_TIMER 分支		光标	76
语句	58	6.5 应用程序示例 Cursor	76
4.2.14 增加 WM_HSCROLL 和		6.5.1 增加 CURSOR 语句	77
WM_VSCROLL 分支语句	58	6.5.2 增加新变量	77
4.2.15 增加 WM_PAINT 分支		6.5.3 设置类光标	78
语句	59	6.5.4 准备砂漏状光标	78
4.2.16 编译与连接	59	6.5.5 增加长操作	78
4.3 小结	60	6.5.6 增加 WM_LBUTTONDOWN、	
第五章 肖像	61	WM_MOUSEMOVE 和	

WM_LBUTTONDOWN 分支	7.7.5 载入加速键表	102
语句	7.7.6 修改消息循环	103
6.5.7 增加 WM_KEYDOWN 和	7.7.7 修改 WM_COMMAND	
WM_KEYUP 分支语句	语句	103
6.5.8 增加 WM_PAINT 分支	7.7.8 编译与连接	104
语句	7.8 小结	104
6.5.9 在 make 文件里加入	第八章 控制窗口	105
BULLSEYE.CUR 文件	8.1 什么是控制窗口	105
6.5.10 编译与连接	8.2 建立控制窗口	105
6.6 小结	8.2.1 指定控制类	106
第七章 菜单	8.2.2 选择控制窗口风格	106
7.1 什么是菜单	8.2.3 设置父窗口	107
7.2 定义菜单	8.2.4 选择控制 ID	107
7.2.1 菜单 ID	8.3 使用控制窗口	107
7.3 在应用程序中包含菜单	8.3.1 接收用户输入	108
7.3.1 为窗口类指定菜单	8.3.2 发送控制窗口消息	108
7.3.2 为特定的窗口指定菜单	8.3.3 禁止或允许对控制窗口的	
7.4 处理菜单输入	输入	108
7.5 用应用程序里的菜单工作	8.3.4 移动控制窗口和改变其	
7.5.1 启动和禁止菜单项	大小	109
7.5.2 选中与不选中菜单项	8.3.5 消除控制窗口	109
7.5.3 增加菜单项	8.4 建立和使用一些常用控制窗口	109
7.5.4 修改现有菜单	8.4.1 按钮控制窗口	109
7.5.5 删除菜单项	8.4.2 静态控制窗口	112
7.5.6 使用点位图作为菜单项	8.4.3 列表框	113
7.5.7 更换菜单	8.4.4 组合框	119
7.5.8 建立新菜单	8.4.5 编辑控制窗口	120
7.5.9 初始化菜单	8.4.6 滚动条	122
7.6 特殊的菜单特性	8.5 应用程序示例 EditCntl	123
7.6.1 提供菜单加速键	8.5.1 在包含文件中增加一个	
7.6.2 使用重叠式菜单	常量	124
7.6.3 使用浮动弹出式菜单	8.5.2 增加新变量	124
7.6.4 设计自己的选中标记	8.5.3 增加 CreateWindow 函数	124
7.6.5 使用自己绘制的菜单	8.5.4 修改 WM_COMMAND 分支	
7.7 应用程序示例 EditMenu	语句	125
7.7.1 把新菜单加入到资源	8.5.5 增加 WM_SETFOCUS 分支	
文件中	语句	126
7.7.2 把定义加入到包含	8.5.6 增加 WM_SIZE 分支	
文件中	语句	126
7.7.3 把加速键表加入到资源	8.5.7 编译与连接	126
描述文件中	8.6 小结	126
7.7.4 增加新变量	第九章 对话框	128

MOUSEMOVE 和 WM_LBUTTONDOWN	172	第十三章 剪接板	193
11.6.8 增加 WM_RBUTTONDOWN 分支语句	173	13.1 使用剪接板	193
11.6.9 增加 WM_ERASEBKGD 分支语句	173	13.1.1 向剪接板中拷贝正文 ...	194
11.6.10 修改 WM_COMMAND 分支语句	173	13.1.2 从剪接板中拷出正文 ...	195
11.6.11 修改 make 文件	175	13.1.3 从剪接板中获取点位图 进行粘接	197
11.6.12 编译与连接	175	13.1.4 Windows 的剪接板应用 程序	199
11.7 小结	176	13.2 使用特殊的剪接板功能	199
第十二章 打印	177	13.2.1 根据请求放弃数据	199
12.1 在 Windows 环境中打印	177	13.2.2 在结束之前去掉数据 格式	200
12.1.1 使用打印机控制码	177	13.2.3 注册私有数据格式	200
12.2 获取当前打印机的信息	178	13.2.4 控制剪接板中的数据 显示	200
12.3 打印一行正文	179	13.3 应用程序示例 ClipText	202
12.4 打印点位图	180	13.3.1 增加新变量	203
12.5 打印过程中的出错处理	181	13.3.2 修改过程实例的初始化 代码	203
12.6 取消打印操作	182	13.3.3 增加 WM_INITMENU 分支语句	203
12.6.1 定义 Abort 对话框	183	13.3.4 修改 WM_COMMAND 分支语句	204
12.6.2 定义 Abort 对话框	183	13.3.5 增加 WM_PAINT 分支 语句	206
12.6.3 定义中止函数	183	13.3.6 增加 OutOfMemory 函数	206
12.6.4 执行可中止的打印 操作	185	13.3.7 编译和连接	207
12.6.5 用 ABORTDOC 控制码来 取消打印操作	185	13.4 小结	207
12.7 使用逐段生成技术来打印图形 ...	185	第十四章 C 与汇编语言	209
12.8 应用程序示例 PrntFile	187	14.1 选择内存模式	209
12.8.1 增加 AbortDlg 对话框 ...	187	14.2 使用 NULL	210
12.8.2 增加支持打印的新变 量	187	14.3 使用命令行参数及利用 DOS 环境	210
12.8.3 增加 IDM_PRINT 分支 语句	188	14.4 编写引出函数(exported function).....	211
12.8.4 创建 AbortDlg 和 AbortProc 函数	190	14.4.1 创建回调函数	211
12.8.5 增加 GetPrinterDC 函数	191	14.4.2 创建函数 WinMain	212
12.8.6 引出 (export) AbortDlg 函 数和 AbortProc 函数	191	14.5 使用 C 运行库函数	212
12.8.7 编译与连接	192	14.5.1 使用 Windows C 的库 程序	212
12.9 小结	192	14.5.2 分配内存	213

14.5.3	字符串操作	213	存储方式	240	
14.5.4	使用文件输入和输出	214	16.2.1	管理自动数据段	241
14.5.5	使用控制台输入和 输出	214	16.2.2	管理局部动态数据块	242
14.5.6	使用图形函数	215	16.2.3	管理全局内存块	246
14.5.7	使用浮点运算	215	16.2.4	使用窗口和类数据结构 中的附加字节	251
14.5.8	执行其他应用程序	215	16.2.5	管理资源	252
14.5.9	使用 BIOS 和 MS-DOS 的接口函数	216	16.3	使用内存模式	254
14.5.10	消除 C 运行库中的启 动代码	216	16.4	使用巨型数据	255
14.6	编写汇编语言代码	217	16.5	管理程序数据时应注意的问题	256
14.6.1	修改中断标志	218	16.6	程序代码占用内存的管理	258
14.6.2	用汇编语言编写引出函 数(exported function)	219	16.6.1	使用代码段属性	258
14.6.3	使用 ES 寄存器	219	16.6.2	使用多个代码段	258
14.7	小结	220	16.6.3	均衡代码段	259
第十五章	内存管理	221	16.6.4	.DEF 文件中代码段的 顺序	259
15.1	使用内存	221	16.7	小结	260
15.1.1	使用全局堆	222	第十七章	打印设置	261
15.1.2	使用局部堆	223	17.1	Windows 对打印设置的管理	261
15.1.3	处理可丢弃的内存	224	17.1.1	打印设置和 DEVMODE 结构	262
15.2	段的使用	225	17.1.2	打印设置和打印机 环境	263
15.2.1	代码段的使用	225	17.2	使用设备驱动程序函数	263
15.2.2	DATA 段的使用	226	17.3	获取打印机驱动程序的特性	264
15.3	应用程序示例 Memory	227	17.4	处理打印设置项	264
15.3.1	分割 C 语言源程序 文件	227	17.4.1	指定 ExtDeviceMode 函数 的输入输出	265
15.3.2	修改 Include 文件	227	17.4.2	获取完整的打印设置参 数	266
15.3.3	增加新的段定义	227	17.4.3	改变打印设置参数	266
15.3.4	修改 make 文件	228	17.4.4	使用 CreateDC 函数进行 打印设置	267
15.3.5	编译与连接	229	17.4.5	改变打印设置而不影响 其他应用程序	269
15.4	小结	229	17.4.6	提示用户改变打印 设置	269
第十六章	关于内存管理的进一步说明	230	17.5	在驱动程序间拷贝打印设置	271
16.1	Windows 的内存配置	230	17.6	维护应用程序自己的打印设置	271
16.1.1	基本内存配置	230	17.7	使用旧的打印机驱动程序	271
16.1.2	EMS 4.0 内存配置	232	17.8	小结	272
16.1.3	Windows 标准模式下的 内存配置	235	第十八章	字库	273
16.1.4	Windows386 增强模式下 的内存配置	238			
16.2	在 Windows 应用程序中使用的数据				

18.1	输出正文	273	第二十章 动态连接库	298
18.2	输出正文时彩色的使用	273	20.1 什么是 DLL	298
18.3	使用 GDI 给定的字库	274	20.1.1 引入(import)库与 DLL	299
18.4	创建逻辑字库	275	20.1.2 DLL 模块与应用程序模块	299
18.5	一行中使用多种字体	276	20.1.3 DLL 与任务	300
18.6	获取所选字库的有关信息	277	20.1.4 DLL 与堆栈	301
18.7	获取逻辑字库的信息	278	20.1.5 Windows 如何定位 DLL	301
18.8	枚举字库	279	20.2 何时使用用户 DLL	302
18.9	检查设备的正文输出能力	280	20.2.1 应用程序之间共享代码和资源	302
18.10	添加字库资源	281	20.2.2 针对不同的要求确定应用程序	303
18.11	设置正文输出格式	282	20.2.3 Windows 的中继器	304
18.12	创建字库资源文件	282	20.2.4 设备驱动程序	304
18.12.1	创建字库文件	282	20.2.5 用户控制	305
18.12.2	创建字库资源描述文件	283	20.2.6 项目管理	311
18.12.3	创建哑代码模块	283	20.3 创建 DLL	311
18.12.4	创建模块定义文件	283	20.3.1 建立 C 语言源程序文件	311
18.12.5	编译和连接字库资源文件	284	20.3.2 建立模块定义文件	316
18.13	应用程序示例 ShowFont	285	20.3.3 建立 make 文件	317
18.14	小结	285	20.4 应用程序访问 DLL 代码	319
第十九章 调色板	286	20.4.1 使用一个库函数的原型	319	
19.1	调色板的作用	286	20.4.2 移入库函数	319
19.2	调色板的工作原理	287	20.5 Windows 对象的拥有权原则	321
19.3	创建和使用逻辑调色板	288	20.6 库的示例 Select	322
19.3.1	创建 LOGPALETTE 数据结构	288	20.6.1 创建库函数	323
19.3.2	创建逻辑调色板	290	20.6.2 创建初始化例程	326
19.3.3	将调色板选入设备描述表	290	20.6.3 创建 Exit 例程	327
19.3.4	实现调色板	291	20.6.4 建立模块定义文件	327
19.4	用调色板上的颜色绘图	291	20.6.5 建立包含文件	327
19.4.1	直接指定调色板中的颜色	291	20.6.6 编译与连接	328
19.4.2	间接指定调色板中的颜色	292	20.7 小结	328
19.4.3	在画点位图时使用调色板	293	第二十一章 多文档界面	329
19.5	改变逻辑调色板	294	21.1 MDI 应用程序的结构	329
19.6	响应系统调色板所作的改变	295	21.2 初始化 MDI 应用程序	330
19.6.1	对 WM_QUERYNEWPALETTE 消息的响应	295	21.2.1 注册窗口类	330
19.6.2	对 WM_PALETTECHANGED 消息的响应	296	21.2.2 创建窗口	331
19.7	小结	297	21.3 编写主消息循环	331
			21.4 编写框架窗口函数	332
			21.5 编写子窗口函数	333
			21.6 数据与子窗口的对应	333
			21.6.1 将数据存储在窗口结构	

中	333	22.2 DDE 的一些概念	339
21.6.2 使用窗口特征	333	22.2.1 用户、服务器和会话	339
21.7 控制子窗口	334	22.2.2 应用程序、主题和项	339
21.7.1 创建子窗口	334	22.2.3 永久性的(“热的”或“暖 的”)数据链	340
21.7.2 撤销子窗口	335	22.3 DDE 消息	340
21.7.3 激活和禁止子窗口	335	22.4 DDE 消息流	341
21.7.4 在屏幕上安排用户子窗 口	335	22.4.1 会话的初始化	341
21.8 小结	336	22.4.2 单个项的传递	343
第二十二章 动态数据交换	337	22.4.3 建立永久性数据链	346
22.1 Windows 中的数据交换	337	22.4.4 在远程应用程序中执行 命令	351
22.1.1 用剪接板传输数据	337	22.4.5 结束会话	353
22.1.2 动态连接库	337	22.5 DDE 用户和服务器应用程序的 例子	354
22.1.3 动态数据交换	338	22.6 小结	355
22.1.4 Windows DDE 的使用	338		
22.1.5 从用户角度看 DDE	338		

第一章 Windows 环境概述

尽管 Windows 应用程序都用 C 语言编写而成,但它们在许多方面与标准 C 语言编写的程序有差别。应用程序要想在 Windows 环境下成功地运行,就必须与 Windows 及其它应用程序相互协作,控制 Windows 的操作,并必须与 Windows 及其它应用程序一道运行,共享系统资源。

在本章中,我们先扼要地介绍一下 Windows 3.0 所提供的环境及其特点。

Windows 3.0 具有许多标准 DOS 环境所没有的新特性。因为这些特性,Windows 应用程序在某种意义上来说比标准 DOS 程序更为复杂。

本章基本内容如下:

- Windows 应用程序与标准 DOS 应用程序的比较
- Windows 环境的特性和对开发应用程序的影响
- Windows 的程序设计模式
- Windows 应用程序的开发过程

1.1 Microsoft Windows 与 DOS 的比较

Microsoft Windows 具有许多标准 DOS 环境所没有的特性。正因如此,第一次接触 Windows 应用程序,可能感觉它比标准 DOS 程序更为复杂。这些特性包括:

- 为应用程序提供的窗口、菜单、对话框和控制特性的图形用户接口
- 队列式输入
- 与设备无关的图形
- 多任务
- 应用程序之间的动态数据交换

在 DOS 环境下编写应用程序时,大多数 C 程序员用标准 C 运行函数库完成程序的输入、输出、存储管理及其它操作。C 运行函数库假定使用的是标准操作环境,即基于字符的终端作为输入输出,独占计算机的系统存储和输入输出设备。而在 Windows 环境里,这些假定就不再有效了,Windows 应用程序和其它应用程序必须共享计算机的资源(包括 CPU)。Windows 应用程序通过基于图形的显示器、键盘和鼠标器与用户进行对话。

以下几节介绍标准 DOS 应用程序和 Windows 应用程序之间的主要差别。

1.1.1 用户界面

Windows 的主要设计目标之一就是能够同时访问大多数(如果做不到全部的话)应用程序。在多任务环境中,若能给所有应用程序在显示屏上分配一块区域,就能确保用户与所有应用程序打交道。有些操作系统处理多任务的方法是把显示屏全部分配给一个程序使用,而让其它程序在后台等待。而在 Windows 环境中,可保证每个应用程序都能对屏幕