

Visual C++ 6.0

编程经典

罗斌 编著

博览

BIANCHENG
JINGDIAN
BOLAN

武汉大学出版社



TP312C
L986

Visual C++ 6.0

编程经典

罗斌 编著

博览

BIANCHENG
JINGDIAN
BOLAN

武汉大学出版社



A0954419

图书在版编目(CIP)数据

Visual C++ 6.0 编程经典博览/罗斌编著. —武汉: 武汉大学出版社,
2001. 9

ISBN 7-307-03258-9

I. V… II. 罗… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 040279 号

责任编辑: 陈 刚 责任校对: 刘 欣 版式设计: 支 笛

出版: 武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件: wdp4@whu.edu.cn 网址: www.wdp.whu.edu.cn)

发行: 新华书店湖北发行所

印刷: 武汉大学出版社印刷总厂

开本: 787×1092 1/16 印张: 27 字数: 653 千字

版次: 2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷

ISBN 7-307-03258-9/TP·104 定价: 45.00 元

版权所有, 不得翻印; 凡购买我社的图书, 如有缺页、倒页、脱页等质量问题者, 请与当地图书销售部门联系调换。

前 言

日新月异的计算机技术对现代社会的每一个人来说既是机遇又是挑战，微软、雅虎等巨人的成功将一个个美丽的神话变成了活生生的现实，地球村每时每刻的变化对一切渴望成功的英才来说无一不是一次次心灵的激荡。

可视化编程技术的推出和发展，对于渴望在计算机编程方面有所作为的人来说无疑是一大福音。微软公司推出的 Visual C++ 6.0 作为迄今为止的面向对象的可视化编程工具的典型代表之一，对于 C++ 编程者来说更是锦上添花。它提供了前所未有的编程能力和便利性，它所带的各种各样的实用工具能适合各种编程风格，它方便、快捷、高效周到的程序设计方式和手段为每一位使用者提供了空前的编程速度和质量。

当然，学习 Visual C++ 6.0 编程也不是一件非常容易的事，但是，一旦你适应了它，它就会适应你，使你取得事半功倍的效果。本书不是一本介绍如何使用 Visual C++ 6.0 的说明书，没有沿着从一开始就讲解 C++ 语法，然后介绍 Visual C++ 6.0 的操作环境，其次再介绍各种工具的使用，最后介绍对话框，文档/视图等结构的应用程序的设计套路进行；而是采用下面这种思维模式进行：我要设计一个程序，该程序应具有某种功能，我如何才能 Visual C++ 6.0 中实现该功能。一旦涉及到操作 Visual C++ 6.0 的某种工具时，在适当的位置就会介绍该工具的使用方法，使读者在潜移默化中理解 Visual C++ 6.0，并充分地发挥它的巨大作用。

本书在编写过程中，集中了近 80 个典型的 Visual C++ 6.0 编程技巧，其中一个显著的特征是没有把这 80 个编程技巧分别作为一个个应用程序编写，而是把它集成在几个应用程序中，把可能出现的问题尽可能地减少到最低限度。因为一个编程技巧在一个应用程序中能够使用，并不能保证它一定能够在另一个程序中不出问题。如果一个编程技巧能和更多的编程技巧在一个程序中和谐共处，则表明这一技巧的兼容性更好，能够在更多的程序中使用。举个例子来说吧，在一个演示字体的程序中，如果这个程序只演示一次变化的字体，只需要把选择的字体对象选入设备情境对象进行输出显示即可，在使用完毕后不恢复原字体对象，程序也不会出现问题，因为程序关闭后自动释放所占用的内存；但是如果该程序需要调用其他字体，就可能引起死机等意想不到的问题发生。

当然，由于本人学识有限，虽然在编写过程中参考了大量的国内外最新资料，在书中一定还存在不少问题，敬请广大读者谅解。同时，本书在编写过程中，对于引用的资料尽可能地标明出处和原作者姓名，对于确实无法查到的资料出处和作者，在此一并表示感谢。

罗 斌

2001 年 6 月于重庆

目 录

第 一 编

如何编写一个具有实用功能的字处理编辑软件

一、程序运行特征	2
二、程序创建过程	2
三、程序运行机制	7
1. EditSoft 应用程序是怎样打开或新建一个文本文件的（文档/视图结构 的应用程序是怎样运行的）？	7
2. 在 MFC 编程中，为什么大多数应用程序采用文档/视图结构？	11
3. 文档和视图是怎样联系起来的？	12
4. 为什么在某些程序新创建的菜单中只输入了一个菜单 ID 标识就可以 实现某一种功能，而没有加入任何代码？	13
5. 为什么在应用程序的菜单（包括浮动菜单和弹出菜单）中有的菜单项 是灰的？	14
6. 为什么几乎所有的命令消息处理函数都必须有一个 ID 标识与之相关联？	15
7. 如何为程序添加一个浮动菜单？	18
8. 操作系统自带的记事本程序中的“帮助\关于记事本”对话框中的系统信息 是如何增加上去的？	26
9. 如何给应用程序做一个每日提示（在程序启动时显示）？	30
10. 为什么在应用程序中直接加入每日提示（Tip of the day）部件后，屏幕上 出现“提示文件”不存在的出错信息？	30
11. 如何汉化每日提示界面的所有英文？	34
12. 每日提示（Tip of the Day）控件是怎样在应用程序中加入自己的菜单的 （如何动态修改菜单）？	36
13. 怎样为应用程序（EditSoft）增加一个日历？	43
14. 如何使 EditSoft（单文档）应用程序成为多文档应用程序？	48
15. 如何修改应用程序标题栏上的显示文字？	55
16. 如何使应用程序以最大化窗口显示？	57
17. 如何在应用程序的状态条上显示时间？	58

18. 如何使应用程序窗口左上角标题栏上的应用程序图标动起来?	60
19. 如何使应用程序支持拖放功能?	64

四、程序功能扩充	65
20. 怎样建立一个文件打开、关闭对话框?	66
21. 如何使用字体、颜色对话框?	70
22. 如何处理文字输入?	82
23. 如何在字处理编辑器中增加一个插入符?	83
24. 如何使应用程序窗口滚动?	92
25. 如何处理方向键、翻页键等虚键操作?	99
26. 如何实现字符串的复制、粘贴?	103
27. 如何为文档增加打印功能?	112
28. 如何实现应用程序数据的读出和写入?	122

第 二 编

怎样编写一个位图及图形操作程序

一、程序运行特征	128
-----------------------	-----

二、程序创建过程	129
-----------------------	-----

三、程序运行机制	133
-----------------------	-----

1. 什么是 DDB 位图, 什么是 DIB 位图?	133
2. 如何创建并显示一幅 DDB 位图?	134
3. 如何显示一幅超过 256 色的位图?	135
4. 如何对位图进行伸缩处理?	136
5. 如何在同一程序中管理多个位图?	140
6. 如何使位图能够随着窗口大小的改变而改变?	147
7. 如何实现位图图像的全屏幕显示?	149
8. 如何在位图中插入图标?	152
9. 如何将一幅彩色位图转换为一幅单色三维位图?	156
10. 如何为位图程序制作一个封面 (即通常所说的闪屏)?	165

四、程序功能扩充	168
-----------------------	-----

11. 位图文件由哪几个部分组成?	168
12. 如何对一个位图文件进行读写操作?	170
13. 如何使程序窗口自适应位图大小?	206
14. 如何为位图程序增加拷贝、剪切和粘贴功能?	208
15. 如何在文件打开对话框中实现位图预览?	219

16. 如何给位图显示程序增加图形绘制能力?	228
17. 如何在绘制图形时改变鼠标的形状?	240
18. 如何实现绘制图形时的拉伸效果?	241
19. 如何对绘图工具的生成结果进行保存?	243
20. 如何在位图显示程序中进行文字输入?	255

第 三 编

怎样在 Windows 窗口实现文字的滚动显示

一、程序运行特征	262
二、程序创建过程	263
三、程序运行机制	275
1. 应用程序怎样实现文字在屏幕上的循环滚动显示?	275
2. 如何改变应用程序中滚动显示文字的字体?	278
3. 为什么在文档/视图结构的应用程序的 OnDraw()函数中, 创建好的字体 和画刷不是直接使用, 而是先通过 SelectObject()函数装入后再使用, 最 后通过 SelectObject()函数恢复原来的字体和画刷?	281
4. 为什么程序在实现背景音乐时, 不是在函数 OnCreate()中直接调用音乐 播放语句, 而是采用了多线程这一方式?	286
5. 如何在应用程序的状态条上实现文字的滚动显示?	287
6. 如何避免 (或减弱) 文字滚动时显示屏幕的闪烁现象?	291
7. 如何在背景上插入一幅精彩的位图 (包括超过 256 色的位图)?	292
8. 如何取消程序标题栏上的最大化按钮、最小化按钮、还原按钮及关闭按 钮?	296
9. 怎样在一幅渐变的背景中显示滚动的文字?	299
10. 如何利用 Windows 自带的“字体”对话框改变滚动文字的字体?	310
11. 如何使窗口客户区的文字大小实现渐变输出?	315
12. 如何改变在屏幕上滚动显示的文字?	317
13. 如何使窗口上滚动的文字能够随着窗口大小的改变而改变?	327
14. 如何使一个应用程序始终处于所有应用程序的前台?	332
15. 如何创建一个初始时极大或者极小的应用程序?	334
四、程序功能扩充	336

第 四 编

怎样编写一个 Windows 风格的简易计算器

一、程序运行特征	373
二、程序创建过程	374
三、程序运行机制	383
1. 计算器能够进行加、减、乘、除四则运算的运行机制是什么?	383
2. 计算器为什么能够进行数据的连续计算?	384
3. 计算器能否在不退出程序的情况下重新进行另外一次计算?	384
4. 计算器为什么能够即时将输入或运算结果显示出来?	384
5. 能否改变计算器的背景颜色?	386
6. 为什么在计算器程序启动后, 按下键盘上的“Esc”键或“Enter”后, 计算器会自动消失?	392
7. 如何控制编辑框的外观和功能?	395
8. 如何修改对话框的风格?	399
9. 如何在对话框程序中控制按钮是否显示?	402
10. 如何实现用鼠标拖动程序窗口中标题栏以外的区域来改变程序窗口在 屏幕上的位置?	415
四、程序功能扩充	416

第 一 编

如何编写一个具有实用功能的 字处理编辑软件

本编经典：

1. EditSoft 应用程序是怎样打开或新建一个文本文件的（文档/视图结构的应用程序是怎样运行的）？
2. 在 MFC 编程中，为什么大多数应用程序采用文档/视图结构？
3. 文档和视图是怎样联系起来的？
4. 为什么在某些程序新创建的菜单中只输入了一个菜单 ID 标识就可以实现某一种功能，而没有加入任何代码？
5. 为什么在应用程序的菜单（包括浮动菜单和弹出菜单）中有的菜单项是灰的？
6. 为什么几乎所有的命令消息处理函数都必须有一个 ID 标识与之相关联？
7. 如何为程序添加一个浮动菜单？
8. 操作系统自带的记事本程序中的“帮助\关于记事本”对话框中的系统信息是如何增加上去的？
9. 如何给应用程序做一个每日提示（在程序启动时显示）？
10. 为什么在应用程序中直接加入每日提示（Tip of the day）部件后，屏幕上出现“提示文件”不存在的出错信息？
11. 如何汉化每日提示界面的所有英文？
12. 每日提示（Tip of the Day）控件是怎样在应用程序中加入自己的菜单的（如何动态修改菜单）？
13. 怎样为应用程序（EditSoft）增加一个日历？
14. 如何使 EditSoft（单文档）应用程序成为多文档应用程序？
15. 如何修改应用程序标题栏上的显示文字？
16. 如何使应用程序以最大化窗口显示？
17. 如何在应用程序的状态条上显示时间？
18. 如何使应用程序窗口左上角标题栏上的应用程序图标动起来？
19. 如何使应用程序支持拖放功能？
20. 怎样建立一个文件打开、关闭对话框？
21. 如何使用字体、颜色对话框？
22. 如何处理文字输入？
23. 如何在字处理编辑器中增加一个插入符？

- 24. 如何使应用程序窗口滚动?
- 25. 如何处理方向键、翻页键等虚键操作?
- 26. 如何实现字符串的复制、粘贴?
- 27. 如何为文档增加打印功能?
- 28. 如何实现应用程序数据的读出和写入?

一、程序运行特征

在 Microsoft Visual C++ 6.0 集成开发环境中, Microsoft 公司提供了非常丰富的 C++ 类, 以至开发一个简单的字处理程序简单到不用添加一行代码即可生成具有字处理软件常用功能的程序的地步。图 VCP1-1 即是一个手工没有加入任何代码、完全由 Visual C++ 6.0 应用程序生成向导生成的字处理程序, 它能够对任何一个纯文本文件进行读入、保存、剪切、拷贝、粘贴、打印、打印预览等操作。本编将讲述怎样生成该程序并如何增强这类程序的功能。

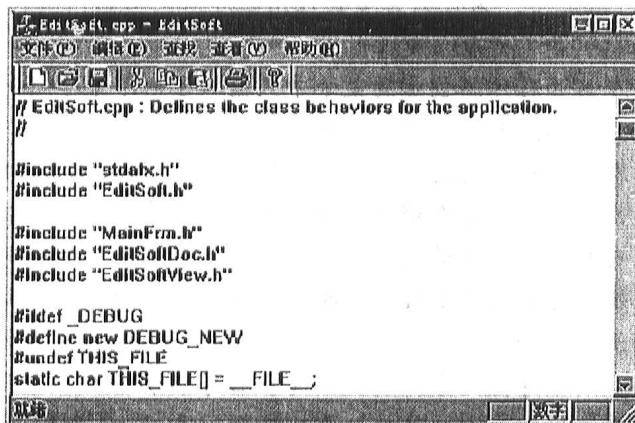


图 VCP1-1

二、程序创建过程

1. 在 Visual C++ 6.0 的集成开发环境中,使用应用程序生成向导(MFC AppWizard)创建一个工程项目 EditSoft, 步骤为:

(1) 打开菜单:File/New, 显示一个新建对话框。注意, 对话框的标签页必须指向“Projects”。

(2) 在新创建对话中, 先选择 MFC AppWizard(EXE), 在 Project Name 编辑框中填入工程项目名: EditSoft, 然后确保 Create new workspace 的圆按钮被选中 和 Win32 的检查框被选中, 单击“OK” 按钮进入下一步。

(3) 在 MFC AppWizard-Step 1 中有几个选项, 在新建程序类型 (What type of application would you like to create) 的单选按钮中选择单文档 (Single document) 类型, 在文档/视图结构支持 (Document/View architecture support) 复选框中做上标记, 表示需要文

档/视图支持, 在语言资源 (What language would you like your resource in) 下拉列表选项中, 选择中文 (中文[中国]APPWZCHS.DLL), 单击“下一步 (Next)”按钮。

(4) 进入 MFC AppWizard-Step2 后, 应用程序生成向导将询问是否需要数据库支持 (What database support would you like to include), 保留缺省值: 不需要 (None 单选按钮选中), 单击“下一步 (Next)”按钮。

(5) 在应用程序生成向导的第 3、4、5 步中均选择保留缺省值, 每一步骤只需单击“下一步 (Next)”按钮即可。

(6) MFC AppWizard-Step 6 是生成 EditSoft 程序最关键的一步, 在应用程序新生成的类中 (AppWizard create the following classes for you) 的列表框中选择视图类 (CEditSoftView), 在基类名 (Base class) 下拉列表框中选择 CEditView 类, 其余保留缺省值, 如图 VCP1-2 所示, 最后单击“完成 (Finish)”按钮, 出现新项目信息 (New Project Information) 对话框, 对话框对刚才应用程序生成向导的选择作了一下简单的总结, 单击“OK”按钮, 应用程序向导就自动生成了新建的 EditSoft 程序代码。

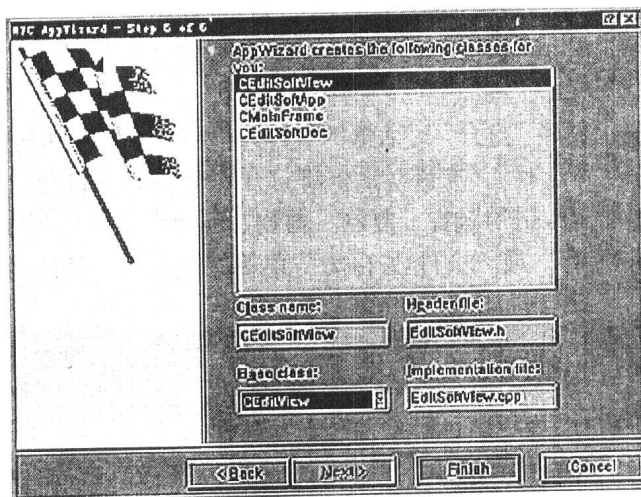


图 VCP1-2

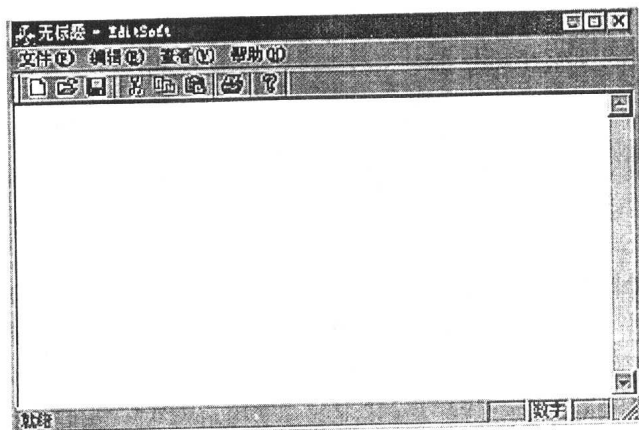


图 VCP1-3

2. 编译、连接并执行 EditSoft 应用程序，出现如图 VCP1-3 所示的效果。

3. 单击应用程序（EditSoft）工具栏上的“文件打开”按钮，应用程序将出现一个文件打开对话框，如图 VCP1-4 所示。

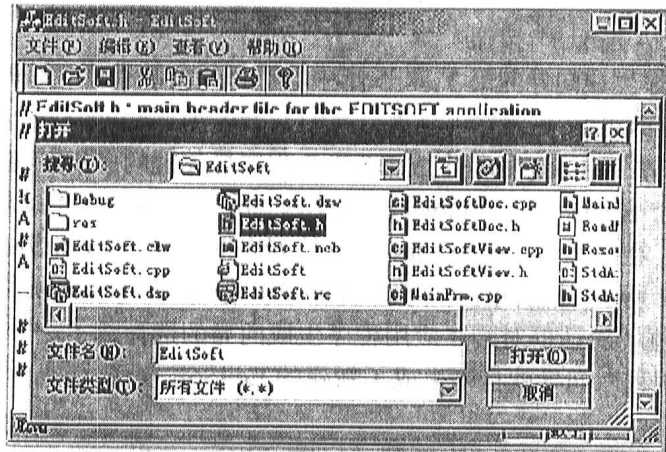


图 VCP1-4

4. 在 EditSoft 应用程序“打开”对话框的文件列表框中，双击任何一个文本文件（C++ 程序源代码也是一个文本文件），文件对话框将打开一个文件，如图 VCP1-1 所示。至此，就可以在程序中编辑（可以进行选择、复制、剪切、粘贴等操作）文本文件了。编辑完成以后可以选择“文件\保存”或“文件\另存为”菜单保存文件，如图 VCP1-5 所示。

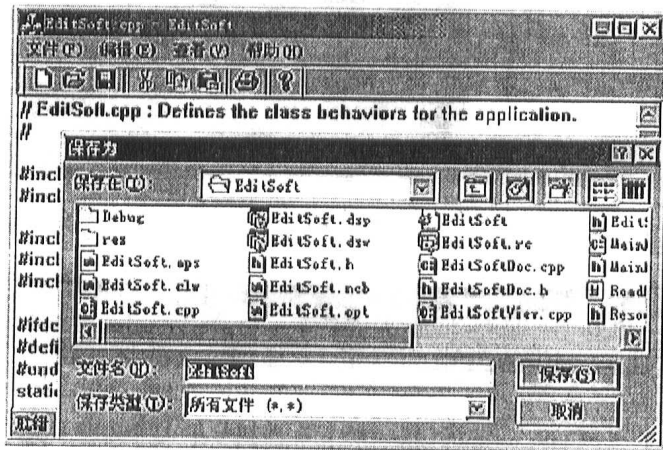


图 VCP1-5

以上只是在缺省状态下，由应用程序生成向导（MFC AppWizard）自动生成的应用程序的默认功能。实际上，CEditView 派生的视图类还有许多其他的功能，其功能几乎与 Windows 操作系统自带的记事本软件差不多。下面，我们将仿照记事本功能编写 EditSoft 程序。

5. 为 EditSoft 程序添加查找功能。打开项目工作空间（Workspace）的菜单资源编辑器

7. 为 EditSoft 程序添加替换功能。按照第 5 步的方法,打开项目工作空间(Workspace) 的菜单资源编辑器(ResourceView\Menu\IDR_MAINFRAME), 出现 IDR_MAINFRAME 菜单资源,在菜单资源中的“查找(一级菜单)”菜单下拉空白菜单(二级子菜单)的虚线框上双击左键,出现“查找”所属子菜单的对象属性(Menu Item Properties)对话框,在 ID 组合框中输入: ID_EDIT_REPLACE,在标题(Caption)编辑框中输入: 替换,如图 VCP1-8 所示,按照该图中所示的选择情况进行该菜单的属性设置。实际上,当在输入菜单 ID 标识 ID_EDIT_REPLACE 后,在提示(Prompt)编辑框中已经自动加入了“替换”菜单所需的提示文字(“用不同的正文替换指定的正文\n 替换”),在标题(Caption)编辑框中加入了“替换”。单击键盘上的回车键(Enter),退出“替换”子菜单的属性对话框。这样,就完成了“替换”菜单的设置。

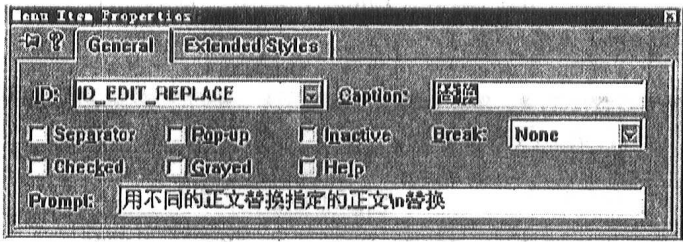


图 VCP1-8

8. 编译、连接并执行程序。打开一个文件,选择菜单“查找\替换”,将出现一个替换对话框,在“查找内容”编辑框中输入待查找的文字,在“替换为”编辑框中输入将要替换的文字,如图 VCP1-9 所示。单击“找下一个”按钮,光标将落在打开文件中的相应位置处,单击“替换”按钮,应用程序将一次替换一个查找内容,单击“全部替换”按钮,应用程序将全部替换打开文件中与“查找内容”相匹配的所有字符。如果选择“区分大小写”复选框,应用程序将区分大小写进行查找和替换。单击“取消”按钮,替换对话框将不做任何操作返回。

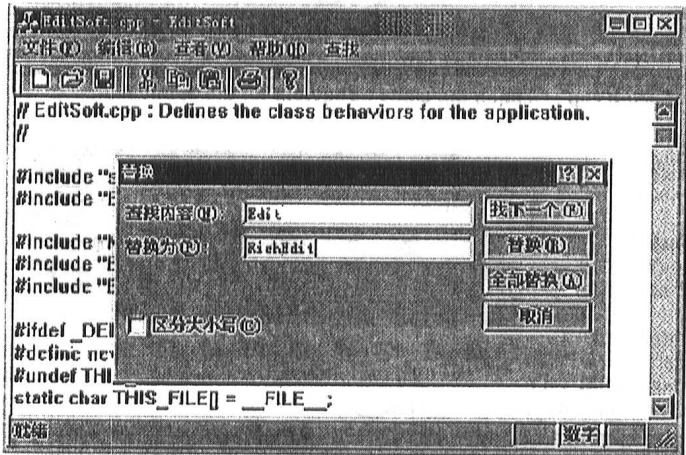


图 VCP1-9

9. 按照第 5、6、7、8 步的办法，在“编辑”菜单下加入以下菜单。

ID 标识	菜单名	功 能
ID_EDIT_CLEAR	删除	删除选定的内容
ID_EDIT_CLEAR_ALL	全部删除	删除应用程序中的全部内容
ID_EDIT_REPEAT	重复	重复上一步的操作
ID_EDIT_REDO	恢复	恢复上一步的操作
ID_EDIT_SELECT_ALL	全部选择	选择应用程序中的所有内容

编辑完成后的菜单如图 VCP1-10 所示。最后我们可以在新添加的菜单上加上一条分隔线以示区别。

10. 编译、连接并执行程序。读入一个文本文件，打开菜单“编辑”，出现如图 VCP1-10 所示的效果。此时，该应用程序就具有一个字处理程序所应具有的基本功能，可以在程序中进行文字的复制、拷贝、粘贴等操作，并可保存和打印文件。

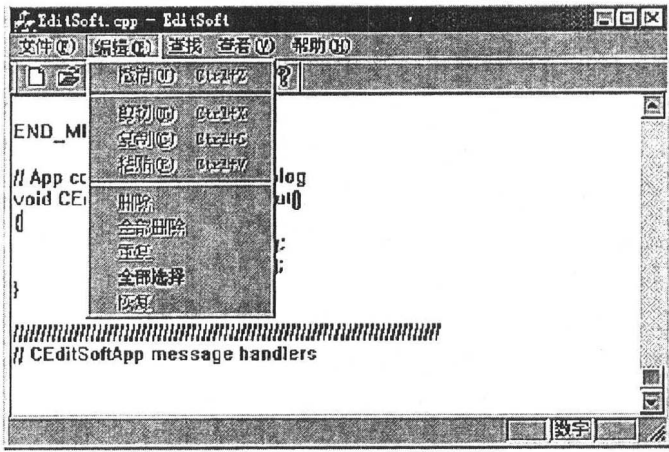


图 VCP1-10

三、程序运行机制

1. EditSoft 应用程序是怎样打开或新建一个文本文件的（文档/视图结构的应用程序是怎样运行的）？

在 Visual C++ 6.0 集成开发环境中，由应用程序生成向导（MFC AppWizard）生成的框架应用程序一般包含以下几个类：应用程序类、主框架类（多文档有子框架类）、视图类、文档类和关于类（About 类，可有可无）。

应用程序类 CWinApp 负责应用程序的初始化、运行和撤消等操作，应用程序类有一

个非常关键的函数 `InitInstance()` 负责程序的初始化, 该函数主要完成以下功能:

(1) 创建一个或几个文档模板, 继而创建文档、视图和框架窗口。文档、视图和框架就是通过文档模板有机地联系起来的。文档模板必须指明文档、框架及视图的类型。在文档模板类的构造函数的四个参数中, 除第一个参数是资源标识符外, 后面三个参数指明了该文档模板对象对应的文档类、框架类和视图类, 这三个类的对象都是在程序运行时动态构造的。如果需要在支持两种或多种文档类型, 必须为每个文档类型调用增加文档模板 `AddDocTemplate()` 函数, 应用程序的文档模板资源表中的每个文档模板都用一个图标来注册并代表它们, 文档模板的次序是由调用增加文档模板 `AddDocTemplate()` 函数添加模板的次序来决定的, MFC 默认应用程序的第一个图标资源是应用程序图标, 下一个图标资源是第一个文档图标, 依此类推。

(2) 从 Windows 操作系统注册文件中载入标准的文件选项, 并注册文档模板。

(3) 处理命令行以决定是打开一个文档, 还是新建一个文档。在启动应用程序时, 通常是新建一个空文档, 这时应用程序类发送 `ID_FILE_NEW` 命令消息, 从而调用应用程序类的 `OnFileNew()` 函数; `OnFileNew()` 函数再调用文档模板类 (单或多文档模板) 的 `OpenDocumentFile()` 函数; 由于没有提供文件名, 因此 `OpenDocumentFile()` 函数总是首先构造文档对象, 再构造文档边框窗口对象并创建文档边框窗口, 此时并没有将框架窗口显示出来, 创建文档边框窗口显示时调用的是 API 函数 `CreateWindowEx()`, 对该函数的调用会导致发送 `WM_CREATE` 消息给文档边框窗口, 使得调用框架类的消息处理函数 `OnCreate()` 函数; 在 `OnCreate()` 函数中要完成构造视图对象、创建视图窗口的工作, 对于 SDI 文档类型, 由于边框窗口就是主边框窗口, 因而还要创建工具条和菜单条; `OpenDocumentFile()` 函数完成了文档、文档边框窗口、视图的构造后, 就调用文档类的 `OnNewDocument()` 函数对文档对象进行初始化, 在 `OnNewDocument()` 函数中又要调用文档类的 `DeleteContents()` 函数进行文档内容的清除, 从而使一开始便出现一个空白的窗口。完成文档的初始化后, 应用程序便调用 `m_pMainWnd->ShowWindow(SW_SHOW)` 函数和 `m_pMainWnd->UpdateWindow()` 函数显示窗口。

见下面的代码:

```
BOOL CEditSoftApp::InitInstance() //应用程序初始化函数
{
    AfxEnableControlContainer();

    // Standard initialization
    // If you are not using these features and wish to reduce the size
    // of your final executable, you should remove from the following
    // the specific initialization routines you do not need.

#ifdef _AFXDLL
    Enable3dControls();           // Call this when using MFC in a shared DLL
#else

```

```

    Enable3dControlsStatic();    // Call this when linking to MFC statically
#endif

    // Change the registry key under which our settings are stored.
    // TODO: You should modify this string to be something appropriate
    // such as the name of your company or organization.
    SetRegistryKey(_T("Local AppWizard-Generated Applications"));

    LoadStdProfileSettings(); // Load standard INI file options (including MRU)
    //载入注册文件
    // Register the application's document templates. Document templates
    // serve as the connection between documents, frame windows and views.

    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate( //新建一个文档模板
        IDR_MAINFRAME,
        RUNTIME_CLASS(CEditSoftDoc),
        RUNTIME_CLASS(CMainFrame),      // main SDI frame window
        RUNTIME_CLASS(CEditSoftView));
    AddDocTemplate(pDocTemplate); //增加文档模板

    // Parse command line for standard shell commands, DDE, file open
    CCommandLineInfo cmdInfo;
    ParseCommandLine(cmdInfo); //分析启动命令行

    // Dispatch commands specified on the command line
    if (!ProcessShellCommand(cmdInfo))
        return FALSE;

    // The one and only window has been initialized, so show and update it.
    m_pMainWnd->ShowWindow(SW_SHOW); //显示窗口
    m_pMainWnd->UpdateWindow();

    return TRUE;
}

```

接着，应用程序类调用内嵌的 WinMain() 函数的 Run() 函数进行消息循环。当从“文件”菜单的“新建”子菜单或“打开”子菜单创建一个新文档时，文档模板也创建一个新的窗口，这是由传给文档模板构造函数的变量决定的。CWinApp 类及其派生类实际上并不直接