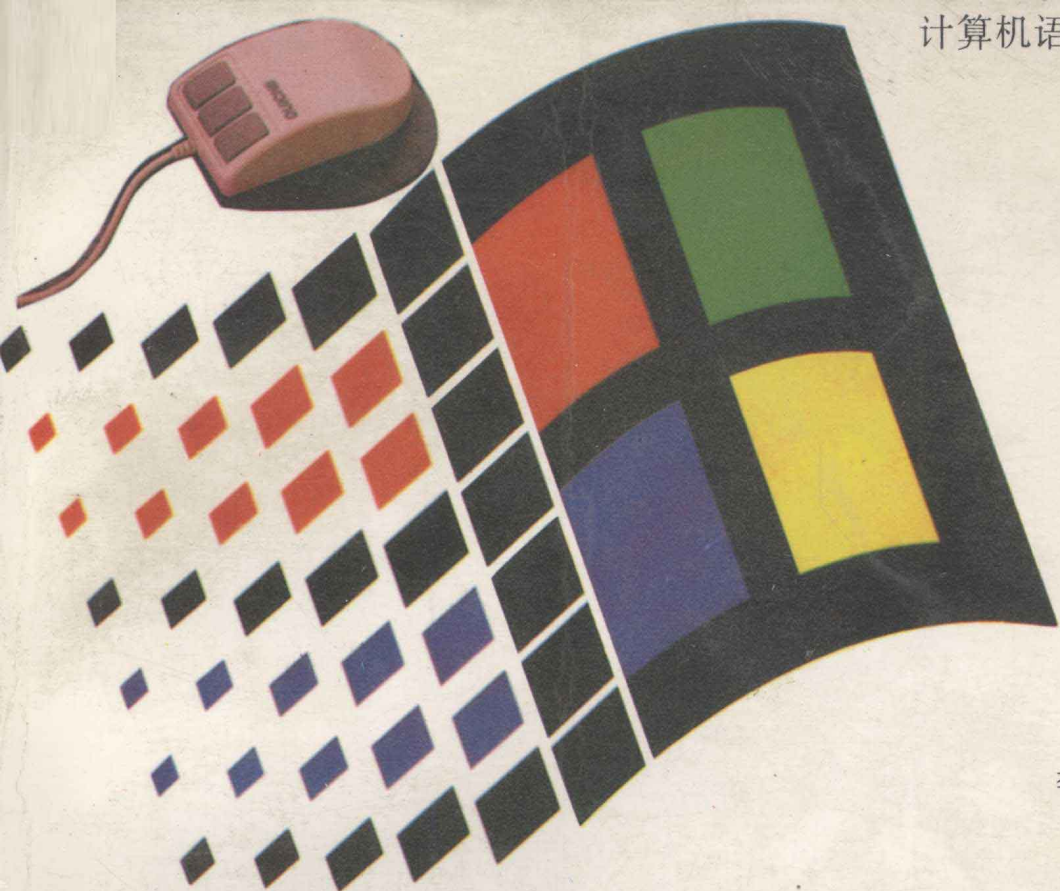


计算机语言技术系列丛书



李 伟 编著

# WINDOWS 3.X

## 程序设计与参考全书

学苑出版社

计算机语言技术系列丛书

Microsoft Windows 3. X

# 程序设计与参考全书

李 伟 亦 鸥 著  
希 望 审校

学苑出版社

1993

(京)新登字 151 号

## 内 容 简 介

本书是进行 Microsoft Windows 程序设计的指导性读物,书中介绍了 Windows 编程的有关要点,并给出了有用的一些参考信息,如相关的函数和消息等。

本书对从事软件设计、开发和应用的技术人员具有重要的参考价值。

需要本书的用户,可与北京 8721 信箱联系,邮码 100080,电话 2562329。

### 计算机语言技术系列丛书 Microsoft Windows 3. X 程序设计与参考全书

---

著 者: 李 伟 亦 鸥  
审 校: 希 望  
责任编辑: 徐建军  
出版发行: 学苑出版社 邮政编码: 100032  
社 址: 北京市西城区成方街 33 号  
印 刷: 北京东升印刷厂印刷  
开 本: 787×1092 1/16  
印 张: 27.125 字数: 631 千字  
印 数: 1—5000 册  
版 次: 1993 年 12 月北京第 1 版第 1 次  
I S B N: 7-5077-0776-8/TP·8  
本册定价: 29.00 元

---

学苑版图书印、装错误可随时退换

# 目 录

## 第一部分 Windows 程序设计基础

|                                 |    |
|---------------------------------|----|
| <b>Windows 简介</b> .....         | 1  |
| Windows 程序设计 .....              | 2  |
| <b>第一章 Windows 环境</b> .....     | 3  |
| 1.1 Windows 发展史 .....           | 3  |
| 1.2 Windows 适用性广 .....          | 3  |
| 1.2.1 方便使用 .....                | 3  |
| 1.2.2 程序设计优势 .....              | 4  |
| 1.3 窗口组成 .....                  | 4  |
| 1.3.1 边框 .....                  | 4  |
| 1.3.2 用户区 .....                 | 5  |
| 1.3.3 控制菜单 .....                | 5  |
| 1.3.4 水平卷滚条 .....               | 5  |
| 1.3.5 扩大窗口按钮 .....              | 5  |
| 1.3.6 菜单项 .....                 | 6  |
| 1.3.7 缩小窗口按钮 .....              | 6  |
| 1.3.8 标题项 .....                 | 6  |
| 1.3.9 垂直卷滚条 .....               | 6  |
| 1.4 Windows 函数 .....            | 6  |
| 1.5 Windows 消息 .....            | 7  |
| 1.6 匈牙利法则 .....                 | 7  |
| 1.7 句柄 .....                    | 7  |
| 1.8 Include 文件 .....            | 8  |
| <b>第二章 Windows 程序设计基础</b> ..... | 9  |
| 2.1 事件驱动 .....                  | 9  |
| 2.2 WinMain 函数 .....            | 9  |
| 2.3 消息循环 .....                  | 12 |
| 2.4 Windows 过程 .....            | 13 |
| 2.5 工程文件 .....                  | 14 |
| 2.6 Windows 应用程序的常用文件类型 .....   | 15 |
| 2.6.1 C 源代码文件 .....             | 15 |

|            |                             |           |
|------------|-----------------------------|-----------|
| 2.6.2      | 模块定义文件 .....                | 15        |
| 2.6.3      | 资源文件 .....                  | 15        |
| 2.6.4      | Make 文件 .....               | 16        |
| 2.7        | 一个简单的 Windows 程序 .....      | 16        |
| <b>第三章</b> | <b>图形设备接口 .....</b>         | <b>20</b> |
| 3.1        | 设备描述表(Device Context) ..... | 20        |
| 3.2        | 映射模式(mapping Mode) .....    | 21        |
| 3.3        | 坐标 .....                    | 22        |
| 3.3.1      | 设备坐标系 .....                 | 22        |
| 3.3.2      | 逻辑坐标系 .....                 | 22        |
| 3.4        | GDI 图形和点 .....              | 22        |
| 3.5        | 画线 .....                    | 25        |
| 3.6        | 椭圆弧 .....                   | 29        |
| 3.7        | 画笔(pens) .....              | 30        |
| 3.8        | 绘图模式 .....                  | 30        |
| 3.9        | 填充区 .....                   | 31        |
| 3.10       | 边框 .....                    | 31        |
| 3.11       | 填充一块区域 .....                | 35        |
| 3.12       | 字符和字体 .....                 | 39        |
| 3.12.1     | 字符显示函数 .....                | 39        |
| 3.12.2     | 字符的设备描述表 .....              | 43        |
| 3.12.3     | 字体 .....                    | 44        |
| 3.12.4     | 字符大小 .....                  | 48        |
| 3.12.5     | 字符对齐 .....                  | 50        |
| 3.12.6     | 滚动字符窗口 .....                | 51        |
| 3.13       | 打印 .....                    | 56        |
| <b>第四章</b> | <b>资源 .....</b>             | <b>58</b> |
| 4.1        | 键盘加速 .....                  | 58        |
| 4.2        | 点位图 .....                   | 62        |
| 4.3        | 光标 .....                    | 66        |
| 4.4        | 对话框 .....                   | 70        |
| 4.5        | 图标 .....                    | 76        |
| 4.6        | 菜单 .....                    | 80        |
| 4.7        | 字符串 .....                   | 85        |
| <b>第五章</b> | <b>键盘 .....</b>             | <b>92</b> |
| 5.1        | 键盘输入 .....                  | 92        |
| 5.2        | 键盘消息 .....                  | 93        |
| 5.3        | lParam .....                | 93        |
| 5.4        | wParam .....                | 94        |

|            |                   |            |
|------------|-------------------|------------|
| 5.5        | 字符消息              | 95         |
| 5.6        | 字符集               | 96         |
| 5.7        | 脱字符               | 96         |
| 5.8        | 键盘应用实例            | 96         |
| <b>第六章</b> | <b>鼠标</b>         | <b>103</b> |
| 6.1        | 鼠标输入              | 103        |
| 6.2        | 鼠标消息              | 104        |
| 6.2.1      | 命中测试消息            | 104        |
| 6.2.2      | 用户区鼠标消息           | 105        |
| 6.2.3      | 非用户区鼠标消息          | 105        |
| 6.3        | 鼠标应用实例            | 106        |
| <b>第七章</b> | <b>子窗口</b>        | <b>112</b> |
| 7.1        | 创建窗口              | 112        |
| 7.1.1      | 定义窗口类             | 112        |
| 7.1.2      | 创建窗口              | 114        |
| 7.2        | 子窗口控制             | 120        |
| 7.3        | 子窗口应用举例           | 122        |
| <b>第八章</b> | <b>内存管理</b>       | <b>126</b> |
| 8.1        | 局部堆和全局堆           | 126        |
| 8.2        | 段                 | 129        |
| <b>第九章</b> | <b>多文档界面(MDI)</b> | <b>130</b> |
| 9.1        | MDI 应用程序          | 130        |
| 9.2        | MDI 消息循环          | 130        |
| 9.3        | MDI 消息            | 131        |
| 9.4        | 主窗口函数和子窗口函数       | 131        |
| 9.5        | MDI 实例            | 131        |
| <b>第十章</b> | <b>动态连接库</b>      | <b>141</b> |
| 10.1       | 静态连接和动态连接         | 141        |
| 10.2       | Import 库          | 141        |
| 10.3       | DLL 的代码结构         | 141        |
| 10.4       | 生成 DLL            | 143        |
| 10.5       | 使用 DLL            | 145        |

## 第二部分 Windows 函数与消息

|             |                     |            |
|-------------|---------------------|------------|
| <b>第十一章</b> | <b>Windows 函数概述</b> | <b>149</b> |
| 11.1        | 应用程序执行函数            | 149        |
| 11.2        | 原子函数                | 149        |
| 11.3        | 点位图函数               | 150        |

|       |              |     |
|-------|--------------|-----|
| 11.4  | 插入符函数        | 150 |
| 11.5  | 剪贴板函数        | 150 |
| 11.6  | 剪贴函数         | 151 |
| 11.7  | 通用对话框函数      | 151 |
| 11.8  | 通讯函数         | 151 |
| 11.9  | 调色板函数        | 152 |
| 11.10 | 坐标函数         | 152 |
| 11.11 | 光标函数         | 152 |
| 11.12 | DDE 函数       | 152 |
| 11.13 | 调试函数         | 153 |
| 11.14 | 设备环境函数       | 153 |
| 11.15 | 对话框函数        | 153 |
| 11.16 | 显示和移动函数      | 154 |
| 11.17 | Drag-drop 函数 | 154 |
| 11.18 | 绘图属性函数       | 155 |
| 11.19 | 绘图工具函数       | 155 |
| 11.20 | 椭圆和多边形函数     | 155 |
| 11.21 | 环境函数         | 155 |
| 11.22 | 错误函数         | 156 |
| 11.23 | 文件 I/O       | 156 |
| 11.24 | 字体函数         | 156 |
| 11.25 | 硬件函数         | 156 |
| 11.26 | 中继函数         | 157 |
| 11.27 | 图标函数         | 157 |
| 11.28 | 信息函数         | 157 |
| 11.29 | 文件初始化函数      | 158 |
| 11.30 | 输入函数         | 158 |
| 11.31 | 设备驱动程序函数     | 158 |
| 11.32 | 压缩、反压缩函数     | 159 |
| 11.33 | 线输出函数        | 159 |
| 11.34 | 映射函数         | 159 |
| 11.35 | 内存管理函数       | 160 |
| 11.36 | 菜单函数         | 160 |
| 11.37 | 消息函数         | 161 |
| 11.38 | 元文件管理函数      | 161 |
| 11.39 | 模块管理函数       | 161 |
| 11.40 | 网络函数         | 162 |
| 11.41 | OLE 函数       | 162 |
| 11.42 | 操作系统中断函数     | 163 |

|             |                             |            |
|-------------|-----------------------------|------------|
| 11.43       | 优化工具函数 .....                | 163        |
| 11.44       | 绘图函数 .....                  | 163        |
| 11.45       | 打印机控制函数 .....               | 164        |
| 11.46       | 特性函数 .....                  | 164        |
| 11.47       | 矩形函数 .....                  | 164        |
| 11.48       | 区域函数 .....                  | 165        |
| 11.49       | 窗口创建函数 .....                | 165        |
| 11.50       | 资源管理函数 .....                | 165        |
| 11.51       | 滚动函数 .....                  | 166        |
| 11.52       | 段函数 .....                   | 166        |
| 11.53       | 外壳(shell)函数 .....           | 166        |
| 11.54       | Stress 函数 .....             | 166        |
| 11.55       | 字符串处理函数 .....               | 167        |
| 11.56       | 系统函数 .....                  | 167        |
| 11.57       | 任务函数 .....                  | 167        |
| 11.58       | 文本输出函数 .....                | 167        |
| 11.59       | Toolhelp 函数 .....           | 168        |
| 11.60       | TrueType 函数 .....           | 168        |
| <b>第十二章</b> | <b>Windows 函数速查索引</b> ..... | <b>169</b> |
| <b>第十三章</b> | <b>Windows 函数参考</b> .....   | <b>191</b> |
| <b>第十四章</b> | <b>Windows 消息简介</b> .....   | <b>347</b> |

### 第三部分 附 录

|             |                               |            |
|-------------|-------------------------------|------------|
| <b>附录 A</b> | <b>Microsoft C 函数索引</b> ..... | <b>373</b> |
| <b>附录 B</b> | <b>数据结构</b> .....             | <b>388</b> |



## Windows 简介

Microsoft Windows 是为 MS-DOS 开发的图形用户界面集成环境。Windows 的多任务环境为其应用程序提供了统一的窗口和菜单结构,所以 Windows 应用程序比传统的 DOS 程序易于掌握和使用。Windows 的多任务性使用户可以同时运行几个程序,这一特性尤其适用于专为 Windows 环境开发的应用程序,但是在 Windows 下也可以同时运行多个非 Windows 的 DOS 应用程序。Windows 集成环境见图 1.1。

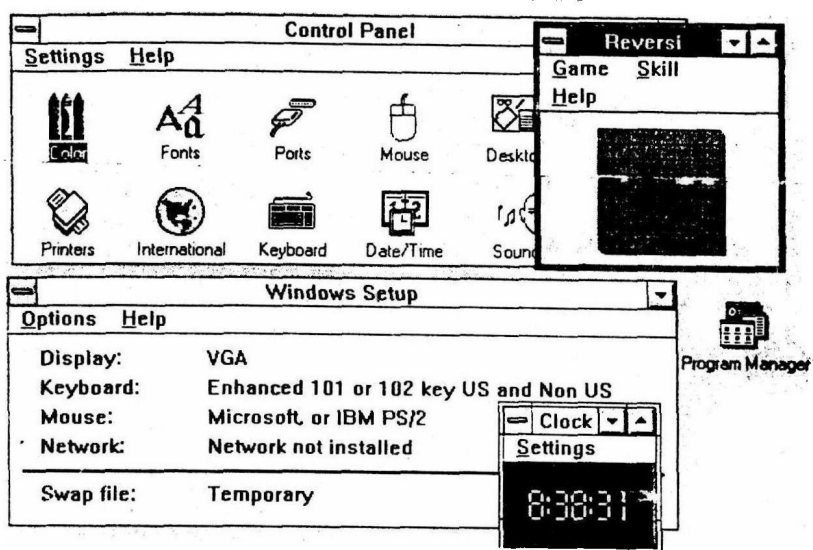


图 1.1 Windows 多任务环境

# Windows 程序设计

第一章简述 Windows 环境的特点及 Windows 编程与传统 DOS 编程的区别。

第二章介绍 Windows 3.1 程序设计所需的基本知识。包括程序的各个组成部分：  
Winmain 函数，消息循环(message loop)，模块定义文件和资源文件。

第三章介绍图形设备接口的功能，包括线、刷、笔和设备环境及文本输出。

第四章介绍键盘加速、点位图(bitmap)、光标、对话框，及使用实例。

第五章介绍使用键盘的原则。包括键盘消息、字符消息、字符集和脱字号(‘^’)。

第六章介绍在 Windows 下使用鼠标的规则。本章的重点是鼠标消息。

第七章介绍 Windows 环境下的窗口和子窗口概念及控制方法。

第八章介绍 Windows 局部堆栈、全局堆栈和段的使用。

第九章介绍多文档界面(Multiple Document Interface MDI)。

第十章介绍动态连接库的概念，本章创建一个动态链接库并演示其作用。

# 第一章 Windows 环境

## 1.1 Windows 发展史

Microsoft 于 1983 年完成了 Windows 的最初版本并在 1985 年公布了 Windows1.1 版。它适用于 256K 内存、两个软驱的 8080 系统。Windows1.1 自动进行应用程序窗口覆盖并支持弹出式菜单。

1987 年,Microsoft 公布了 Windows2.0。这一版本具有运行期窗口覆盖功能,并可以使用扩展内存。但是 Windows2.0 只能在实模式下运行,其寻址空间为 1 兆字节。

Microsoft 于 1990 年发表了 Windows3.0,这一版本增加了对用户界面的菜单、列表框、按键的支持。在 386 或 486 机上运行时可以寻址 16 兆字节的内存空间。Windows3.0 很快成为了 MS-DOS 机器的标准图形用户界面。

Windows3 的重大改进有:

- 更好的动态数据交换功能
- 客体链接及嵌入(embedded)功能
- 多媒体运行
- 更多的辅助工具
- 更好的内存管理

## 1.2 Windows 适用性广

图形用户界面使用点位图(bitmap)显示信息。因此不局限于传统 MS-DOS 的 ASCII 码,可以生动地显示各种的图形。输出设备与屏幕上图形的关系称为“所见即所得”(WYSIWYG),这也是 Wicrosoft Windows 流行的一个原因。Microsoft Windows 就是 MS-DOS 的图形用户界面,目前没有其它界面可以与 Windows 相提并论。

### 1.2.1 方便使用

界面的标准统一给用户带来很大方便。在 Windows 环境下,每一个应用程序的界面都有相同的基本特征,所以用户会很容易地学会使用新的应用程序。

Windows 使用图像表示程序或数据,如文件,应用程序,窗口和目录。用户可以使用鼠标来操纵这些物理结构,如选择、确认、移动等等。在某一图标上击键比到特定的目录下键入文件名字要简单得多。

WYSIWYG(所见即所得)是另一大优点。大多数文本方式下的字处理软件使用 ROMBIOS 的字符进行屏幕显示。因为其字符大小、形状、字体都是固定不变的,所以屏幕上

显示的图形与打印机上打印的图形并不相同。在 Windows 的 WYSIWYG 功能下,Windows 把文件看成一系列的图象,并可以把其显示为打印机打印的效果。

### 1.2.2 程序设计优势

统一的界面给程序员带来很大方便。程序员可以把主要精力用于程序功能的设计而不是界面的设计。

Windows 允许程序员设计表示物理结构的图标,并且提供弹出式菜单和对话框。Windows 还可直接执行鼠标和键盘输入,这大大减少了开发所用的时间。

Windows 的内存管理功能使程序可以用传统的 MS-DOS 程序访问更多的内存空间。程序员可以更好地利用系统的内存资源并保持设计的灵活性。

Windows 还有设备独立图形(Device independent graphics)的功能。设计良好的 Windows 应用程序不会直接访问图形硬件(屏幕和打印机),所以在具有 Windows 设备驱动程序的情况下应用程序可以运行于任何子系统或打印机上。对于程序员来说,设备独立图形意味着编码不依赖于特定的系统配置,不需要开发任何图形显示,适配器或打印机的驱动程序。

## 1.3 窗口组成

Windows 应用程序使用窗口进行屏幕的输入、输出。程序创建窗口并访问它,但窗口的管理由 Windows 和应用程序共同完成。Windows 管理窗口的大小、位置和窗口和各个组成部分。应用程序管理窗口的用户区(工作区)。应用程序窗口包括以下组成部分:

- 窗口边框
- 用户区
- 控制菜单框
- 水平卷滚条
- 扩大边框
- 菜单项
- 缩小边框
- 窗口名称
- 垂直卷滚条

图 1.2 是 Windows Notepad 的窗口,它含有以上所列的各项。下面分别介绍各个组成部分的功能及特点。

### 1.3.1 边框

窗口的边框处于应用程序窗口的最外侧。边框包括以下三个基本要素:

- 四个角
- 垂直边框
- 水平边框

窗口的四个角决定窗口在水平和垂直方向上的大小。垂直边框可以决定窗口水平方向上的大小。水平边框可以决定窗口竖直方向上的大小。

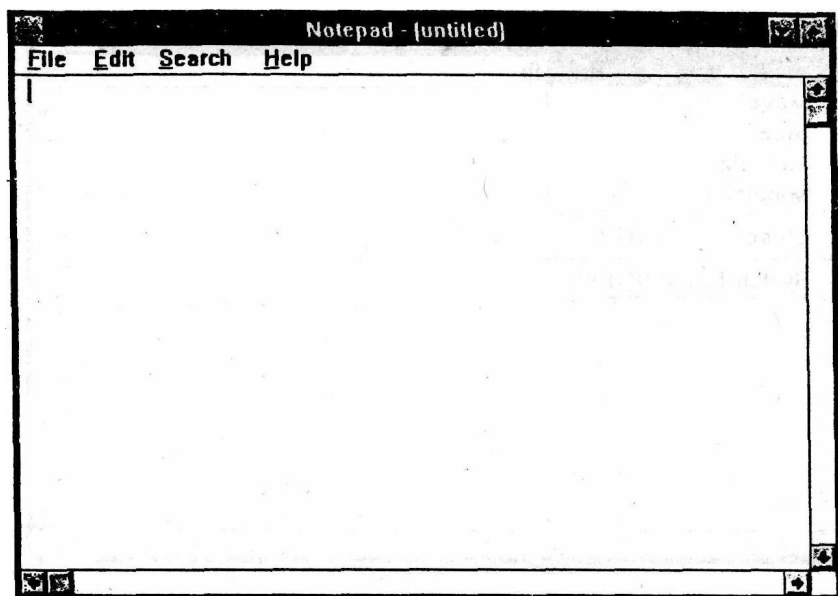


图 1.2 Notepad 窗口

### 1.3.2 用户区

用户区即窗口中没有被菜单选项、卷滚条、边框或其它成分占据的部分。应用程序把用户区作为工作空间,应用程序管理用户区。

### 1.3.3 控制菜单

控制菜单框位于窗口的左上角。提供对控制菜单的访问。

利用控制菜单中的选择项,用户可以恢复、移动、扩大、缩小或关闭应用程序窗口。如果没有鼠标,控制菜单(或者称为系统菜单)提供对应用程序窗口各个部分的基本访问。图 1.3 是 Notepad 的控制菜单。

### 1.3.4 水平卷滚条

水平卷滚条包括三个部分。第一个部分是左箭头,在卷滚条的最左侧。在此箭头上击键显示用户区左边框左侧的部分。第二部分是右箭头,位于卷滚条的最右侧,在此箭头上击键显示右边框右边的部分。最后一部分是一个卷滚块,它表示当前显示部分所处的位置。

### 1.3.5 扩大窗口按键

它可以把应用程序窗口扩大为整屏。按下此键后,它变为恢复(Restore)状态。按恢复键可以使窗口恢复到原来的大小。在窗口恢复原来大小后,恢复键又变为了扩大键。

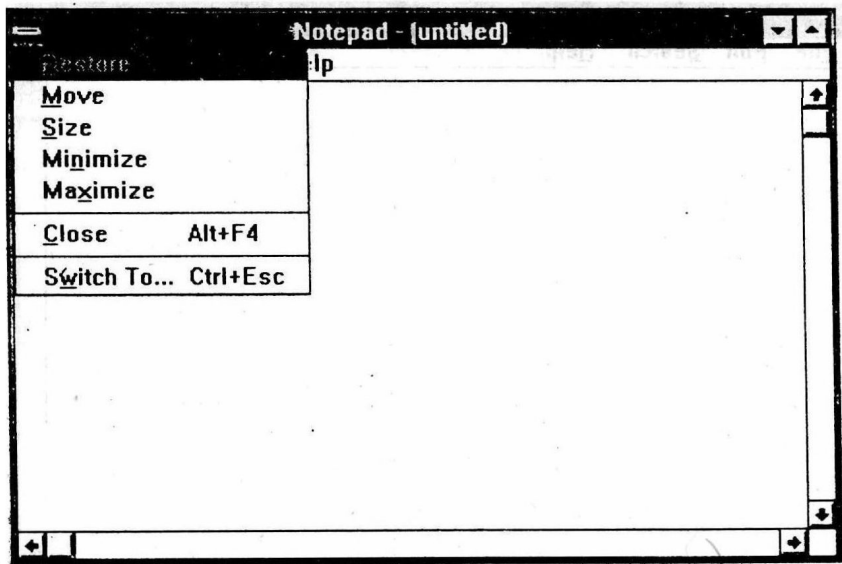


图 1.3 Notepad 的控制菜单

#### 1.3.6 菜单项

菜单项列出程序的菜单。大多数应用程序包括 FILE、EDIT 和 Help 项。但程序员可以改变菜单项的数量和类型。

#### 1.3.7 缩小窗口按键

此键可以缩小应用程序窗口到最小值。按此键后,应用程序窗口变小,再按可以使窗口恢复到原来的尺寸。

#### 1.3.8 标题项

标题项列出应用程序的名字和应用程序窗口中的当前激活的文件名。激活态窗口的标题项在颜色和亮度上都区别于处于挂起状态窗口的标题项。

#### 1.3.9 垂直卷滚条

垂直卷滚条包括三个部分。第一部分是向上箭头,处于卷滚条的最上端。按此箭头显示窗口上边框以上的部分。第二部分是向下箭头,位于卷滚条的最下端,按此箭头显示窗口下边框以下的文件部分。最后一部分是卷滚箭头,指出当前显示区的开始部分。卷滚条中间的方块表示当前显示区在整个文件中所处的位置。

### 1.4 Windows 函数

Windows 函数是 Windows 应用程序中最主要的部分。用 C 语言写的 MS-DOS 程序使用

编译器运行库的各种函数。编译器的运行库函数都是面向 80X86 结构专门开发的。当开发 Windows 应用程序时,其函数的设备独立性会给用户带来很大的方便。

Windows 函数总共有 600 多个。本书给出每一个函数的参考资料。

## 1.5 Windows 消息

Windows 的程序使用消息与其它 Windows 应用程序或 Windows 系统通讯。因为 Windows 应用程序是消息驱动的(或者说是事件驱动的),所以弄清 Windows 消息是如何作用的非常重要。

## 1.6 匈牙利法则

匈牙利法则是一种 Microsoft 程序员命名变量的规则。它以 Microsoft 程序员 Charles Symony 的原籍命名。Microsoft 在开发应用程序和系统程序时大量采用这一法则。

该法则在变量名前加一个小写字母前缀表示变量的类型。其余部分表示变量的作用。例如,

nCharacterCounter

表示变量是一个整型变量(n 表示整型),是一个字符计数器(Character Counter)。

表 1.1 列出了 Microsoft 通常命名变量和参数的几种前缀。

表 1.1 Windows 变量的几种通用前缀

| <i>Prefix</i> | <i>Meaning</i>                                                 |
|---------------|----------------------------------------------------------------|
| b             | Boolean(nonzero is true,0 is false)                            |
| c             | Character(one byte value)                                      |
| dw            | Long 32-bit unsigned integer                                   |
| f             | Bit flags pscked into a 16-bit integer                         |
| h             | 16-bit handle                                                  |
| l             | Long 32-bit integer                                            |
| lp            | Long 32-bit pointer                                            |
| n             | Short 16-bit integer                                           |
| p             | Short 16-bit pointer                                           |
| pt            | Coordinate pair(x and y)packed into an unsigned 32-bit integer |
| rgb           | RGB color value packed into a 32-bit integer                   |
| w             | Short 16-bit unsigned integer                                  |

## 1.7 句柄

句柄(handle)是个基本概念,所以必须弄明白它的含义。句柄是一个整数值,Windows 用它来识别应用程序中的客体如窗口、菜单、内存、输出设备、控制或文件。例如,菜单资源中的一个菜单项与一个句柄相对应。第一个菜单的第一项可以定义句柄值为 100,第二项的句柄

值可以定义为 101。在程序中可以由句柄 100 和 101 调用这些菜单项。Windows 通常只能访问句柄,而不能直接访问句柄所代表的数据。Windows 控制对数据的访问,因此在多任务的环境中就可以保护数据不受破坏。

## 1.8 Include 文件

Microsoft 的软件开发工具箱(SDK)提供 Windows.h 及其它几个专用的 include 文件,因而用户可以调用各个 Windows 函数和消息。Windows.h 和其它专用的 Include 文件中含有 Windows 的常量、变量、数据结构以及函数定义。所有的 Windows 应用程序必须包括 Windows.h,这样才能在自己的原程序中调用 Windows 函数及消息。

## 第二章 Windows 程序设计基础

Windows 编程对初学者来说比较困难。程序员对 Windows 存在不少的误解。如 Windows 程序和 C 程序差不多、熟悉 Windows 函数就可以了、甚至 Windows 与 C 是完全一样的！

为了澄清这些误解，本章开始介绍 Windows 编程的基本概念。

### 2.1 事件驱动

大多数 DOS 程序是顺序的，过程驱动形式。它有唯一的开始，唯一的过程，唯一的结果。因此程序直接控制程序事件和过程的序列。

Windows 程序与 DOS 程序相反，是事件驱动的。程序由事件的发生来控制，而不是事件的序列。下面的例子将说明了其基本区别。

假设要计算一个产品三次评比的平均分，过程如下：

1. 输入产品的名称
2. 输入第一次评比的得分
3. 输入第二次评比的得分
4. 输入第三次评比的得分
5. 计算并显示平均值

在过程驱动的程序中，流程图类似于图 2.1，程序顺序提示输入产品名称、第一次、第二次、第三次评比结果，最后显示出平均成绩。这一方法很合逻辑并遵循一种顺序结构。但是必须遵循设计好的过程序列。不能有什么例外，例如在第五步时要修改第二部的成绩。

虽然有许多方法可以处理序列中的意外情况，但其处理意外情况时也必须遵循顺序的，过程驱动的结构。事件驱动的程序设计可以避免顺序的，过程驱动方法处理非顺序情况时带来的局限性的。

事件驱动的程序在各个事件间循环并处理消息。一个消息就是一个事件发生的信号。Windows 程序员的任务就是管理应用程序发送或接收的消息。消息的出现顺序不是事先定义好的。

图 2.2 显示了该程序用事件驱动结构实现。过程驱动的全部功能都保留了下来，但用户不需要按特定的顺序输入。例如，可以直接输入第三次评比的结果而不输入前两次，这给用户带来极大的方便。

### 2.2 WinMain 函数

WinMain 函数是 Windows 应用程序的起始点。有三个基本部分：过程说明、程序初始化和消息循环。基本格式如下：