

应用程序“右顺序对齐菜单” 编程方法

丁健* 周游 (南京解放军理工大学工程兵工程学院 210007)

江南 (南京中科院南京地理与湖泊研究所 210008)

张万军 (芜湖总装工程兵驻蚌埠地区军代室 233000)

1 问题的提出

当今流行的编程语言 Visual Basic、Visual C++、Delphi 等均支持很方便的应用程序菜单设计,通常,编译后应用程序菜单显示的默认效果是从菜单条最左向右依次排列。但是我们也常常见到有些应用程序菜单条的某些菜单显示在最右侧(为表达方便本文将显示在菜单条最右侧的菜单称为“右顺序对齐菜单”),如“WPS2000集成办公系统”有文件打开时的“定制界面”菜单(见图1)、Origin2.5的“Help”菜单等。“右顺序对齐菜单”在突出菜单、增强提示、显示特定信息等方面有其特殊的应用意义,我们在车辆定位相关课题中将车辆位置经纬度值实时显示在“右顺序对齐菜单”,非常醒目,同时避免了占用状态栏或窗体空间。本文以 VC 和 VB 环境为例介绍“右顺序对齐菜单”的编程方法。

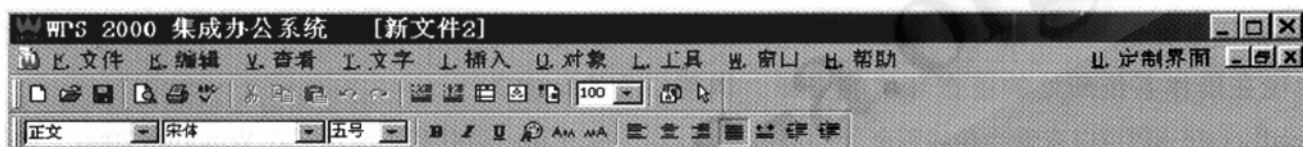


图1 WPS2000集成办公系统的右顺序对齐“定制界面”菜单

2 涉及的数据结构和 API 函数

2.1 涉及的数据结构

“右顺序对齐菜单”实现的关键是在设计或后续应用程序运行过程中改变菜单的信息状况,这种信息状况包括菜单的类型(图形式、字符式、分割条、右向左排列等)、菜单的状态(是否标记、是否使能等)、菜单的标识号、菜单的指针等等。Windows 环境下菜单信息存放在 MENUITEMINFO 数据结构中,数据结构 MENUITEMINFO 的定义如下:

```
typedef struct tagMENUITEM-INFO {
```

```
    UINT    cbSize;  
    UINT    fMask;  
    UINT    fType;  
    UINT    fState;  
    UINT    wID;  
    HMENU  hSubMenu;  
    HBITMAP hbmpChecked;  
    HBITMAP hbmpUnchecked;  
    DWORD  dwItemData;  
    LPTSTR  dwTypeData;
```

*丁健 南京中科院南京地理与湖泊研究所(中国科学院研究生院)在读博士研究生

```
UINT cch;
} MENUITEMINFO, FAR *LPMENUITEMINFO;
```

MENUITEMINFO 结构的详细解释读者可参考相关文献, 本文给出与“右顺序对齐菜单”实现密切相关的 fType 参数的解释。FType 存放菜单项的类型参数, 它可以是以下一个或几个预定义值:

```
MFT_BITMAP
MFT_MENUBARBREAK
MFT_MENUBREAK
MFT_OWNERDRAW
MFT_RADIOCHECK
MFT_RIGHTJUSTIFY //将菜单项及其随后的菜单
调整到菜单条的右侧排列
MFT_RIGHTORDER
MFT_SEPARATOR
MFT_STRING
```

2.2 涉及的 API 函数

Windows 提供了 ModifyMenu、SetCheck 等许多改变菜单标题、设置或取消标记状态的 API 函数, 但对于改变特定菜单使之右顺序对齐的目标只能通过 SetMenuItemInfo API 函数实现, 且须事先调用 GetMenuItemInfo 函数以取得该菜单的信息。事实上, SetMenuItemInfo 可以代替绝大部分菜单类 API 函数。

2.2.1 GetMenuItemInfo 函数简介

函数功能: 取得一个菜单项的信息。

函数原型: `BOOL GetMenuItemInfo(HMENU hMenu, UINT uItem, BOOL fByPosition, LPMENUITEMINFO lpMii);`

参数:

hMenu: 包含指定菜单项的菜单的句柄。

uItem: 将取得其信息的菜单项的标识符或位置。此参数的含义取决于参数 fByPosition 的值。

fByPosition: 此值用于指定参数 uItem 的含义。如果此

参数是 FALSE, 则 uItem 表示菜单项的标识符。否则, 表示菜单项的位置。

lpMii: 指向结构 MENUITEMINFO 的指针, 该结构指定要取得的信息并接收菜单项的信息。

返回值: 如果函数调用成功, 返回非零值; 如果函数调用失败, 返回值是零。

调用注意点: Windows CE 环境下, 由参数 lpMii 指向的 MENUITEMINFO 结构的 fMask 成员不能取 MIM_CHECKMARKS 标志。

2.2.2 SetMenuItemInfo 函数简介

函数功能: 该函数改动一个菜单项的信息。

函数原型: `BOOL SetMenuItemInfo(HMENU hMenu, UINT uItem, BOOL fByPosition, LPMENUITEMINFO lpMii);`

参数:

hMenu: 包含菜单项的菜单的句柄。

uItem: 将被修改的菜单项的标识符或位置。此参数的含义由参数 fByPosition 确定。

fByPosition: 用于指定参数 uItem 的含义的值。如果此参数值为 FALSE, 则参数 uItem 是菜单项的标识符。否则, 表示菜单项的位置。

lpMii: 指向结构 MENUITEMINFO 的指针, 该结构含有菜单项的信息, 并且指定将被修改的菜单项的属性。

返回值: 如果函数调用成功, 返回非零值; 如果函数调用失败, 返回值是零。

调用注意点: 为了使键盘加速键能控制位图或自己绘制的菜单项, 菜单的拥有者必须处理 WM_MENUCHAR 消息。参见自绘制菜单和 WM_MENUCHAR 消息。Windows CE 环境下, 只有下列值对参数 lpMii 指向的 MENUITEMINFO 结构中的 fMask 成员有效: MIIM_DATA; MIIM_ID; MIIM_TYPE。如果 MIIM_TYPE 被指定, 结构 MENUITEMINFO 的 fType 成员必须为菜单项的当前类型, 也就是说, 该类型不能被改变。

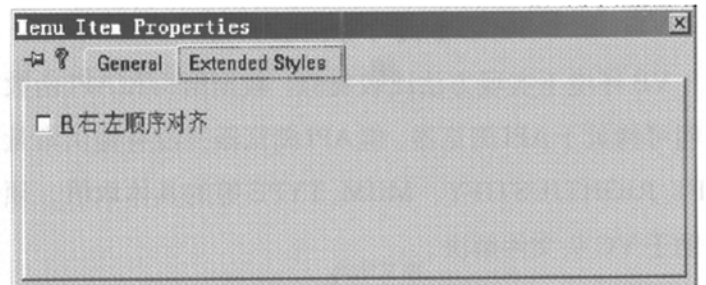
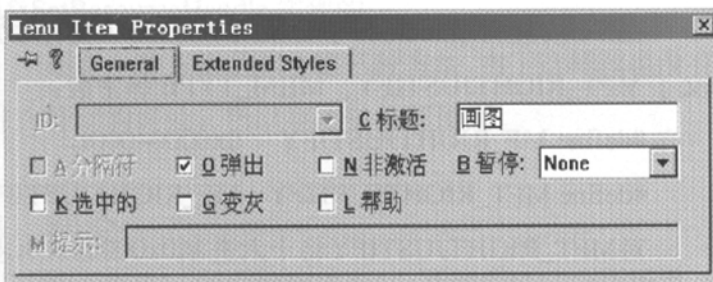


图 2 菜单属性对话框

2.3 程序流程

显然，如果能通过 SetMenuItemInfo 函数设置或改变一个菜单信息结构的 fType 参数为 MFT_RIGHTJUSTIFY，即可实现目标。

因此，程序流程为：

定义 MenuItemInfo 对象

->用 GetMenuItemInfo 取得指定菜单信息

->将信息中的 fType 部分设为 MFT_RIGHTJUSTIFY

->用 SetMenuItemInfo 重新设置菜单

3 Visual C++ 环境下实现的两种方法

3.1 在设计菜单时直接指定

VC IDE 支持可视化的菜单设计，选定菜单后双击或在右键菜单中选“属性”均能打开如图 2 的菜单属性对话框，选“Extended Styles”标签，标记“右-左顺序对齐”项即可。

3.2 编程实现

如将图 3 所示菜单条的“画图”菜单实现右顺序对齐，代码段为：

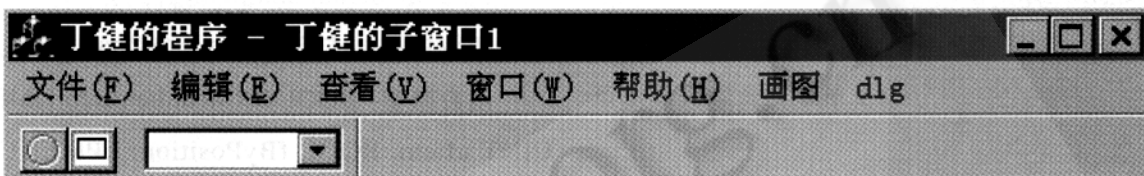


图 3 原始菜单

```

HMENU appmenu=AfxGetMainWnd()->GetMenu()->
m_hMenu; //取得应用程序菜单句柄
    MENUITEMINFO menuItemInfo; //定义
MENUITEMINFO 数据结构对象
    menuItemInfo.cbSize =sizeof(menuItemInfo); //
MENUITEMINFO 数据结构的大小以字节为单位
    menuItemInfo.fMask = MIIM_TYPE; //取得或设置结构的 fType 和 dwTypeData 数据成员
    menuItemInfo.cch = 128; //菜单标题字符串的长度
    GetMenuItemInfo(appmenu,5, TRUE,
&menuItemInfo);
    
```

```

//取得“画图”菜单的信息并将其存放在menuItemInfo
变量中
//菜单位置以 0 为基数从左开始算起
    menuItemInfo.fType =MFT_RIGHTJUSTIFY;
//更改 menuItemInfo 结构变量的 fType 参数为
MFT_RIGHTJUSTIFY
//其他参数内容保持不变
    SetMenuItemInfo(appmenu,5, TRUE,
&menuItemInfo); //设置菜单信息
    DrawMenuBar(); //重画菜单，实现目标
最后效果如图 4。
    
```

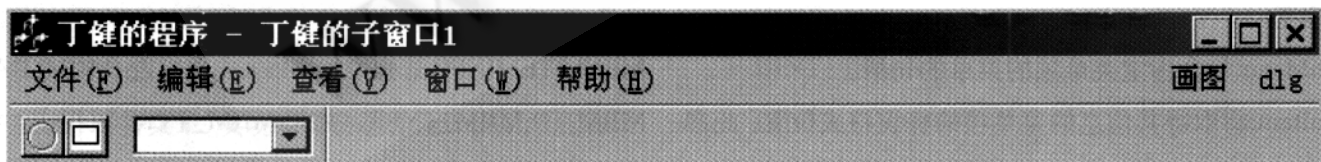


图 4 “画图”右顺序对齐菜单

4 Visual Basic 环境下实现方法

VB 环境下实现方法比较类似，数据结构和 API 函数声明可拷贝于 API 浏览器，但 API 浏览器并没有提供常量 MFT_RIGHTJUSTIFY、MIIM_TYPE 等的具体取值，须借助于 VC 头文件解决。

4.1 VB 程序员 MFT_RIGHTJUSTIFY 等常量值的获取

在 VC 安装目录 include 子目录的 Winuser.h 文件中定义了 MFT_RIGHTJUSTIFY、MIIM_TYPE 等常数：

```

#define MF_RIGHTJUSTIFY 0x00004000L
#define MFT_RIGHTJUSTIFY MF_RIGHTJUSTIFY
    
```

对 MFT_RIGHTJUSTIFY 将十六进制 0x00004000L 换算成十进制 16384 后即可在 VB 程序中直接使用，其他常量也须查找、换算。

4.2 关键源代码

```

' 数据结构声明
Public Type menuItemInfo
    cbSize As Long
    fMask As Long
    fType As Long
    fState As Long
    wID As Long
    hSubMenu As Long
    hbmpChecked As Long
    hbmpUnchecked As Long
    dwItemData As Long
    dwTypeData As String
    cch As Long
End Type

' API 函数声明
Public Declare Function
SetMenuItemInfo Lib "user32" Alias
"SetMenuItemInfoA" (ByVal hMenu As Long, ByVal un
As Long, ByVal bool As Boolean,
lpMenuItemInfo As menuItemInfo) As Long

Public Declare Function
GetMenuItemInfo Lib "user32" Alias
"GetMenuItemInfoA" (ByVal hMenu As
Long, ByVal un As Long, ByVal b As
Long, lpMenuItemInfo As
menuItemInfo) As Long

Public Declare Function
DrawMenuBar Lib "user32" (ByVal
hwnd As Long) As Long

Public Declare Function GetMenu
Lib "user32" (ByVal hwnd As Long) As
Long

' 在窗体载入过程（也可放在其
他过程）对菜单进行更改
Private Sub Form_Load()
    Dim my_menuItemInfo As
menuItemInfo
    Dim return_value As Long
    my_menuItemInfo.cbSize = 44
    my_menuItemInfo.fMask = 16
    Public Declare Function 'MIIM_TYPE
    my_menuItemInfo.cch = 128
    my_menuItemInfo.dwTypeData
= Space$(128)
    return_value =
GetMenuItemInfo(GetMenu(Me.hwnd),
6, 1, my_menuItemInfo)
    my_menuItemInfo.fType =
16384 'MFT_RIGHTJUSTIFY
    return_value = SetMenuItemInfo
(GetMenu(Me.hwnd), 6, 1,
my_menuItemInfo)
    DrawMenuBar GetMenu(Me.
hwnd)
End Sub

```

图 5 为 VB 环境下编译的应用程序右顺序对齐菜单实现效果。■

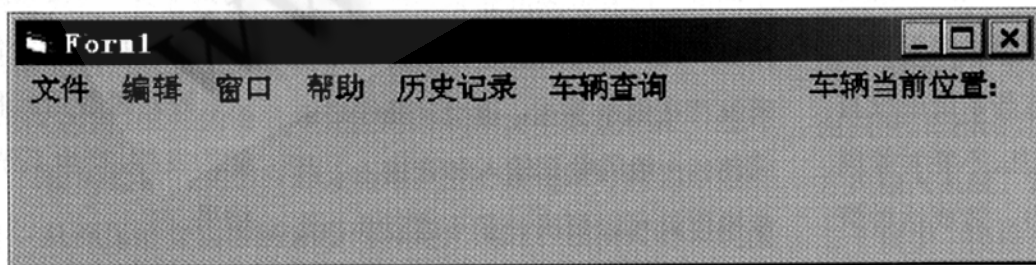


图 5 VB 编译产生应用程序实现效果