第四课　菜单

一、菜单的分类

1. 顶层菜单：窗口标题栏下方的菜单。

2. 弹出式菜单：单击右键弹出的菜单，以及下拉菜单。

3. 系统菜单：单击窗口左上角图标弹出的菜单。

二、菜单句柄和菜单标识

1. 菜单对象(也是一种窗口对象)通过菜单句柄——HMENU——访问。

2. 菜单中的每个菜单项通过其唯一标识——菜单ID——访问。

三、顶层菜单

1. 创建菜单

```
HMENU CreateMenu (void);
```

成功返回菜单句柄，否则返回NULL。

该函数所创建的是一个空菜单，不包含任何菜单项。

2. 添加菜单项

1) 追加菜单项

```
BOOL AppendMenu (
    HMENU   hMenu,       // 菜单句柄
    UINT    uFlags,      // 菜单项标志
    UINT    uIDNewItem, // 菜单项ID
    LPCTSTR lpNewItem   // 菜单项名称
);
```

成功返回TRUE，失败返回FALSE。

2) 插入菜单项

```
BOOL InsertMenu (
    HMENU   hMenu,       // 菜单句柄
    UINT    uPosition,  // 在此菜单项(ID或位置)之前插入
    UINT    uFlags,      // 菜单项标志
    UINT    uIDNewItem, // 菜单项ID
    LPCTSTR lpNewItem   // 菜单项名称
);
```

成功返回TRUE，失败返回FALSE。

3) 菜单项标志

```
MF_STRING     - 点击后发出WM_COMMAND消息，
                通过uIDNewItem参数的值区分具体哪个菜单项被点击了
MF_POPUP      - 可弹出子菜单，uIDNewItem参数取子菜单的句柄
MF_SEPARATOR  - 分割线，忽略uIDNewItem和lpNewItem参数
MF_CHECKED    - 勾选
MF_UNCHECKED  - 撤选
MF_ENABLED    - 可用
MF_DISABLED   - 禁用但不置灰
MF_GRAYED     - 禁用同时置灰
MF_BITMAP     - 位图
MF_OWNERDRAW  - 自绘
MF_BYCOMMAND  - uPosition参数为菜单项ID
MF_BYPOSITION - uPosition参数为菜单项位置
```

3. 将菜单安装到窗口上

```
BOOL SetMenu (
    HWND hWnd,  // 窗口句柄
    HMENU hMenu // 菜单句柄
);
```

成功返回TRUE，失败返回FALSE。

4. 菜单命令处理

```
WM_COMMAND - 命令消息
    wParam - HIWORD通知码(控件)/0(菜单)/1(加速键),
             LOWORD控件/菜单项/加速键的ID
    lParam - 句柄(控件)/NULL(其它)
```

5. 菜单项的状态

1) 在增加菜单项的时候设置初始状态

```
AppendMenu/InsertMenu (..., MF_STRING | MF_CHECKED, ...);
```

2) 动态改变菜单项的状态

A. 改变菜单项的勾选状态：

```
DWORD CheckMenuItem (
    HMENU hmenu,         // 菜单句柄
    UINT  uIDCheckItem, // 菜单项ID或位置
    UINT  uCheck         // 勾选/撤选
);
```

成功返回菜单项原勾选状态(MF_CHECKED/MF_UNCHECKED),
失败返回0xFFFFFFFF。

uCheck取值：
    MF_CHECKED   - 勾选
    MF_UNCHECKED - 撤选
    MF_BYCOMMAND - uIDCheckItem参数为菜单项ID
    MF_BYPOSITION - uIDCheckItem参数为菜单项位置

B. 改变菜单项的可用状态

```
DWORD EnableMenuItem (
    HMENU hMenu,          // 菜单句柄
    UINT  uIDEnableItem, // 菜单项ID或位置
    UINT  uEnable         // 可用/禁用
);
```

成功返回菜单项原可用状态(MF_ENABLED/MF_DISABLED/MF_GRAYED)，
失败返回0xFFFFFFFF。

uEnable取值：
    MF_ENABLED   - 可用
    MF_DISABLED   - 禁用但不置灰
    MF_GRAYED   - 禁用同时置灰
    MF_BYCOMMAND - uIDEnableItem参数为菜单项ID
    MF_BYPOSITION - uIDEnableItem参数为菜单项位置

6. 菜单的弹出初始化

```
WM_INITMENUPOPUP - 弹出菜单被激活但尚未显示的瞬间，
                   窗口过程函数收到此消息。
                   可在此消息处理中动态调整弹出菜单中菜单项的状态
          wParam - 弹出菜单句柄
          lParam - LOWORD弹出菜单在上层菜单中的索引，
                   HIWORD弹出菜单是否是窗口菜单
                   只有顶层菜单和系统菜单是窗口菜单
```

范例：WinMenu

四、系统菜单
---

1. 恢复/获取系统菜单

```
HMENU GetSystemMenu (
    HWND hWnd,    // 窗口句柄
    BOOL bRevert // TRUE恢复系统菜单，FALSE获取系统菜单
);
```

若bRevert为TRUE则返回NULL，否则返回系统菜单句柄。

2. 修改系统菜单

1) 追加/插入菜单项

```
AppendMenu/InsertMenu (...);
```

2) 删除菜单项

```
BOOL DeleteMenu (
    HMENU hMenu,      // 菜单句柄
    UINT  uPosition, // 菜单项ID或位置
    UINT  uFlags      // 按ID还是按位置删除菜单项
);
```

uFlags取值：
    MF_BYCOMMAND - uPosition参数为菜单项ID
    MF_BYPOSITION - uPosition参数为菜单项位置

3. 处理系统菜单

```
WM_SYSCOMMAND - 点击系统菜单触发此消息
       wParam - 具体命令，如SC_CLOSE(关闭)等
       lParam - 鼠标位置。LOWORD宏取低字，水平位置，
                HIWORD宏取高字，垂直位置
```

范例：WinSys

五、右键菜单(上下文菜单)
---

1. 创建菜单

```
HMENU CreatePopupMenu (void);
```

成功返回菜单句柄，否则返回NULL。

2. 显示菜单

```
BOOL TrackPopupMenu (
    HMENU       hMenu,      // 菜单句柄
    UINT        uFlags,     // 显示方式
    int         x,          // 显示位置水平坐标 \
                                                > 屏幕坐标
    int         y,          // 显示位置垂直坐标 /
    int         nReserved, // 保留，填0
    HWND        hWnd,       // 处理菜单消息的窗口句柄
    const RECT* prcRect     // 忽略
);
```

成功返回TRUE，失败返回FALSE。

uFlags取值：
```
    TPM_LEFTALIGN    \
    TPM_CENTERALIGN   > 菜单弹出时鼠标光标在菜单的左/中/右侧
    TPM_RIGHTALIGN   /
    TPM_TOPALIGN     \
    TPM_VCENTERALIGN  > 菜单弹出时鼠标光标在菜单的上/中/下端
    TPM_BOTTOMALIGN  /
```

```
    TPM_RETURNCMD      - 函数返回所选菜单项ID或0(取消菜单),
                         此时不触发菜单消息
    TPM_LEFTBUTTON     - 只能通过鼠标左键选择菜单项(缺省)
    TPM_RIGHTBUTTON    - 可通过鼠标左右键选择菜单项
```

3. 在收到WM_RBUTTONUP消息时弹出右键菜单

1) 客户区坐标转换到屏幕坐标

```
BOOL ClientToScreen (
    HWND    hWnd,   // 窗口句柄
    LPPOINT lpPoint // 点结构体(输入输出参数)
);
```

成功返回TRUE，失败返回FALSE。

2) 屏幕坐标转换到客户区坐标

```
BOOL ScreenToClient (
    HWND    hWnd,   // 窗口句柄
    LPPOINT lpPoint // 点结构体(输入输出参数)
);
```

成功返回TRUE，失败返回FALSE。

4. 在收到WM_CONTEXTMENU消息时弹出右键菜单

```
WM_CONTEXTMENU - 在WM_RBUTTONUP消息之后产生
        wParam - 被右键点击的窗口句柄
        lParam - LOWORD点击位置水平坐标,
                 HIWORD点击位置垂直坐标,
                 均为屏幕坐标
```

5. 右键菜单的弹出初始化

```
WM_INITMENUPOPUP - 弹出菜单被激活但尚未显示的瞬间,
                   窗口过程函数收到此消息。
                   可在此消息处理中动态调整弹出菜单中菜单项的状态
          wParam - 弹出菜单句柄
          lParam - LOWORD弹出菜单在上层菜单中的索引,
                   HIWORD弹出菜单是否是窗口菜单
                   只有顶层菜单和系统菜单是窗口菜单
```

范例：WinPopup

六、菜单资源

1. 资源

1) 资源编辑器：可视化界面开发工具

2) 资源脚本文件：.rc

3) 资源编译器：rc.exe

4) 资源文件：.res

2. 菜单也是一种资源

1) 添加菜单资源：Insert/Resource.../Menu

2) 编辑菜单资源

3) 加载菜单资源

A. 注册窗口类时加载菜单资源

```
typedef struct {
    ...
    LPCTSTR lpszMenuName; // 菜单资源名
    ...
}   WNDCLASSEX, *PWNDCLASSEX;
```

```
#include "resource.h"

WNDCLASSEX wcex = {0};
...
wcex.lpszMenuName = MAKEINTRESOURCE (IDR_MENU_MAIN);
...
```

根据此类创建的所有的窗口都使用此菜单。

B. 创建窗口时加载菜单资源

```
HWND CreateWindowEx (
    ...
    HMENU hMenu, // 菜单句柄
    ...
);
```

加载菜单资源：

```
HMENU LoadMenu (
    HINSTANCE hInstance, // 应用程序实例句柄
    LPCTSTR   lpMenuName // 菜单资源名
);
```

成功返回菜单句柄，失败返回NULL。

```
HWND hwndMain = CreateWindowEx (
    ...
    LoadMenu (g_hInstance, MAKEINTRESOURCE (IDR_MENU_MAIN)),
    ...
);
```

仅这个窗口都使用此菜单。

C. 加载右键菜单资源

```
case WM_CONTEXTMENU:
    ...
    HMENU hmenuPopup = LoadMenu (g_hInstance,
        MAKEINTRESOURCE (IDR_MENU_POPUP));
    HMENU hmenuSub = GetSubMenu (hmenuPopup, 0);
    TrackPopupMenu (hmenuSub,
        TPM_LEFTALIGN | TPM_TOPALIGN | TPM_RIGHTBUTTON, ...);
    ...
```

D. 动态切换菜单资源

```
case IDM_FILE_NEW:
    ...
    SetMenu (hWnd, LoadMenu (g_hInstance,
        MAKEINTRESOURCE (IDR_MENU_EDIT)));
    ...
case IDM_FILE_OPEN:
    ...
    SetMenu (hWnd, LoadMenu (g_hInstance,
        MAKEINTRESOURCE (IDR_MENU_EDIT)));
    ...
case IDM_FILE_CLOSE:
    ...
    SetMenu (hWnd, LoadMenu (g_hInstance,
        MAKEINTRESOURCE (IDR_MENU_MAIN)));
    ...
```

范例：WinRes