## СООБЩЕНИЯ МЕНЮ ПРИЛОЖЕНИЙ WINDOWS

При смене подсвеченного элемента меню (если, к примеру, пользователь "пробегает" по элементам меню с помощью клавиш со стрелками вверх и вниз) в оконную процедуру посылается сообщение WM\_MENUSELECT. Это сообщение посылают все элементы меню.

Когда же пользователь производит выбор (нажимает клавишу "Enter", к примеру), сообщение WM\_COMMAND оконной процедуре посылают только обычные элементы меню. Запрещенные и "серые" элементы меню в этом случае никаких сообщений не посылают.

В элементах wParam и IParam посылаемых сообщений хранится информация, достаточная для того, чтобы программа смогла определить, какие действия ей необходимо выполнить случае выбора пользователем того или иного элемента меню.

Помимо обычного меню у окна в большинстве случаев есть еще и системное меню. Сказанное относится и к системному меню. Отличие между обычным меню и системным состоит в том, что оконной процедуре посылаются сообщения WM\_SYSMENUSELECT и WM\_SYSCOMMAND. Кроме этого, сообщения WM\_SYSCOMMAND оконная процедура получает и в случае нажатия кнопок минимизации, максимизации и закрытия окна, которые находятся не в системном меню, а в правом углу заголовка окна.

Параметры сообщения WM\_MENUSELECT. В младшем слове wParam оконная процедура получает сведения о том, какой элемент стал подсвеченным. Если учесть, что макросы LOWORD () и HIWORD () выделяют соответственно младшее и старшее слово 32-битного аргумента, и назвать источник сообщения uItem, то можно записать:

uItem = (UINT) LOWORD (wParam);

В зависимости от обстоятельств смысл uItem различается:

если подсвеченный элемент является конечным и не влечет за собой вызов popup-меню, то ultem содержит идентификатор элемента меню;

если подсвеченный элемент при выборе влечет за собой вызов popup-меню, то ultem содержит номер (индекс) этого элемента в том меню, в котором оно находится;

В старшем слове wParam содержатся характеристики подсвеченного элемента. Аналогично предыдущему,

fuFlags = (UINT) HIWORD (wParam).

Возможные значения fuFlags характеризуются идентификаторами, начинающимися с MF\_ (табл.1).

Таблица 1

Характеристики подсвеченного элемента меню

|  |  |
| --- | --- |
| Флаг | Описание |
| MF\_BITMAP | Вместо строки в качестве элемента меню применяется bitmap |
| MF\_CHECKED | Элемент отмечаемый (со "значком") |
| MF\_DISABLED | Элемент запрещен |
| MF\_GRAYED | Элемент запрещен и отображается серым цветом |
| MF\_HILITE | Элемент подсвечен |
| MF\_MOUSESELECT | Элемент выбран мышью |
| MF\_OWNERDRAW | За прорисовку элемента отвечает не система, а программа |
| MF\_POPUP | Элемент вызывает появление рорир-меню более низкого уровня |
| MF\_SYSMENU | Элемент из системного меню |

lParam содержит в себе хэндл того меню, которому принадлежит подсвеченный элемент. Обозначив хэндл меню как hMenu, получим:

hMenu = (HMENU) lParam;

Параметры сообщения WM\_COMMAND. Как и в случае с WM\_SELECTMENU, младшее слово wParam содержит сведения об источнике сообщения. Так как сообщение WM\_COMMAND посылается только конечными элементами меню, то в младшем слове wParam содержится идентификатор выбранного элемента меню:

wID = LOWORD (wParam);

Старшее слово wParam указывает, от какого управляющего элемента пришло сообщение. Если сообщение пришло от меню, то это слово равно нулю, т.е.

wNotifyCode = HIWORD (wParam) = 0;

lParam в случае сообщения от меню ВСЕГДА равно NULL!

Параметры сообщения WM\_INITMENU. Сообщение WM\_INITMENU поступает перед отображением главного меню.

Параметр wParam равен дескриптору меню. Если сообщение WM\_INITMENU обрабатывают, то возвращают 0. Обработка обычно сводится к изменению состояния элементов меню

Параметры сообщения WM\_INITMENUPOPUP.

Это сообщение поступает перед отображением временного меню.

Параметр wParam равен дескриптору меню.

Младшее слово параметра lParam равно позиции этого меню в меню верхнего уровня:

uItem = (UINT) LOWORD (lParam);

Старшее слово lParam равно 1 для системного меню и 0 - для обычного.

Если это сообщение обрабатывают, то возвращают 0. Обработка обычно сводится к изменению состояния элементов меню.

Итак, меню имеет строгую древовидную структуру, которая начинается с меню первого уровня (оно обычно называется главным меню программы или menubar'ом и располагается сразу под заголовком окна). К этому меню первого уровня могут быть присоединены как конечные элементы меню, так и элементы, выбор которых приводит к появлению так называемых всплывающих (popup) меню, к которым, в свою очередь, присоединяются элементы очередного уровня и т.д. Перед началом создания меню вся его структура должна быть тщательно продумана. Неплохо, если бы программист имел перед глазами графическое представление этого меню.

Для создания меню необходимо выполнить следующие действия:

Выбрать подменю самого низкого уровня, которые содержат только конечные элементы меню, и создать их с помощью функций CreateMenu () или CreatePopupMenu () в зависимости от потребностей. Эти функции возвращают хэндл созданного меню. Меню создается пустым.

Посредством функции AppendMenu () добавляем в них требуемые элементы.

Создаем меню следующего, более высокого уровня, и добавляем в них требуемые элементы и меню, созданные нами на предыдущем шаге.

Повторяем пункты 1 - 3 до тех пор, пока создание всех подменю не будет закончено.

Создаем главное меню программы посредством использования функции CreateMenu ().

Присоединяем созданные подменю самого высокого уровня к главному меню программы с помощью функции AppendMenu ().

Присоединяем меню к окну посредством использования функции SetMenu ().

Прорисовываем меню с помощью функции DrawMenuBar ().

Если в ходе программы сложилась такая ситуация, что меню оказалось не присоединенным к окну, перед выходом из программы обязательно уничтожаем его, вызывая функцию DestroyMenu () (присоединенное к окну меню уничтожается автоматически при уничтожении окна).

Формат названных API-функций выглядит следующим образом:

HMENU CreateMenu ()

Параметры: не имеются.

Возвращаемое значение: хэндл вновь созданного меню, иначе - NULL.

HMENU CreatePopupMenu ()

Параметры: не имеются.

Возвращаемое значение: хэндл вновь созданного всплывающего меню, иначе - NULL.

BOOL AppendMenu (HMENU hMenu, UINT uFlags, UINT idNewItem, LPCTSTR lpszNewItem)

Параметры:

hMenu - хэндл того меню, в которое добавляется новый пункт.

uFlags - комбинация битовых флагов, определяющих внешний вид и поведение добавляемого меню. Начинаются с префикса MF\_.

idNewItem - идентификатор для нового пункта меню либо хэндл добавляемого всплывающего меню (зависит от флага MF\_POPUP, того, установленного во втором параметре).

lpszNewItem - содержимое нового пункта меню. Зависит от установленных флагов uFlags. Установлен флаг MF\_BITMAP - параметр содержит хэндл bitmap'а. Установлен флаг MF\_STRING - параметр содержит указатель на строку символов и т.д.

Возвращаемое значение: при успешном выполнении - TRUE, иначе - FALSE.

BOOL SetMenu (HWND hWnd, HMENU hMenu)

Параметры:

hWnd - хэндл окна, для которого нужно закрепить меню.

hMenu - хэндл добавляемого меню.

Возвращаемое значение: при успешном выполнении - TRUE, иначе - FALSE. BOOL DrawMenuBar (HWND hWnd)

Параметры:

hWnd - хэндл окна, которое имеет меню.

Возвращаемое значение: при успешном выполнении - TRUE, иначе - FALSE.



Например. Создадим окно, которое содержит основное меню (рис.1). Основное меню окна состоит из двух всплывающих меню: "File" и "Help". В первом всплывающем меню находятся два элемента, "Enable exit" и "Exit", во втором - один элемент, "About", который остается запрещенным в течение всего периода существования окна. Кроме этого, элемент "Exit" при запуске объявляется "серым", т.е. из программы можно выйти только через системное меню. Однако в случае выбора элемента "Enable exit" "Exit" становится обычным, а вместо "Enable exit" возникает "Disable exit" и наоборот.

Тогда в функцию WinMain () необходимо добавить:

const IDM\_Enable\_Disable = 0;

const IDM\_Exit = 1;

const IDM\_About = 2;

……

HMENU hMenu,hFileMenu,hHelpMenu;

hFileMenu=CreatePopupMenu ();

AppendMenu (hFileMenu,MF\_ENABLED | MF\_STRING, IDM\_Enable\_Disable, "&Enable exit");

AppendMenu (hFileMenu, MF\_GRAYED | MF\_STRING, DM\_Exit,"E&xit");

hHelpMenu=CreatePopupMenu ();

AppendMenu (hHelpMenu,MF\_DISABLED | MF\_STRING, IDM\_About, "&About");

hMenu = CreateMenu ();

AppendMenu (hMenu, MF\_ENABLED | MF\_POPUP, (UINT) hFileMenu, "&File");

AppendMenu (hMenu, MF\_ENABLED | MF\_POPUP, (UINT) hHelpMenu,"&Help");

SetMenu (hWnd, hMenu);

/\*Прорисовка окна и меню\*/

ShowWindow (hWnd, nCmdShow);

UpdateWindow (hWnd);

DrawMenuBar (hWnd);

……. .

/\*Добавляемый фрагмент в WndProc \*/

static UINT nFlag = MF\_ENABLED;

char\* pContent [] =

{

"Enable exit",

"Disable exit"

};

static UINT nIndex = 0;

switch (Message)

{

case WM\_COMMAND:

switch (wParam)

{

case IDM\_Enable\_Disable:

EnableMenuItem (hFileMenu, IDM\_Exit, MF\_BYCOMMAND | nFlag);

nFlag = (nFlag == MF\_ENABLED)? MF\_GRAYED: MF\_ENABLED;

nIndex = (nIndex == 0)? 1: 0;

ModifyMenu (hFileMenu, IDM\_Enable\_Disable, MF\_BYCOMMAND | MF\_STRING, IDM\_Enable\_Disable, pContent [nIndex]);

break;

case IDM\_Exit:

DestroyWindow (hWnd);

break;

}

….

}

Как видно из примера, при получении WM\_COMMAND, младшее слово wParam которого равно равно IDM\_Enable\_Disable, производятся следующие действия:

с помощью функции EnableMenuItem () запрещается или делается доступным элемент "Exit";

с помощью функции ModifyMenu () изменяется текст элемента, выбор которого приводит к состоянию элемента "Exit".

Функция EnableMenuItem () позволяет изменять состояние элемента (разрешенный, запрещенный, "серый") меню по своему усмотрению. Синтаксис функции:

BOOL EnableMenuItem (HMENU hMenu, UINT uItem, UINT uFlags)

Параметры:

hMenu - хэндл меню, которому принадлежит пункт меню.

uItem - пункт меню, состояние которого изменяется.

uFlags - комбинация битовых флагов, определяющее новое состояние uItem пункта меню. Флаг MF\_BYCOMMAND показывает, значение представляет собой идентификатор пункта меню.

Возвращаемое значение: предыдущее состояние пункта меню (MF\_DISABLE, MF\_ENABLE, MF\_GRAYED), иначе - 0хffffffff (-1).

Функция ModifyMenu () После изменения состояния элемента "Exit" с разрешенного на серое и наоборот, необходимо изменить текст в элементе, от которого зависит это состояние. Это изменение производится посредством вызова функции ModifyMenu (), которой передаются пять аргументов:

BOOL ModifyMenu (HMENU hMenu, UINT uItem, UINT uFlags, UINT uNewItem, LPCTSTR lpszNewItem)

Параметры:

hMenu - хэндл меню, которому принадлежит пункт меню.

uItem - пункт меню, состояние которого изменяется.

uFlags - комбинация битовых флагов, во-первых, определяющих элемент, подлежащий изменению (MF\_BYCOMMAND или MF\_BYPOSITION), а во-вторых, определяющих состояние элемента после изменения.

uNewItem - указывает или идентификатор измененного элемента, или хэндл нового меню (если, конечно, в третьем аргументе установлен флаг MF\_POPUP).

lpszNewItem - содержимое нового измененного пункта меню. Зависит от установленных флагов uFlags. Установлен флаг MF\_BITMAP - параметр содержит хэндл bitmap'а. Установлен флаг MF\_STRING - параметр содержит указатель на строку символов и т.д.

Возвращаемое значение: при успешном выполнении - TRUE, иначе - FALSE.

Функция DeleteMenu () удаляет элемент из меню и освобождает все связанные с ним ресурсы. Если удаляемый элемент указывает на временное меню, то функция DeleteMenu удаляет и это временное меню. Синтаксис функции:

BOOL DeleteMenu (HMENU hMenu, UINT uPosition, UINT uFlags)

Параметры:

hMenu - хэндл меню, которому принадлежит пункт меню.

uPosition - определяет удаляемый элемент. Его значение связано со значением параметра uFlags.

uFlags - задает способ истолкования параметра uPosition пункта меню. Если uFlags=MF\_BYCOMMAND (значение по умолчанию), то значение uPosition равно идентификатору команды удаляемого элемента. Если же uFlags=MF\_BYPOSITION, то значение uPosition равно позиции удаляемого элемента в меню.

Возвращаемое значение: при успешном выполнении - TRUE, иначе - FALSE.

Функция RemoveMenu () удаляет элемент из меню, не разрушая связанные с ним ресурсы. Эти ресурсы можно использовать в дальнейшем. Синтаксис функции:

BOOL RemoveMenu (HMENU hMenu, UINT uPosition, UINT uFlags)

Параметры:

hMenu - хэндл меню, которому принадлежит пункт меню.

uPosition - определяет удаляемый элемент. Его значение связано со значением параметра uFlags.

uFlags - задает способ истолкования параметра uPosition пункта меню. Если uFlags=MF\_BYCOMMAND (значение по умолчанию), то значение uPosition равно идентификатору команды удаляемого элемента. Если же uFlags=MF\_BYPOSITION, то значение uPosition равно позиции удаляемого элемента в меню.

Возвращаемое значение: при успешном выполнении - TRUE, иначе - FALSE.

Для разрушения меню вызывают функцию DestroyMenu (). Эта функция разрушает заданное меню и освобождает ресурсы, которые меню занимает. Эту функцию окна вызывают для разрушения тех меню, которые они создали, но не подключили к себе. Подключенные Меню автоматически разрушаются при разрушении окна. Синтаксис функции:

BOOL DestroyMenu (HMENU hMenu)

Параметры:

hMenu - хэндл меню, которому принадлежит пункт меню.

Возвращаемое значение: при успешном выполнении - TRUE, иначе - FALSE

Функция GetMenu () возвращает дескриптор главного меню окна hWnd.

HMENU GetMenu (HWND hWnd)

Параметры:

hWnd - хэндл окна, которому принадлежит главное меню.

Возвращаемое значение: при успешном выполнении - дескриптор меню, иначе - NULL.

Функция GetSubMenu () возвращает дескриптор всплывающего меню, которое расположено в позиции nPos указанного меню hMenu. Причем первому временному меню в hMenu соответствует нулевое значение параметра nPos. Если функция GetSubMenu () вернула значение NULL, то hMenu в позиции nPos не содержит временное меню:

HMENU GetSubMenu (HMENU hMenu, int nPos)

Параметры:

hMenu - хэндл меню.

nPos - позиция требуемого временного меню.

Возвращаемое значение: при успешном выполнении - дескриптор меню, иначе - NULL.

Плавающее меню создают обычным способом, но не вставляют в другое меню. Для отображения и выбора строк этого меню вызывают функцию TrackPopupMenu ().

Функция TrackPopupMenu () предназначена для вывода на экран плавающего меню и создания своего собственного цикла обработки сообщений, завершающего работу после выбора строки. Она не возвращает управление до тех пор, пока работа с меню не будет завершена выбором строки или отказом от выбора. Синтаксис функции:

BOOL TrackPopupMenu (HMENU hMenu, UINT uFlags, int x, int y, int nReserved, HWND hwnd, CONST RECT \*prcRect)

Параметры:

hMenu - дескриптор отображаемого плавающего меню. Он может быть создан функцией CreatePopupMenu () или получен с помощью функции GetSubMenu ().

uFlags - комбинация флагов, которые задают функциональные параметры плавающего меню.

х - координата по горизонтали от левого края экрана.

у - координата по вертикали от верхнего края экрана.

nReserved - зарезервированный параметр, должен быть всегда равен нулю.

hwnd - дескриптор уже существующего окна-владельца, которое получит сообщения от меню. Сообщение WM\_COMMAND окно получит только после завершения работы функции TrackPopupMenu ().

prcRect - указатель на прямоугольную область, находясь в пределах которой можно работать с меню. Если сделать щелчок мышью за пределами этого прямоугольника, плавающее меню исчезнет. Если prcRect=NULL, то эта область ограничена прямоугольной рамкой плавающего меню.

Возвращаемое значение: в случае успешного выполнения функция возвращает ненулевое значение. Если в параметре uFlags задано TPM\_RETURNCMD, то возвращаемое значение равно идентификатору команды выбранной строки. Если элемент не выбран, возвращаемое значение - нуль.

Следующие константы uFlags задают способ размещения меню по горизонтали относительно параметра х:

| Константа | Пояснение |
| --- | --- |
| ТРМ\_CENTERALIGN | Центр меню по горизонтали совпадает с х |
| ТРМ\_LEFTALIGN | Левый край меню совпадает с х |
| TPM\_RIGHTALIGN | Правый край меню совпадает с х |

Следующие константы uFlags задают способ размещения меню по вертикали относительно параметра у:

|  |  |
| --- | --- |
| Константа | Пояснение |
| TPM\_BOTTOMALIGN | Нижний край меню совпадает с у |
| TPM\_TOPALIGN | Верхний край меню совпадает с у |
| TPM\_VCENTERALIGN | Центр меню по вертикали совпадает с у |

Следующие константы uFlags задают способ выбора строк меню без указания окна-владельца для меню:

|  |  |
| --- | --- |
| Константа | Пояснение |
| ТРМ\_NONOTIFY | Не посылать сообщения о выборе строки |
| TPM\_RETURNCMD | Возвращать идентификатор выбранной команды |

Следующие константы uFlags задают кнопку мыши, которую прослеживает плавающее меню:

|  |  |
| --- | --- |
| Константа | Пояснение |
| ТРМ\_LEFTBUTTON | Прослеживает левую кнопку мыши |
| TPM\_RIGHTBUTTON | Прослеживает правую кнопку мыши |