

Лекция 2. Управление окнами в Windows API

Архитектура ОС Windows

19 сентября 2013 г.

Простейший пример

Пример

```
#include <windows.h>

int APIENTRY WinMain(
    HINSTANCE hInstance, HINSTANCE hInstancePrev,
    LPSTR lpszCmdLine, int nCmdShow)
{
    MessageBox(
        NULL, TEXT("Hello World"), TEXT("My application"),
        MB_OK | MB_ICONINFORMATION);
    //
    return 0;
}
```

Макросы APIENTRY, MB_OK, MB_ICONINFORMATION

Определение макроса APIENTRY

```
#define _stdcall __attribute__((stdcall))  
// ...  
#define APIENTRY _stdcall
```

Макросы APIENTRY, MB_OK, MB_ICONINFORMATION

Определение макроса APIENTRY

```
#define _stdcall __attribute__((stdcall))  
// ...  
#define APIENTRY _stdcall
```

Определение макросов MB_OK и т. д.

```
#define MB_OK 0  
// ...  
#define MB_ICONINFORMATION 64
```

Макрос TEXT()

Определение макроса TEXT()

```
#ifndef UNICODE
# define __TEXT(q) L##q
#else
# define __TEXT(q) q
#endif
#define TEXT(q) __TEXT(q)
```

Пример

- TEXT("Hello World") → L"Hello World"
- TEXT("Hello World") →

Макрос TEXT()

Определение макроса TEXT()

```
#ifndef UNICODE
# define __TEXT(q) L##q
#else
# define __TEXT(q) q
#endif
#define TEXT(q) __TEXT(q)
```

Пример

- TEXT("Hello World") → L"Hello World"
- TEXT("Hello World") → "Hello World"

Пример

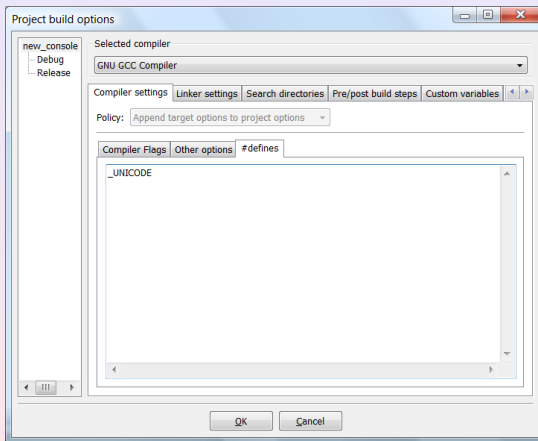


Рис. 1 : определение символа `_UNICODE`

Пример

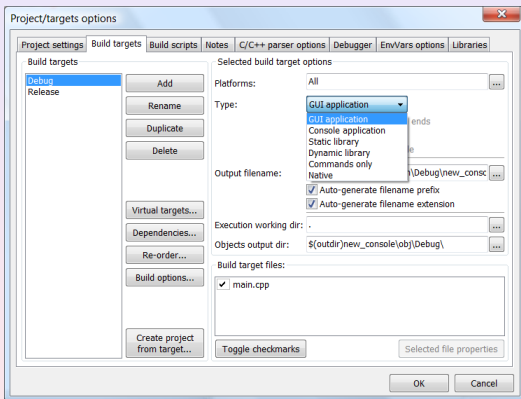


Рис. 2 : определение типа приложения (GUI)

Пример

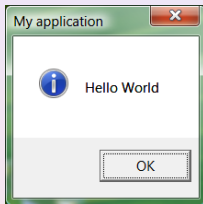


Рис. 3 : работа приложения

Функция MessageBox()

Определение функции MessageBox()

```
int WINAPI MessageBox(  
    __in_opt HWND    hWnd,  
    __in_opt LPCTSTR lpctszText,  
    __in_opt LPCTSTR lpctszCaption,  
    __in      UINT    uType  
);
```

Флаги и возвращаемые значения функции MessageBox()

MB_OK	MB_ICONINFORMATION	MB_DEFBUTTON1
MB_OKCANCEL	MB_ICONEXCLAMATION	MB_DEFBUTTON2
MB_YESNO	MB_ICONERROR	MB_DEFBUTTON3
MB_YESNOCANCEL
...		

Таблица 1 : флаги (uType)

IDOK	IDYES
IDCANCEL	IDNO
...	...

Таблица 2 : возвращаемые значения

Пример вопроса при помощи функции MessageBox()

Пример

```
const int cnAnswer = MessageBox(
    NULL, TEXT("Продолжить обработку?"), TEXT("My application"),
    MB_OKCANCEL | MB_ICONEXCLAMATION);
if (cnAnswer == IDOK)
{
    // Обработка...
}
```

Пример создания окна

Пример

```
#include <windows.h>

// Функция обратного вызова окна
LRESULT CALLBACK WindowProcedure(HWND, UINT, WPARAM, LPARAM);

// Имя класса окна
const TCHAR g_ctorszClassName[] = TEXT("CodeBlocksWindowsApp");

int APIENTRY WinMain(
    HINSTANCE hThisInstance, HINSTANCE hPrevInstance,
    LPSTR lpszArgument, int nCmdShow)
{
    // Класс окна
```

Пример создания окна (продолжение)

Пример (продолжение)

```
WNDCLASSEX wnd_class;  
wnd_class.cbSize = sizeof (WNDCLASSEX);  
wnd_class.hInstance = hThisInstance;  
wnd_class.lpszClassName = g_ctszClassName;  
wnd_class.lpfnWndProc = WindowProcedure;  
wnd_class.style = CS_DBLCLKS;  
wnd_class.hIcon = LoadIcon(NULL, IDI_APPLICATION);  
wnd_class.hIconSm = LoadIcon(NULL, IDI_APPLICATION);  
wnd_class.hCursor = LoadCursor(NULL, IDC_ARROW);  
wnd_class.lpszMenuName = NULL;  
wnd_class.cbClsExtra = 0;  
wnd_class.cbWndExtra = 0;  
wnd_class.hbrBackground = (HBRUSH) COLOR_BACKGROUND;
```

Пример создания окна (продолжение)

Пример (продолжение)

```
//  
// Регистрация класса окна  
//  
if (!RegisterClassEx(&wnd_class))  
    return -1;  
//  
// Создание окна  
//
```

Пример создания окна (продолжение)

Пример (продолжение)

```
HWND hWnd = CreateWindowEx(  
    0, // расширенный стиль окна  
    g_ctszClassName, // класс окна  
    TEXT("Code::Blocks Template Windows App"),  
    WS_OVERLAPPEDWINDOW, // стиль окна  
    CW_USEDEFAULT, // расположение по горизонтали  
    CW_USEDEFAULT, // расположение по вертикали  
    CW_USEDEFAULT, // ширина  
    CW_USEDEFAULT, // высота  
    HWND_DESKTOP, // родительское окно  
    NULL, // меню  
    hThisInstance, // экземпляра программы  
    NULL); // параметры функции окна
```


Пример создания окна (продолжение)

Пример (продолжение)

```
//  
// Показ окна  
//  
ShowWindow(hWnd, nCmdShow);  
//  
// Цикл обработки сообщений (до GetMessage() == 0)  
//  
MSG messages;
```

Пример создания окна (продолжение)

Пример (продолжение)

```
while (GetMessage(&messages, NULL, 0, 0))
{
    //
    // Преобразование сообщений с кодами виртуальных
    // клавиш в символные
    //
    TranslateMessage(&messages);
    //
    // Отправка сообщения в WindowProcedure()
    //
    DispatchMessage(&messages);
}
```

Пример создания окна (продолжение)

Пример (продолжение)

```
//  
// Вернуть системе значение, переданное PostQuitMessage()  
//  
return messages.wParam;  
//  
} // WinMain()  
  
//  
// Вызывается системой во время вызова DispatchMessage()  
//  
LRESULT CALLBACK WindowProcedure(  
    HWND hWnd, UINT uMessage, WPARAM wParam, LPARAM lParam)  
{
```

Пример создания окна (окончание)

Пример (окончание)

```
switch (uMessage)
{
    case WM_DESTROY:
        PostQuitMessage(0);           // отправить WM_QUIT в очередь
        break;
    default: // все остальные сообщения обрабатываются системой
        return DefWindowProc(hWnd, uMessage, wParam, lParam);
}
//
return 0;
//
} // WindowProcedure()
```

Пример описания ресурса меню

Пример (menu.rc)

```
APP_MENU MENU
BEGIN
  POPUP "&Файл"
  BEGIN
    MENUITEM SEPARATOR
    MENUITEM "&Выход",          1
  END
  POPUP "&Справка"
  BEGIN
    MENUITEM "&О программе...",  2
  END
END
```

Пример подключения меню к классу окна

Пример

```
//  
// Класс окна  
//  
WNDCLASSEX wnd_class;  
wnd_class.cbSize = sizeof (WNDCLASSEX);  
// ...  
wnd_class.lpszMenuName = TEXT("APP_MENU");  
// ...
```

Пример обработки событий меню

Пример

```
LRESULT CALLBACK WindowProcedure(/* ... */)
{
    switch (uMessage)
    {
        case WM_COMMAND:
            switch (wParam)
            {
                case 1:
                    MessageBox(
                        hWnd, TEXT("Выбран Файл -> Выход"),
                        g_ctszAppTitle, MB_OK);
                    break;
                // ...
            }
        }
}
```

Пример обработки событий меню

Пример

```
LRESULT CALLBACK WindowProcedure(  
    HWND hWnd, UINT uMessage, WPARAM wParam, LPARAM lParam)  
{  
    switch (uMessage)  
    {  
        case WM_COMMAND:  
            switch (wParam)  
            {  
                case 1:  
                    DestroyWindow(hWnd);  
                    break;  
                // ...  
            }  
        }  
    }
```


Основные оконные сообщения

WM_CREATE	WM_LBUTTONDOWN
WM_DESTROY	WM_MOUSEMOVE
WM_CLOSE	WM_LBUTTONUP
WM_PAINT	WM_RBUTTONDOWN
WM_SIZE	WM_RBUTTONUP
	WM_LBUTTONDBLCLK
	WM_RBUTTONDBLCLK

Таблица 3 : сообщения

Обработка запроса на закрытие окна

Пример

```
case WM_CLOSE:
{
    int nAnswer = MessageBox(
        hWnd,
        TEXT("Вы уверены, что хотите закрыть приложение?"),
        TEXT("Вопрос"), MB_YESNO | MB_ICONEXCLAMATION);
    if (nAnswer == IDYES)
        return DefWindowProc(hWnd, uMessage, wParam, lParam);
    break;
}
```

Вызов оконной процедуры

Пример

```
case WM_COMMAND:
    switch (wParam)
    {
        case 1:
            SendMessage(hWnd, WM_CLOSE, 0, 0);
            break;
        // ...
    }
    break;
```

Функции SendMessage() и PostMessage()

Определение SendMessage()

```
LRESULT WINAPI SendMessage(  
    __in    HWND    hWnd,  
    __in    UINT    uMsg,  
    __in    WPARAM  wParam,  
    __in    LPARAM  lParam  
);
```

Определение PostMessage()

```
BOOL WINAPI PostMessage(  
    __in_opt HWND    hWnd,  
    __in    UINT    uMsg,  
    __in    WPARAM  wParam,  
    __in    LPARAM  lParam  
);
```