ЛЕКЦИЯ №8

**1. API - ФУНКЦИИ WINDOWS УПРАВЛЕНИЯ  
 ВНЕШНИМИ ПРИЛОЖЕНИЯМИ**

**API-функции** - это набор стандартных функций операционной системы Windows, через которые можно программно управлять всеми запущенными программами и ресурсами компьютера. API-функции являются некоторым современным аналогом функциям-прерываниям **INT21h** MS-DOS. Возможность использовать данные функции в Делфи организуется через автоматически подключаемый модуль «Windows».

Для получения справки по API-функциям из Делфи необходимо выбрать пункт «Help > **Windows SDK** (Win32 Developer's References)». Данный хелп содержит подробную информацию о правилах вызова основных API- функций (синтаксис языка «С»).

Все визуальные компоненты Делфи, **например «Edit, Label, Button» и др. являются окнами,** как и форма. Для универсальной работы с любыми окнами используются дескрипторы (handle) окон. Дескриптор окна - это указатель на окно в виде целого числа (аналог порядкового номера), типа **«HWND» (прототип «Integer»).** При открытии любого окна (видимого или невидимого, типа формы или кнопки), Windows присваивает ему уникальный дескриптор. Данный указатель действителен до закрытия окна или перезагрузки Windows. Любому окну Windows может присвоить любой дескриптор.

Для того, чтобы узнать дескрипторы окон какой-либо программы, применяется программа «WinSight32» из пакета программ Delphi или другая подобная программа.

|  |  |  |
| --- | --- | --- |
|  | **ПРИМЕР ВЫЗОВА API-ФУНКЦИИ** | ПОЯСНЕНИЕ **(VAR H,H2:HWND;**  **I: LONGWORD; S: STRING;)** |
| 1. | GetWindowText(h,pchar(s),I); | Получение текста из окна «h» в переменную «s» длиной не более «i» символов |
| 2. | i:=GetWindowtextLength(h); | Получение длины текста из окна «h»  в переменную «i» |
| 3. | SetWindowText(h,pchar(s)); | Установка текста «s» в окне «h» |
|  |  |  |
| 4. | h:=FindWindow(pchar(s),nil); | Поиск дескриптора окна с именем класса «s» в переменную «h» |
| 5. | h2:=GetWindow(h,GW\_CHILD); | Поиск подчиненного окна «h2» для окна «h» |
| 6. | h2:=GetWindow(h,GW\_HWNDNEXT); | Поиск следующего подобного окна «h2»  для окна «h» |
| 7. | If h=0 then WinExec(‘calc.exe’, SW\_SHOWMODAL); | Если искомое окно не найдено, то запустить указанную программу «calc.exe» |
|  |  |  |
| 8. | ShowWindow(h,SW\_HIDE);  ShowWindow(h,SW\_MINIMIZE);  ShowWindow(h,SHOWNORMAL); | Установить для окна «h» заданный  режим отображения |
| 9. | SetForegroundWindow(h); | Поместить видимое на экране окно «h»  поверх всех окон |
|  |  |  |
| 10. | SendMessage(h,WM\_LBUTTONDOWN,0,0); | Посылка окну «h» сообщение с параметрами (эмуляция нажатия левой кнопки мыши) |
| 11. | MoveWindow(h,210,38,30,20,true); | Задать окну «h» координаты x и y верхнего левого угла, ширину, высоту и флаг автоматической перерисовки окна |
| 12. | CloseWindow(h); | Сворачивание окна «h» |
| 13. | PostMessage(h,WM\_QUIT,0,0); | Посылка сообщения для закрытия окна «h» |
|  |  |  |
| 14. | Edit1.Perform(WM\_CHAR,byte(“!”),0);  Combobox1.Perform(WM\_KEYDOWN,VK\_DOWN,0); | Посылка компоненту формы сообщения (эмуляция ввода символа в компонент, эмуляция нажатия стрелки вниз). Используется в собственных программах на Delphi |
|  |  |  |
| 15. | ShellExecute(handle,’open’,’c:\’,nil,nil,sw\_shownormal) | Открывает нужную папку в Проводнике |
| 16. | ShellExecute(application.MainForm.handle, ’open’,  ’test.doc’, nil, ’c:\temp’, sw\_shownormal) | Открывает документ «test.doc». Папка для временных файлов «c:\temp» |
| 17. | ShellExecute(handle, ’open’, ’mailto: tsn\_prof@mail.ru’, nil, nil, sw\_shownormal) | Открывает почтовый клиент для создания письма для заданного электронного адреса |
| 18. | ShellExecute(handle, ’open’, ’http://www.pau.kz’, nil, nil, sw\_shownormal) | Открывает Интернет-браузер для просмотра указанной WWW-страницы |

**2. ПРИМЕР РАБОТЫ С API-ФУНКЦИЯМИ**

function GetText1(Wnd: HWND): string;

{ Получение текста из окна, способ 1 (универсальный) }

var textlength: Integer; Text: PChar;

begin

textlength := SendMessage(Wnd, WM\_GETTEXTLENGTH, 0, 0);

if textlength = 0 then Result := '' else begin

GetMem(Text, textlength + 1);

SendMessage(Wnd, WM\_GETTEXT, textlength + 1, Integer(Text));

Result := Text;

FreeMem(Text);

end;

end;

function GetText2(h: HWND): string;

{ Получение текста из окна, способ 2 }

var i:LongWord; s:string;

begin

i:=GetWindowTextLength(h)+1; SetLength(s,i);

GetWindowText(h,pchar(s),i); SetLength(s,i-1);

GetText2:=s;

end;

procedure SetTxt1 (h: HWND; s: string);

{ Установка текста в окне (универсальный) }

begin

SendMessage(h,WM\_SETTEXT,0, longint(pchar(s)));

end;

procedure SetTxt2 (h: HWND; s: string);

{ Установка текста в окне }

begin

SetWindowText(h,pansichar(s));

end;

Procedure ClickWnd(h: HWND);

{ "Нажатие" на окно левой кнопкой мыши }

begin

SendMessage(h,WM\_LBUTTONDOWN,0,0);

SendMessage(h,WM\_LBUTTONUP,0,0);

end;

procedure ClickTB(h: HWND; BTN\_ID: integer);

{ "Нажатие" на кнопку тулбара }

begin

SendMessage(h, WM\_COMMAND, BTN\_ID, 0);

end;

procedure Button3Click(Sender: TObject);

{ Клик на кнопку "Добавить" в архиваторе "7z" }

var h: hwnd;

begin

h:=Findwindow('FM',nil);

if h=0 then winexec('C:\Program Files\7-Zip\7zFM.exe',sw\_show);

h:=Findwindow('FM',nil); sleep(1000);

if h=0 then exit;

SetForeGroundWindow(h);

SetTxt2 (h,’ok’);

MoveWindow(h,210,38,30,20,true);

h:=GetWindow(h,gw\_child);

h:=GetWindow(h,gw\_child);

h:=GetWindow(h,GW\_HWNDNEXT);

ClickTB(h, 2000);

end;

**3. СОЗДАНИЕ ПРОГРАММНЫХ ЛОВУШЕК В DELPHI**

unit Loader\_DLL\_TSN;

…

var

Form1: TForm1;

procedure hook(switch : Boolean); stdcall; external 'hook\_dll\_tsn.dll';

const IDM\_TSN=$C0;

implementation

{$R \*.dfm}

procedure TForm1.FormActivate(Sender: TObject);

var h: HWND; p, pp: HMENU;

begin

h:=FindWindow('SciCalc',nil); if h=0 then winexec('calc',SW\_SHOW);

h:=FindWindow('SciCalc',nil); if h=0 then exit else SetForegroundWindow (h);

hook(true); // Устанавливаем наш перехватчик событий Windows

p:=GetMenu(h); // Верхнее меню

// Меняем название пункта в верхнем меню "Справка"

ModifyMenu(p, 2, MF\_BYPOSITION, GetMenuItemID(p,2), '-TSN-' );

pp:= GetSubMenu(p,2); // Подпункты в верхнем меню (с нуля отсчет)

// Меняем название подпункта в подменю "Справка -> О программе"

ModifyMenu(pp, 2, MF\_BYPOSITION, GetMenuItemID(pp,2) , 'About Sergius ...' );

// Добавляем новый элемент подменю с идентификатором IDM\_TSN

AppendMenu(pp, MF\_STRING ,IDM\_TSN, 'А это новый пункт !');

DrawMenuBar(h); // Прорисовываем модифицированное меню

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

{ при закрытии формы - освобождаем Hook }

var h: HWND;

begin

hook(false); // Деактивизируем наш перехватчик событий Windows

h:=FindWindow('SciCalc',nil);

h:=FindWindow('SciCalc',nil); if h<>0 then Postmessage(h,WM\_QUIT,0,0);

end;

end.

library HOOK\_DLL\_TSN;

uses Windows, Messages, SysUtils;

var SysHook: HHook = 0; Wnd: Hwnd = 0;

const IDM\_TSN=$C0;

function SysMsgProc(code : integer; wParam : word; lParam : longint) : longint; stdcall;forward;

{Процедура установки HOOK-а}

procedure hook(switch : Boolean) export; stdcall;

begin

if switch=true then

// Устанавливаем хук на сообщения после их обработки в приложении

// Наша процедура захвата называется "SysHook", в данном модуле

SysHook := SetWindowsHookEx(WH\_GETMESSAGE, @SysMsgProc, HInstance, 0)

else UnhookWindowsHookEx(SysHook);

end;

function SysMsgProc(code : integer; wParam : word; lParam : longint) : longint; stdcall;

{ Обработчик перехвата }

label 1;

var dc: HDC; h, h2: HWND; x: cardinal; ss: string;

begin

{ Если нам не стоит обрабатывать это событие, то и не надо,

передадим упраление следующиму хуку }

if (code<0) or (code<>HC\_ACTION) then goto 1;

// Ищем калькулятор, если он не запущен, но отдыхаем ...

h:=FindWindow('SciCalc',nil); if h=0 then goto 1;

h2:=GetWindow(h,GW\_CHILD); // Нашли окно отображения (табло) у кальулятора

// Если калькулятор не виден, то ничего не надо ...

if not IsWindowVisible(h) then goto 1;

// Получаем указатель на текущее обрабатываемое окно, ждем калькулятор

Wnd:=TMsg(Pointer(lParam)^).hwnd;

// Механизм нашей прорисовки на калькуляторе для любого окна

ss:='Кликни правой по калку ...';

dc:=GetDC(h); SetBkMode(dc, TRANSPARENT);

TextOut(dc,15,4,pansichar(ss),length(ss));

// ---------------------------------------------

x:=TMsg(Pointer(lParam)^).message; // Получаем код сообщения окна

// Запрещаем щелкать по окошку калькулятора

if (wnd=h2) and (x=WM\_LBUTTONDOWN) then begin

MessageBox(0, 'Нельзя !!!', 'Щелкать по окошку',0 );

TMsg(Pointer(lParam)^).message:=0; // Анулируем сообщение

end;

// Механиз обработки "нашего" пункта меню в калькуляторе

if (wnd=h) and (x=WM\_COMMAND)

and (lo(TMsg(Pointer(lParam)^).wParam) = IDM\_TSN)

then MessageBox(0, 'Да!', 'Калькулятор говорит', 0);

{ Передаем сообщение дальше по цепочке hook-ов }

1: Result:=CallNextHookEx(SysHook, Code, wParam, lParam);

end;

exports hook;

begin

end.