**2 ЛАБОРАТОРНАЯ РАБОТА №2**

**ХУКИ**

**Цель работы:** Освоить понятие хука. Изучить механизмы работы хуков. Получить практические навыки по установке хуков и разработке функций-обработчиков хуков.

**2.1 Текст программы полученной в пункте 2.3**

#include <windows.h>

#include <tchar.h>

#define EXPORT extern "C" \_\_declspec(dllexport)

EXPORT LRESULT CALLBACK on\_wm\_char(int nCode, WPARAM wParam, LPARAM

lParam);

#include "hook.h"

EXPORT LRESULT CALLBACK on\_wm\_char(int nCode, WPARAM wParam, LPARAM

lParam){

\_TCHAR data[10] = L"";

data[0] = (\_TCHAR)wParam;

data[1] = '\0';

MessageBox(NULL, data, L"WM\_CHAR", MB\_OK);

return CallNextHookEx(NULL, nCode, wParam, lParam);

}

#include <windows.h>

#include <stdio.h>

#include <tchar.h>

LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam){

switch (msg){

case WM\_CREATE:

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

};

return DefWindowProc(hWnd, msg, wParam, lParam);

}

int WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst, LPSTR lpCmdLine, int

nCmdShow){

WNDCLASS hWc;

hWc.cbClsExtra = 0;

hWc.cbWndExtra = 0;

hWc.hbrBackground = (HBRUSH)GetStockObject(WHITE\_BRUSH);

hWc.hCursor = LoadCursor(NULL, IDC\_ARROW);

hWc.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

hWc.hInstance = hInst;

hWc.lpfnWndProc = WndProc;

hWc.lpszClassName = L"MyClass";

hWc.lpszMenuName = NULL;

hWc.style = CS\_HREDRAW | CS\_VREDRAW;

RegisterClass(&hWc);

HWND hWnd = CreateWindow(L"MyClass", L"Test", WS\_OVERLAPPEDWINDOW,

50, 50, 400, 300, NULL, NULL, hInst, 0);

ShowWindow(hWnd, SW\_SHOW);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Load hook dll \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

HINSTANCE dll\_descr = LoadLibrary(L"hook.dll");

if (dll\_descr==NULL){

MessageBox(hWnd, L"Load error", L"Error", MB\_OK);

return 0;

}

HOOKPROC on\_wm\_char = (HOOKPROC)GetProcAddress(dll\_descr, "on\_wm\_char");

if (on\_wm\_char==NULL){

MessageBox(hWnd, L"GetProcAddress error", L"Error", MB\_OK);

return 0;

}

HHOOK h\_hook = SetWindowsHookEx(WH\_KEYBOARD, (HOOKPROC)on\_wm\_char,

dll\_descr, 0);

if (h\_hook==NULL){

MessageBox(hWnd, L"SetWindowsHookEx error", L"Error", MB\_OK);

return 0;

}

MSG msg;

while (GetMessage(&msg, NULL, NULL, NULL)){

DispatchMessage(&msg);

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Clear hook; unload dll \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

UnhookWindowsHookEx(h\_hook);

FreeLibrary(dll\_descr);

return 0;

};

**2.2 Текст программы заданной на самостоятельную работу**

Файл filter\_func.cpp

#include "filter\_functions.h"

EXPORT LRESULT CALLBACK on\_wm\_lbutton(int nCode, WPARAM wParam, LPARAM lParam){

if(wParam == WM\_LBUTTONDOWN && GetKeyState(VK\_MENU) < 0){

MOUSEHOOKSTRUCT \*mhs = (MOUSEHOOKSTRUCT\*)lParam;

HWND hWnd = mhs->hwnd;

SendMessage(hWnd, WM\_DESTROY, 0, 0);

}

return CallNextHookEx(NULL, nCode, wParam, lParam);

}

Файл filter\_func.h

#include <windows.h>

#include <tchar.h>

#define EXPORT extern "C" \_\_declspec(dllexport)

EXPORT LRESULT CALLBACK on\_wm\_lbutton(int nCode, WPARAM wParam, LPARAM lParam);

Файл main.cpp

#include <stdio.h>

#include <windows.h>

LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam);

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow){

TCHAR class\_name[] = L"main window";

TCHAR app\_title[] = L"simple\_window";

WNDCLASS wc;

HWND hWnd;

MSG msg;

memset(&wc, 0, sizeof(wc));

wc.lpszClassName = class\_name;

wc.lpfnWndProc = &WndProc;

wc.style = CS\_HREDRAW | CS\_VREDRAW;

wc.hInstance = hInstance;

wc.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wc.hbrBackground = (HBRUSH)GetStockObject(WHITE\_BRUSH);

wc.hCursor = LoadCursor(NULL, IDC\_ARROW);

wc.lpszMenuName = NULL;

RegisterClass(&wc);

hWnd = CreateWindow(class\_name,

app\_title,

WS\_OVERLAPPEDWINDOW,

180,

20,

500,

400,

NULL,

NULL,

hInstance,

NULL);

if(!hWnd){

MessageBox(NULL, L"cannot create window", app\_title, MB\_OK|MB\_ICONSTOP);

return FALSE;

}

ShowWindow(hWnd, SW\_SHOWNORMAL);

UpdateWindow(hWnd);

HINSTANCE dll\_descr = LoadLibrary(L"filter\_fun\_dll.dll");

if(dll\_descr == NULL){

MessageBox(hWnd, L"Load error", L"error", MB\_OK);

return 0;

}

HOOKPROC on\_wm\_lbutton = (HOOKPROC)GetProcAddress(dll\_descr, "on\_wm\_lbutton");

if(on\_wm\_lbutton == NULL){

MessageBox(hWnd, L"GetProcAddress error", L"error", MB\_OK);

return 0;

}

HHOOK h\_hook = SetWindowsHookEx(WH\_MOUSE, (HOOKPROC)on\_wm\_lbutton, dll\_descr, 0);

if(on\_wm\_lbutton == NULL){

MessageBox(hWnd, L"SetWindowsHookEx error", L"error", MB\_OK);

return 0;

}

while(GetMessage(&msg, hWnd, 0, 0)){

TranslateMessage(&msg);

DispatchMessage(&msg);

}

UnhookWindowsHookEx(h\_hook);

FreeLibrary(dll\_descr);

return msg.wParam;

}

LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam){

TCHAR str[] = L"Text";

switch(msg){

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

return DefWindowProc(hWnd, msg, wParam, lParam);

}

return 0;

}

**Выводы**

В ходе лабораторной работы был изучен механизм хуков в ОС Windows. Было создано приложение, которое устанавливает хук WH\_MOUSE, который перехватывает нажатие клавиш «Alt+LeftMouseClick» и посылает сообщение WM\_DESTROY окну, которому данное нажатий клавиш было адресовано.