Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №6

по дисциплине «Основы операционных систем»

на тему «Перехват и фильтрация информации с использованием системных перехватчиков»

Выполнили студенты группы 22ВВП1

Хоссейни Нежад С.А.С.М.

Захаров А.С.

Приняли:

Егоров В. Ю.

Федюнин Р. Н.

Пенза 2024

**Название**

Перехват и фильтрация информации с использованием системных перехватчиков

**Цель работы**

Изучить методику установки и снятия системных перехватчиков для эффективной реакции программы на действия пользователя.

**Лабораторное задание**

Разработать программу, выводящую на экран текстовое сообщение при запуске нового оконного приложения.

**Описание данных**

HHOOK hHook: Дескриптор глобального хука для клавиатуры.

typedef LRESULT (\*ShellProc)(int, WPARAM, LPARAM): Определение типа функции, которая будет использоваться в качестве процедуры обработки.

int\* pStopFlag: Указатель на переменную, определяющую, следует ли завершить работу программы.

int stopFlag: Глобальная переменная в библиотеке HokkWindAppLib.dll, указывающая на необходимость остановки хука (0 - продолжать, 1 - остановить).

MSG msg: Структура, используемая для хранения сообщений Windows.

**Текст программы**

**Code**

**main**

#include <windows.h>

#include <stdio.h>

HHOOK hHook = NULL;

typedef LRESULT(\*ShellProc)(int, WPARAM, LPARAM);

int main() {

HINSTANCE hLib = LoadLibrary(TEXT("HookWindAppLib.dll"));

if (!hLib) {

printf("Failed to load DLL.\n");

return 1;

}

ShellProc shellProc = (ShellProc)GetProcAddress(hLib, "ShellProc");

int\* pStopFlag = (int\*)GetProcAddress(hLib, "stopFlag");

if (!shellProc || !pStopFlag) {

printf("Failed to get function addresses.\n");

FreeLibrary(hLib);

return 1;

}

if (hHook == NULL) {

hHook = SetWindowsHookEx(WH\_SHELL, shellProc, hLib, 0);

if (hHook == NULL) {

wprintf(L"Failed to set hook\n");

FreeLibrary(hLib);

return 1;

}

}

MSG msg;

while (1) {

if (\*pStopFlag) {

PostQuitMessage(0);

break;

}

if (PeekMessage(&msg, NULL, 0, 0, PM\_REMOVE)) {

TranslateMessage(&msg);

DispatchMessage(&msg);

if (msg.message == WM\_QUIT) {

break;

}

}

}

if (hHook != NULL) {

UnhookWindowsHookEx(hHook);

hHook = NULL;

}

FreeLibrary(hLib);

return 0;

}

**HookWindAppLib**

**main**

#include <windows.h>

#include <stdio.h>

#include "hookWindAppLib.h"

int stopFlag = 0;

if (nCode == HSHELL\_WINDOWCREATED && stopFlag == 0) {

stopFlag = 1;

MessageBox(NULL, L"New window has been created!", L"Window Creation Alert", MB\_OK);

}

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

}

**hookWindAppLib.h**

#ifndef HOOK\_WIND\_APP\_LIB\_H

#define HOOK\_WIND\_APP\_LIB\_H

#ifdef HOOK\_WIND\_APP\_LIB\_EXPORTS

#define HOOK\_WIND\_APP\_LIB\_API \_\_declspec(dllexport)

#else

#define HOOK\_WIND\_APP\_LIB\_API \_\_declspec(dllimport)

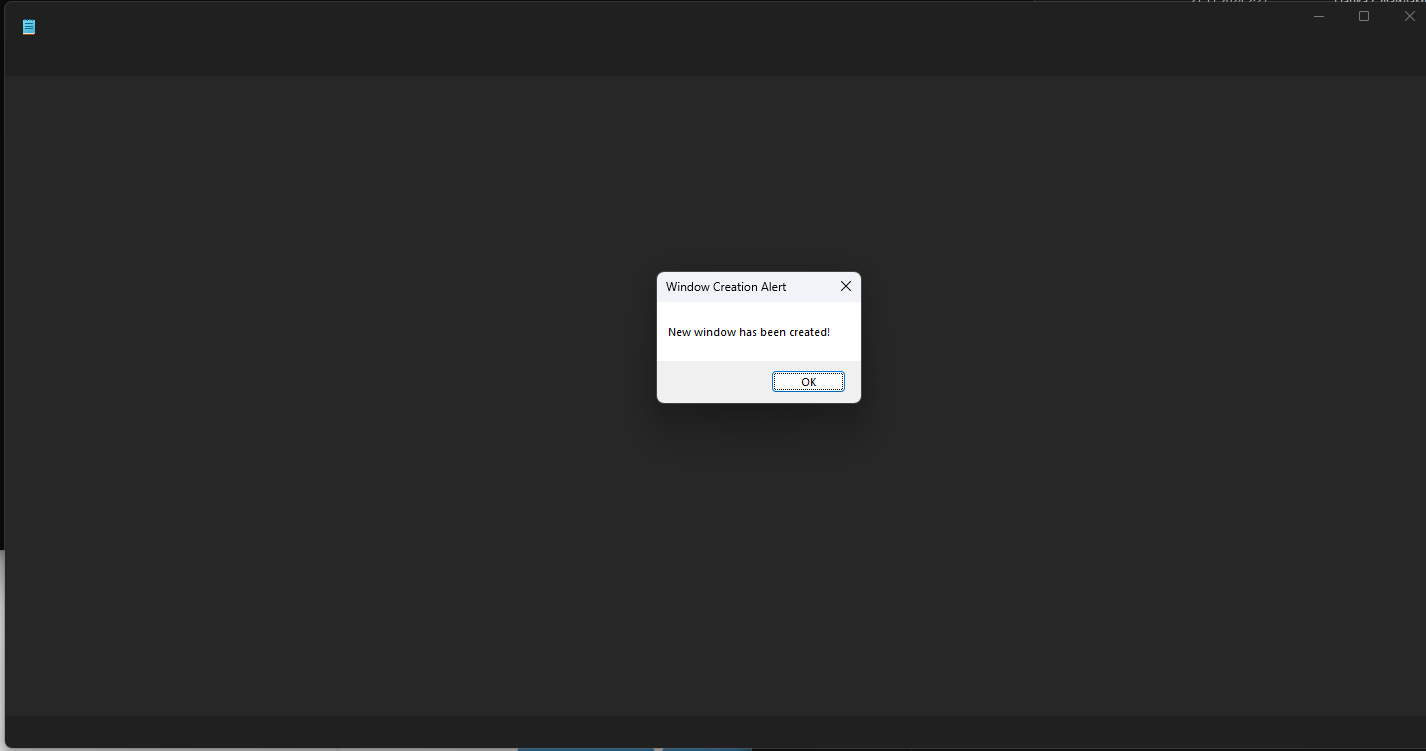
#endif

HOOK\_WIND\_APP\_LIB\_API extern int stopFlag;

HOOK\_WIND\_APP\_LIB\_API LRESULT CALLBACK ShellProc(int nCode, WPARAM wParam, LPARAM lParam);

#endif

**Результат работы программы**



**Вывод**

Изучили методику установки и снятия системных перехватчиков для эффективной реакции программы на действия пользователя.