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

Демин М. С.

Беляев Д. И.

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 нажатии пользователем комбинации клавиш Alt+S.

**Описание данных**

1. Переменные и структуры:

• HHOOK hHook: Дескриптор глобального хука для клавиатуры.

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

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

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

• MSG msg: Структура, используемая для хранения сообщений Windows.

2. Флаги и сообщения:

• WM\_KEYDOWN и WM\_SYSKEYDOWN: Сообщения, указывающие на нажатие клавиши.

• VK\_MENU: Константа, представляющая клавишу Alt.

**Описание структуры программы**

1. Файл main.c:

• Загружает DLL-библиотеку keyHook.dll.

• Получает адрес функции KeyboardProc и указателя на переменную stopFlag из загруженной библиотеки.

• Устанавливает глобальный хук для клавиатуры с помощью функции SetWindowsHookEx, передавая адрес функции обработки клавиатуры.

• В бесконечном цикле проверяет состояние флага остановки (stopFlag). Если он установлен в 1, программа завершает работу.

• Обрабатывает сообщения Windows, используя PeekMessage, TranslateMessage и DispatchMessage.

• При завершении работы снимает хук с помощью UnhookWindowsHookEx и освобождает загруженную библиотеку с помощью FreeLibrary.

2. Файл main.c [keyHook]:

• Определяет обработчик клавиатурного хука (KeyboardProc), который реагирует на события нажатия клавиш.

• Если нажата комбинация клавиш Alt+S, устанавливает флаг остановки (stopFlag = 1) и выводит сообщение в консоль.

• Возвращает результат вызова следующего обработчика с помощью функции CallNextHookEx.

3. Файл keyHook.h:

• Содержит определения для экспорта переменной stopFlag и функции KeyboardProc.

• Определяет макросы для управления экспортом функций в DLL.

**Текст программы**

**main.c [code]**

#include <windows.h>

#include <stdio.h>

HHOOK hHook = NULL;

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

int main() {

HMODULE hHookLib = LoadLibrary(TEXT("keyHook.dll"));

if (hHookLib == NULL) {

wprintf(L"Error when loading DLL\n");

return 0;

}

KeyboardProc keyboardProc = (KeyboardProc)GetProcAddress(hHookLib, "KeyboardProc");

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

if (!keyboardProc || !pStopFlag) {

wprintf(L"Error when finding function or stopFlag\n");

FreeLibrary(hHookLib);

return 1;

}

if (hHook == NULL) {

hHook = SetWindowsHookEx(WH\_KEYBOARD\_LL, keyboardProc, hHookLib, 0);

if (hHook == NULL) {

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

FreeLibrary(hHookLib);

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(hHookLib);

return 0;

}

**main.c [keyHook]**

#include <windows.h>

#include <stdio.h>

#include "keyHook.h"

int stopFlag = 0;

LRESULT CALLBACK KeyboardProc(int nCode, WPARAM wParam, LPARAM lParam) {

if (nCode == HC\_ACTION) {

if (wParam == WM\_KEYDOWN || wParam == WM\_SYSKEYDOWN) {

KBDLLHOOKSTRUCT\* pKeyboard = (KBDLLHOOKSTRUCT\*)lParam;

if ((GetAsyncKeyState(VK\_MENU) & 0x8000) && pKeyboard->vkCode == 'S') {

wprintf(L"Alt+S pressed");

stopFlag = 1;

}

}

}

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

}

**keyhook.h [keyHook]**

#ifndef KEY\_HOOK

#define KEY\_HOOK

#ifdef KEY\_HOOL\_EXPORTS

#define KEY\_HOOK\_API \_\_declspec(dllexport)

#else

#define KEY\_HOOK\_API \_\_declspec(dllimport)

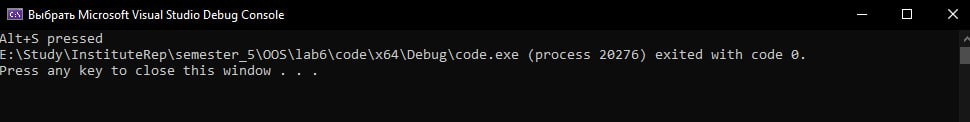
#endif

KEY\_HOOK\_API extern int stopFlag;

KEY\_HOOK\_API LRESULT CALLBACK KeyboardProc(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.