Университет ИТМО

Кафедра прикладной математики и информатики

Системное программное обеспечение

Лабораторная работа № 6

Выполнил:

Сорокин Юрий

Группа Р3217

Преподаватель:

Зыков А.Г.

СПб

2016 г.

# Лабораторная работа № 6

#include "windows.h"

#include "stdio.h"

#define VK\_Q 0x51

HHOOK hhkLowLevelKybd;

HANDLE console;

CONSOLE\_SCREEN\_BUFFER\_INFO info;

bool ENABLED = false;

bool L\_PRESSED = false;

bool R\_PRESSED = false;

bool CTRL\_PRESSED = false;

LRESULT CALLBACK LowLevelKeyboardProc(int nCode, WPARAM wParam, LPARAM lParam)

{

if (nCode < 0) return CallNextHookEx(hhkLowLevelKybd, nCode, wParam, lParam);

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

switch (wParam)

{

case WM\_KEYDOWN:

case WM\_SYSKEYDOWN: {

if (CTRL\_PRESSED && (l->vkCode == VK\_Q)) {

ENABLED = !ENABLED;

}

if (ENABLED) {

SetConsoleTextAttribute(console, FOREGROUND\_GREEN);

printf("+%ld\n", l->vkCode);

SetConsoleTextAttribute(console, info.wAttributes);

}

L\_PRESSED = l->vkCode == VK\_LCONTROL || L\_PRESSED;

R\_PRESSED = l->vkCode == VK\_RCONTROL || R\_PRESSED;

CTRL\_PRESSED = L\_PRESSED || R\_PRESSED || CTRL\_PRESSED;

break;

}

case WM\_KEYUP:

case WM\_SYSKEYUP: {

L\_PRESSED = l->vkCode != VK\_LCONTROL && L\_PRESSED;

R\_PRESSED = l->vkCode != VK\_RCONTROL && R\_PRESSED;

CTRL\_PRESSED = L\_PRESSED || R\_PRESSED;

if (ENABLED) {

SetConsoleTextAttribute(console, FOREGROUND\_RED);

printf("-%ld\n", l->vkCode);

SetConsoleTextAttribute(console, info.wAttributes);

}

break;

}

}

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

}

SERVICE\_STATUS serviceStatus;

SERVICE\_STATUS\_HANDLE hStatus;

int main(DWORD argc, LPTSTR argv[])

{

console = GetStdHandle(STD\_OUTPUT\_HANDLE);

GetConsoleScreenBufferInfo(console, &info);

// Install the low-level keyboard & mouse hooks

hhkLowLevelKybd = SetWindowsHookEx(WH\_KEYBOARD\_LL, LowLevelKeyboardProc, 0, 0);

// Keep this app running until we're told to stop

MSG msg;

while (!GetMessage(&msg, NULL, NULL, NULL)) { //this while loop keeps the hook

TranslateMessage(&msg);

DispatchMessage(&msg);

}

printf("Ping!\n");

UnhookWindowsHookEx(hhkLowLevelKybd);

SetConsoleTextAttribute(console, info.wAttributes);

CloseHandle(console);

return 0;

}**Вывод**

При выполнении данной работы я ознакомился с перехватыванием событий в Windows.