**Системное программное обеспечение**

**Лабораторная работа №6**

Съестов Дмитрий Вячеславович

P3217

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

using namespace std;

#define MAX\_KEYSTROKES 255

#define CONTINUE\_INPUT 0

#define STOP\_INPUT   1

#define ACTION\_HOTKEY\_ID 1

#define EXIT\_HOTKEY\_ID   2

#define VK\_K 0x4B

#define VK\_Q 0x51

INPUT keystrokes[MAX\_KEYSTROKES];

unsigned char keyCount = 0;

int getKeystroke()

{

    HANDLE hStdin = GetStdHandle(STD\_INPUT\_HANDLE);

    INPUT\_RECORD inputBuffer[MAX\_KEYSTROKES];

    DWORD inputCount = 0;

    ReadConsoleInput(hStdin, inputBuffer, MAX\_KEYSTROKES, &inputCount);

    for (int i = 0; i < inputCount; i++)

    {

        if (inputBuffer[i].EventType != KEY\_EVENT) continue;

        auto keyEvent = inputBuffer[i].Event.KeyEvent;

        if (keyEvent.wVirtualKeyCode == VK\_RETURN) return STOP\_INPUT;

        INPUT input;

        auto vk = keyEvent.wVirtualScanCode;

        input.type = INPUT\_KEYBOARD;

        input.ki.wVk = keyEvent.wVirtualKeyCode;

        input.ki.wScan = keyEvent.wVirtualScanCode;

        input.ki.time = 0;

        input.ki.dwExtraInfo = 0;

        input.ki.dwFlags = KEYEVENTF\_UNICODE;

        if (keyEvent.bKeyDown == FALSE) input.ki.dwFlags |= KEYEVENTF\_KEYUP;

        keystrokes[keyCount] = input;

        keyCount++;

        if (keyCount == MAX\_KEYSTROKES) return STOP\_INPUT;

    }

    return 0;

}

int main()

{

HANDLE hMutex = CreateMutex( NULL, TRUE, L"spo\_lab6\_mutex" );

    if(GetLastError() == ERROR\_ALREADY\_EXISTS)

    {

        puts("Program is already running!");

        puts("Press any key to exit.");

        getchar();

        return EXIT\_FAILURE;

    }

    puts("Enter your keystroke sequence.\n");

    while (keyCount < MAX\_KEYSTROKES)

    {

        if(getKeystroke() == STOP\_INPUT) break;

    }

    bool actionHotkey = RegisterHotKey(NULL, ACTION\_HOTKEY\_ID, MOD\_ALT, VK\_K);

    bool exitHotkey = RegisterHotKey(NULL, EXIT\_HOTKEY\_ID, MOD\_ALT, VK\_Q);

    if (!actionHotkey || !exitHotkey)

    {

        puts("Failed to initialize hotkeys!");

        puts("Press any key to exit.");

        getchar();

        ReleaseMutex(hMutex);

        CloseHandle(hMutex);

        return EXIT\_FAILURE;

    }

    puts("Press <Alt-K> to simulate keystrokes or <Alt-Q> to exit.");

    MSG msg = {0};

while (GetMessage(&msg, NULL, 0, 0) != 0)

{

if (msg.message == WM\_HOTKEY)

{

WORD vk = msg.lParam >> 16;

            if (vk == VK\_K)

            {

                Sleep(300); //Подождём, пока пользователь не отпустит Alt

                UINT simulatedEvents = SendInput(keyCount, keystrokes, sizeof(INPUT));

                if (simulatedEvents == keyCount) puts("Simulated keystroke sequence.");

                else puts("Failed to simulate keystroke sequence!");

            }

            else if (vk == VK\_Q)

            {

                UnregisterHotKey(NULL, ACTION\_HOTKEY\_ID);

                UnregisterHotKey(NULL, EXIT\_HOTKEY\_ID);

                ReleaseMutex(hMutex);

                CloseHandle(hMutex);

                return EXIT\_SUCCESS;

            }

}

}

}