Universidad Rafael Landívar

Facultad de Ingeniería

Curso de Microprogramación

Sección: 01

MANUAL DE USUARIO

Proyecto Práctico 02: Keylogger

Integrantes:
Carlos Lam 1193916
Max Díaz 1146916
Alexander Solorzano1243717
Alexander Rodríguez 1053016

Ciudad de Guatemala, 23 de Noviembre del 2019.

Objetivo

El objetivo del proyecto es aplicar los conocimientos aprendidos en clase para realizar KeyLogger. El usuario presiona teclas y estas son captadas por el programa, para posteriormente mostrarlo en consola y almacenar los caracteres en un archivo de texto en el escritorio.

Desarrollo de la aplicación

1. Directivas

Segmentos simplificados

.MODEL *modelo*: Debe estar ubicada antes de otra directiva de segmento.
El *modelo* utilizado es el siguiente:

- **SMALL**: Los datos caben en un segmento de 64 KB y el código cabe en otro segmento de 64 KB.

.STACK *Define el segmento de pila de la longitud especificada*.
.CODE Define el segmento de código.
.DATA: Define un segmento de datos NEAR con valores iniciales

El siguiente símbolo está definido cuando se usan las directivas anteriores:

- @**data**: Nombre del segmento definido con la directivas **.DATA**, **.DATA?**, **.CONST** y **.STACK** (los cuatro están en el mismo segmento).

Definición de datos

Ubica memoria para un ítem de datos y opcionalmente asocia un nombre simbólico con esa dirección de memoria y/o genera el valor inicial para ese ítem.

DB define datos que pueden ser cadenas o una expresión numérica por medio de bits.

DUP define una cantidad de datos y la repetición del valor que debe contener (el valor que queremos que contenga va entre paréntesis)

Definición de segmento

Organizan el programa para utilizar los segmentos de memoria.
Control del ensamblador
END: Debe ser la última sentencia del código fuente.

2. Registros

Los registros de datos, también llamados “de propósito general”, como su nombre lo indica tienen generalmente datos. Aunque tienen distinto nombres, cuentan con básicamente con la misma funcionalidad.

AX es a menudo llamado acumulador.
BX se puede usar como registro base en algunos modos de direccionamiento, es decir, para apuntar a posiciones de memoria con él.
CX es usado por algunas instrucciones como contador (en ciclos, rotaciones)
DX o registro de datos; a veces se usa junto con AX.

Cada registro de estos está dividido a su vez en dos registros de 8 bits, que pueden ser leídos o escrito de manera independiente:

AX = **AH** | **AL** BX = **BH** | **BL**
CX = **CH** | **CL** DX = **DH** | **DL**

En donde AH es la parte alta del registro AX y AL es la parte baja del registro AX. (H viene de High y L de Low)

DS: registro de segmento adicionales, el primero llamado de datos (Data) y el segundo Extra. Con ellos apuntaremos a los segmentos donde tengamos nuestros datos (ya que del código se encarga CS), esto es, nuestras variables.

SI: registro índice, Empleado para direccionar datos fuente en forma indirecta y utilizarlos con las instrucciones de cadenas o arreglos.

Instrucciones de transferencia de datos

MOV destino, fuente. Copia el contenido del operando fuente en el destino.

Instrucciones aritméticas

INC destino. Incrementa en uno el destino.
DEC destino. Decrementa en uno el destino.

Instrucciones lógicas

XOR destino, fuente. Realiza la operación lógica XOR, almacenando el resultado en destino.

Instrucciones comparativas

CMP destino, fuente**.** Realiza una operación (destino – fuente)

Instrucciones de transferencia de control

CALL etiqueta. Llama a etiqueta

RET Retorno de procedimiento.

Instrucciones de saltos

JMP etiqueta. Salto incondicional hacia etiqueta.
JZ dir/ **JE** dir Saltar si el resultado es cero (Z = 1).
JNE dir. Saltar a *dir* si no es igual (Z=0).

Interrupciones

INT n. Ejecuta el manejador de la interrupción especificada en el operando.
LOOP etiqueta. Salta a etiqueta si CX <> 0, decrementando CX.

Operadores analíticos

OFFSET Devuelve el desplazamiento de una ubicación de memoria.

Funciones usadas

GetKeyState

Devuelve el estado de la tecla virtual especificada. El estado especifica si la tecla está arriba, abajo o alternada (activada, desactivada, alternando cada vez que se presiona la tecla).

Tecla virtual: Si la clave virtual deseada es una letra o un dígito (de la A a la Z, de la a a la z, o del 0 al 9), nVirtKey debe establecerse en el valor ASCII de ese carácter. Para otras claves, debe ser un código de clave virtual.

CreateFileA

Crea o abre un archivo. Los dispositivos de E/S más utilizados son los siguientes: archivo, flujo de archivos, directorio, disco físico, volumen, búfer de consola, unidad de cinta, recurso de comunicaciones, buzón de correo y canalización. La función devuelve un identificador que se puede utilizar para acceder al archivo o dispositivo para varios tipos de E/S en función del archivo o dispositivo y de las marcas y atributos especificados.