Manual Tecnico*

* Proyecto: Arquitectura y Ensambladores Sección A
Douglas Daniel Aguilar Cuque 201503935

*

Resumen– En este documento se describe el ensamblador utilizado en laproyecto, tambien se describe las partes relevantes del codigo, el modo de video utilizado y las interrupciones utilizadas en la practica.

Palabra clave-INT 21H, INT 10H, Modo video, registro.

***Abstract*—This document describes the assembler used in proyect, also describes the relevant parts of the code, the video mode used and the interruptions used in practice.**

***Index Terms*—INT 21H, INT 10H, Video mode, registration**

I. Marco Teórico

A. NASM: el ensamblador de Netwide

El ensamblador Netwide, NASM, es un ensamblador 80x86 y x86-64 diseñado para portabilidad y modularidad. Es compatible con una amplia gama de formatos de archivo de objetos, incluyendo Linux y *BSD a.out, ELF, COFF, Mach-O, 16 bits y 32 bits OBJ de formato (OMF), Win32y Win64. También generará archivos binarios simples, formatos Intel He y S-Record de Motorola. Su sintaxis está diseñada para ser simple y fácil de entender, similar a la sintaxis en el Manual del desarrollador de software Intel con una complejidad mínima. Es compatible con todas las extensiones de arquitectura x86 actualmente conocidas y tiene un gran soporte para macros.

NASM también viene con un conjunto de utilidades para manejar el RDOFFformato de archivo de objeto personalizado.

B. Explicación de partes relevantes del código

1) Macro Lectura de archivo: Primero se limpian los registros ax, cx y dx con la operacion xor entre ellos mismos. Despues se intenta abrir el archivo, si el archivo no existe muestra un mensaje de error, luego lee un total de 255 bites. De ultimo mueve la cantidad de caracteres leidos a una variable. De ultimo cierra el archivo.

2) Macro escritura de archivo: Primero crea el archivo, si no lo crea regresa al menu. Despues de crear el archivo, se abre el archivo en modo escritura. Luego se escribe el numero de caracteres que se ingresaron. Al final se cierra el archivo.

3) Poner pixel: Este macro se encarga de poner un pixel en la coordenada ingresada y con el color que se ponga.

C. pintar ejes

Esta parte se encarga de pintar los ejes X y Y

```
%macro leerarchivo 3
    ; 1 ruta archivo, 2 contenedor de los datos archiv
    ; 3 se pondra el numero de caracteres leidos
    ;limpiar ax,cx,dx
    xor ax, ax
    xor cx, cx
    xor dx, dx
    xor bx, bx
    ;abrir archivo
    mov ah, 3dh
    mov al, 0h
    mov cx, 00
    mov dx, %1
    int 21h
    ;archivo no existe
    jc noarchivo
    ;archivo si existe se lee
    mov bx, ax
    mov ah, 3fh
    mov cx, 499
    mov dx, [%2]
    int 21h

    ;numero de caracteres
    mov [%3], ax
    ;cierre del archivo
    mov ah, 3eh
    int 21h
%endmacro
```

Fig. 1. Codigo Macro lectura de archivo

```
%macro escribir_archivo 3
    ; 1 nombre archivo, 2 numero de caracters,
    ; 3 texto a escribir
    ; crear archivo
    mov ah, 3ch
    mov cx, 0
    mov dx, %1
    int 21h
    jc menu
    mov bx, ax
    mov ah, 3eh
    int 21h

    ;abrir el archivo
    mov ah,3dh
    mov al,1h ;Abrimos el archivo en solo escritura.
    mov dx, %1
    int 21h
    jc menu ;Si hubo error

    ;Escritura de archivo
    mov bx,ax ; mover hadfile
    mov cx,[%2] ;num de caracteres a grabar
    mov dx,%3
    mov ah,40h
    int 21h

    cmp cx,ax
    jne menu ;error salir
    mov ah,3eh  ;Cierre de archivo
    int 21h
%endmacro
```

Fig. 2. Codigo Macro Escritura de archivo

```
%macro pintar_pixel 3
    ; 1. Color, 2. Posicion X, 3. Posicion Y
    xor ax, ax
    xor bx, bx
    xor cx, cx
    mov ah, 0Ch
    mov al, %1
    mov bh, 0
    mov cx, %2
    mov dx, %3
    int 10h
%endmacro
```

Fig. 3. Macro poner pixel

```
pintarejes:
    xor bx, bx
    mov bx, 0
    mov [numcar], bx
    ciclox:
        pintar_pixel 6, 160, [numcar]
        mov bx, [numcar]
        inc bx
        mov [numcar], bx
        cmp bx, 200
        jne ciclox
    mov bx, 0
    mov [numcar], bx
    cicloy:
        pintar_pixel 6, [numcar], 100
        mov bx, [numcar]
        inc bx
        mov [numcar], bx
        cmp bx, 320
        jne cicloy
    ret
```

Fig. 4. Pintar ejes

D. Evaluar funcion

Se obtine la funcion en que se esta evaluando y su valor pasa a si, se reinician los valores luego se inicia a realizar los calculos de cada valor

```
evaluarpintarfuncion:
    xor eax, eax
    xor ebx, ebx
    mov eax, [numerorecorrer]
    mov ebx, 18
    mul ebx
    mov [numerofuncionespor], eax

    mov si, [numerofuncionespor]


    xor eax, eax
    mov eax, 0
    mov [numx], eax
    mov [sigx], al
    mov [resultadogrado0], eax
    mov [resultadogrado1], eax
    mov [resultadogrado2], eax
    mov [resultadogrado3], eax
    mov [resultadogrado4], eax
    mov [resultadogrado5], eax
    mov [resultadofinal], eax
```

Fig. 5. Evaluar funcion

```
call multiplicacion
xor eax, eax
mov eax, [resultado]
mov [resultadogrado1], eax
xor al, al
mov al, [sigres]

mov [sigresgrado1], al
xor al, al
mov al, [arreglofunciones + si + 6]
mov [signum1], al
xor al, al
mov al, [arreglofunciones + si + 7]
mov [num1], al
```

Fig. 6. ¿datos

E. Modo video

1) INT 10H Función 06H: Esta funcion se utilizo para limpiar la pantalla.

Desplazar líneas de texto hacia arriba

- AH = 06H
- AL = Número de líneas a desplazar. Si AL=0, se borra toda la ventana seleccionada mediante los registros CX y DX
- BH = Atributo a usar en las líneas borradas.
- CH = Línea donde comienza la ventana de texto.
- CL = Columna donde comienza la ventana de texto.
- DH = Línea donde acaba la ventana de texto.
- DL = Columna donde acaba la ventana de texto.

.

F. INT 10H Función 03H

Establecer modo de Vídeo

- AH = 00H
- AL = Modo de vídeo.

.

G. INT 10H Función 13H

Establecer modo de Vídeo

- AH = 00H
- AL = Modo de vídeo.

.

1) INT 10H Función 02H: Esta funcion se utilizo para posicionar el cursor despues de limpiar la pantalla

- AH = 02H
- BH = Página de vídeo
- DH = Línea donde situar el cursor
- DL = Columna donde situar el cursor

.

H. Interrupciones utilizadas

1) INT 21H Función 01H: Leer un caracter con eco

- AH = 01H
- AL = Código ASCII del Carácter leído y Echo a pantalla

.

2) INT 21H Función 02H: imprimir un caracter

- AH = 02H
- DL = Código ASCII a enviar al dispositivo de salida.

.

3) INT 21H Función 08H: Leer un caracter sin eco

- AH = 08H
- AL = Código ASCII del Carácter leído

.

4) INT 21H Función 3CH: Crea un fichero

- AH = 3CH
- CX = Atributos del fichero: 00H Fichero Normal. 01H Fichero de Sólo Lectura. 02H Fichero Oculto. 03H Fichero de Sistema. DS:DX = Segmento: Desplazamiento de una cadena ASCII con el nombre de fichero.
- Si se ejecutó correctamente: Flag de acarreo (Cf) = 0 AX = Handle o manejador de fichero. Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1
- AX = Código de error.

.

5) INT 21H Función 3DH: Abre un fichero

- AH = 3DH
- AL = Modo de acceso. Bits 0-2: Permiso de lectura/escritura. DS:DX = Segmento: Desplazamiento de una cadena ASCII con el nombre de fichero.
- Si se ejecutó correctamente Flag de acarreo (Cf) = 0 AX = Handle o manejador de fichero
- Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1 AX = Código de error.

.

6) INT 21H Función 40H: Escribe en un fichero

- AH = 40H
- BX = Handle.
- CX = Número de bytes a escribir.
- DS:DX = Segmento: Desplazamiento del buffer desde donde se van a tomar los caracteres a escribir.
- Si se ejecutó correctamente: Flag de acarreo (Cf) = 0 AX = Bytes transferidos.
- Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1 AX = Código de error.

.

7) INT 21H Función 3EH: Cierra un fichero

- H = 3EH
- BX = Handle.
- Si se ejecutó correctamente: Flag de acarreo (Cf) = 0
- Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1 AX = Código de error.

.

8) INT 16H Función 00H: Retorna:

- AH = código de escaneo del teclado
- AL = carácter ASCII o cero si la tecla de función especial

.

9) INT 16H Función 01H: Retorna:

- ZF = 0 si se pulsa una tecla (incluso Ctrl-Break)
- AX = 0 si no hay un código de exploración disponible
- AH = código de escaneo
- AL = carácter ASCII o cero si la tecla de función especial

.

REFERENCES

[1] INT 10H, Modo video, INT 10H Recuperado de: http://ict.udlap.mx/people/oleg/docencia/Assembler/asm$_interrup_10.html$

[2] INT 21H, interrupción, INT 21H Recuperado de: http://ict.udlap.mx/people/oleg/docencia/ASSEMBLER/asm$_interrup_21.html$

[3] Manejo de archivos, Ensamblador básico, archivo Recuperado de: http://miensamblador.blogspot.com/2013/07/manejo-de-archivos.html

[4] NASM, NASM - The Netwide Assembler, NASM Recuperado de: https://www.nasm.us/doc/nasmdoc0.html