Manual Técnico Practica 4 (17 Octubre 2018)

Lima Ramirez, Juan Jose 201503738

Resumen— La presente documentación es un manual técnico donde se explica el código utilizado para la practica Cuatro

I. Introducción

II. Ensamblador Utilizado MASM

El Microsoft Macro Assembler (MASM) es un ensamblador para la familia x86 de microprocesadores. Fue producido originalmente por Microsoft para el trabajo de desarrollo en su sistema operativo MS-DOS, y fue durante cierto tiempo el ensamblador más popular disponible para ese sistema operativo. El MASM soportó una amplia variedad de facilidades para macros y programación estructurada, incluyendo construcciones de alto nivel para bucles, llamadas a procedimientos y alternación (por lo tanto, MASM es un ejemplo de un ensamblador de alto nivel). Versiones posteriores agregaron la capacidad de producir programas para los sistemas operativos Windows. MASM es una de las pocas herramientas de desarrollo de Microsoft para las cuales no había versiones separadas de 16 bits y 32 bits.

CODIGO RELEVANTE:

Imprimir: Durante la realización de la práctica se utilizaron varios macros, el más utilizado sería el macro llamado Imprimir a el cual se le envía como parámetro una cadena a imprimir, se manda al registro AX el @data que representa que se va a escribir una cadena, con el mnemónico mov ah,09 se le indica a la interrupción 21H que se iniciara una impresión de cadena en pantalla y al registro DX se le envía la dirección donde se almacena la cadena a imprimir

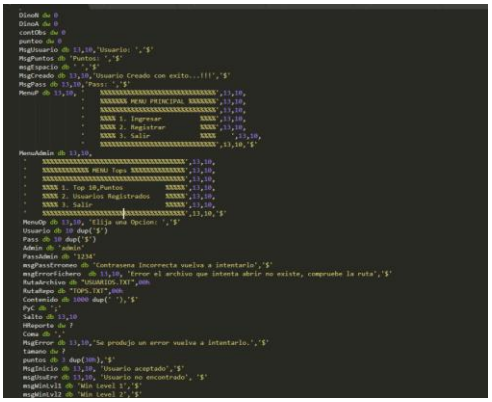
```
Imprimir macro cadena
        push ax
        push dx
        mov ax,@data
        mov ds,ax
        mov ah,09
        mov dx,offset cadena
        int 21h
        pop dx
        pop ax
endm
```

Macros para el manejo de archivos

```
AbrirArchivo macro Path,H
    lea dx,Path         ;Obtiene la direccion en memoria de ruta
    mov ah,3dh
    mov al, 00h
    int 21h
    mov H,ax
    jc Error
endm

LeerArchivo macro Tam,DataB,H
    mov ah,3fh
    mov bx,H
    mov cx,Tam
    lea dx,DataB
    int 21h
    jc Error
endm

CrearFichero macro ruta,handle
  lea dx,ruta
  mov ah,3ch
  mov cx,00h
  int 21h
  mov handle,ax
  jc Error
endm

EscribirFichero macro numbytes,databuffer,handle
  mov ah,40h
  mov bx,handle
  mov cx,numbytes
  lea dx,databuffer
  int 21h
  jc Error
endm

cerrar macro handle
  mov ah,3eh
  mov bx,handle
  int 21h
  jc Error
endm
```

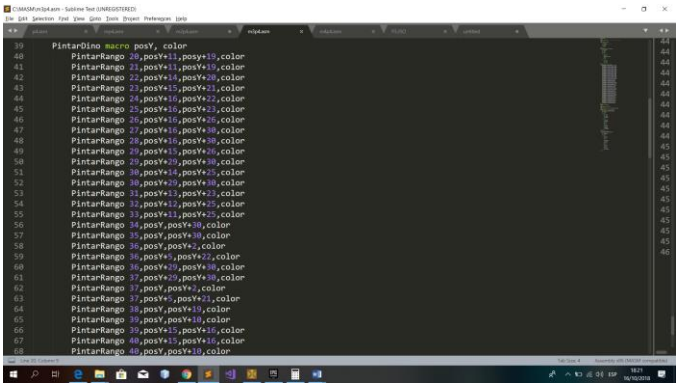
En estos macros se maneja una variable , que es un número que representa de manera única a un archivo para poder ser manejado de una manera más accesible. Para entrar más a detalle acerca de los valores que reciben las interrupciones para poder realizar el manejo de archivos consultar la siguiente página donde se documenta que valores devuelve y pide cada función dada en el registro alto de AX.

Se declararon todos los mensajes que podían ser mostrados en consola con variables que los representaban de tipo byte:



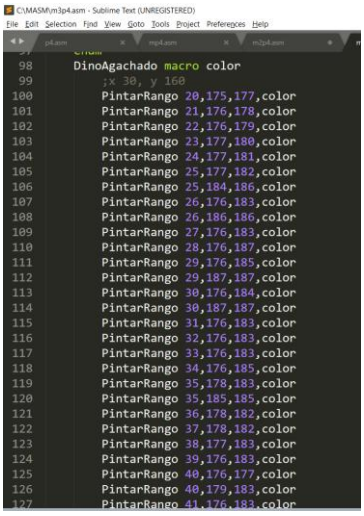
Macros Pintar dino:

En este macro se hace uso del Macro pintar rango para poder pintar el dinosaurio en pantalla, además se hace dinámico en el eje y, ya que se le suma a los puntos ya establecidos la nueva ubicación que se desea, también se puede cambiar el color del dinosaurio



Macros para pintar dinosaurio agachado

En este macro se pinta la posición del dinosaurio agachado, haciendo uso de macros que pintan tanto líneas verticales como horizontales, este macro recibe como parámetro el color que deseamos colocarle a cada pixel que será pintado



```
98      DinoAgachado macro color
99          ;x 30, y 160
100         PintarRango 20,175,177,color
101         PintarRango 21,176,178,color
102         PintarRango 22,176,179,color
103         PintarRango 23,177,180,color
104         PintarRango 24,177,181,color
105         PintarRango 25,177,182,color
106         PintarRango 25,184,186,color
107         PintarRango 26,176,183,color
108         PintarRango 26,186,186,color
109         PintarRango 27,176,183,color
110         PintarRango 28,176,187,color
111         PintarRango 29,176,185,color
112         PintarRango 29,187,187,color
113         PintarRango 30,176,184,color
114         PintarRango 30,187,187,color
115         PintarRango 31,176,183,color
116         PintarRango 32,176,183,color
117         PintarRango 33,176,183,color
118         PintarRango 34,176,185,color
119         PintarRango 35,178,183,color
120         PintarRango 35,185,185,color
121         PintarRango 36,178,182,color
122         PintarRango 37,178,182,color
123         PintarRango 38,177,183,color
124         PintarRango 39,176,183,color
125         PintarRango 40,176,177,color
126         PintarRango 40,179,183,color
127         PintarRango 41,176,183,color
```

Macros Obstaculo:

En este macros se pinta el obstáculo terrestre, a este macros se le envía la posición en x, así como el color que deseamos pintar los pixeles de dicho obstáculo, Este macros deja puntos fijos los cuales forman la figura del obstáculo y se hace uso de la posición x para crear así un desfase

```
2       Obstaculo macro posX,color
3           ;y max 187
4           xor ax,ax
5           xor bx,bx
6           xor cx,cx
7           xor dx,dx
8           mov ax,posX
9           PintarRango posX,170,185,color
10          mov ax,posX
11          add ax,1
12          PUNTO ax,169,color
13          mov ax,posX
14          add ax,1
15          PUNTO ax,172,color
16          mov ax,2
17          add ax,posX
18          mov bx,ax
19          add bx,7
20          PintarHorizontal ax,bx,168,color
21          mov ax,2
22          add ax,posX
23          mov bx,ax
24          add bx,7
25          PintarHorizontal ax,bx,173,color
26          mov ax,10
27          add ax,posX
28          PUNTO ax,169,color
29          mov ax,10
30          add ax,posX
31          PUNTO ax,172,color
```

MACROS Nube:

En este macros se pinta el obstáculo Aéreo, a este macros se le envía la posición en x, así como el color que deseamos pintar los pixeles de dicho obstáculo, Este macros deja puntos fijos los cuales forman la figura del obstáculo y se hace uso de la posición x para crear así un desfase

```
Nube macro posx,color
    ;160 pos max y
    mov ax,posx
    add ax,3
    mov bx,ax
    add bx,17
    PintarHorizontal ax,bx,160,color
    mov ax,posx
    add ax,2
    PUNTO ax,159,color
    mov ax,posx
    add ax,21
    PUNTO ax,159,color
    mov ax,posx
    add ax,1
    PUNTO ax,158,color
    mov ax,posx
    add ax,22
    PUNTO ax,158,color
    mov ax,posx
    PintarRango ax,155,157,color
    mov ax,posx
    add ax,23
    PintarRango ax,155,157,color
    mov ax,posx
    add ax,1
    PUNTO ax,154,color
    mov ax,posx
    add ax,22
    PUNTO ax,154,color
```

Macros Punto y Pintar rango:

El macro punto es utilizado para pintar un punto en la pantalla del color que se le envié, para eso necesita una posición en x, en y, por ultimo el color que queremos pintar dicho pixel

El macro pintar Rango es utilizado para crear una línea Vertical de pixeles, en este macro se hace un for que va pintando pixel por pixel en x,y

```
PUNTO MACRO x,y,color
    MOV CX,x
    MOV AH,0CH
    MOV AL,color    ;Color
    MOV BH,0
    MOV DX,y
    INT 10H
ENDM
PintarRango macro x,y,yf,color
    local R1,fnPR
    xor cx,cx
    xor dx,dx
    mov cx,x
    mov dx,y
    R1:
        cmp dx,yf
        jg fnPR
        PUNTO cx,dx,color
        add dx,1
        jmp R1
    fnPR:
        xor ax,ax
        xor bx,bx
        xor cx,cx
        xor dx,dx
endm
```

INTERRUPCIONES UTILIZADAS:

INT21H:

La mayoría de los servicios o funciones del sistema operativo MS-DOS se obtienen a través de la interrupción software 21H. Es por esto por lo que se le denomina DOS-API: DOSAPPLICATION-PROGRAM-INTERFACE La INT 21H está compuesta por un grupo de funciones. Cuando se accede a la INT 21H, hay que indicar el número de función que queremos ejecutar. La llamada a la INT 21H se realizará como sigue:

- Introducimos en (AH) el número de función a la que deseamos acceder.
- En caso de que deseemos acceder a una sub-función dentro de una función, debemos indicarlo introduciendo en (AL) el número de esa sub-función.
- Llamar a la INT 21H.

MODO DE VIDEO UTILIZADO:

INT10H:

Selecciona y activa el modo de vídeo especificado. A no ser que se utilice el truco que se indica a continuación, al llamar a esta función, se borra la pantalla. Pero se borra el contenido de pantalla. Por suerte hay una especie de 'truco' para evitar este borrado automático de la pantalla. Consiste en poner con valor 1 el bit 7 del registro AL (que contiene el modo de vídeo) en la llamada a la función.
Así por ejemplo, si queremos cambiar a modo 13h, y queremos que se pierda el contenido que hubiera en la pantalla en este modo, en vez de introducir en AL el número 13h (00010011b), introduciríamos el número 93h (10010011b).
La llamada a la INT 10H se realizará como sigue:

- Introducimos en (AH) 00h.
- En al introducimos el modo de video a utilizar
- Llamar a la INT 10H.
- Para esta práctica se utilizó el modo 13h

Referencias

[1] http://moisesrbb.tripod.com/unidad4.htm#unidad422https://www.youtube.com/watch?v=FtdzzVxbxWA
[2] http://ict.udlap.mx/people/oleg/docencia/ASSEMBLER/asm_interrup_21.html
[3] http://ict.udlap.mx/people/oleg/docencia/Assembler/asm_interrup_10.htmlhttps://programarfacil.com/tutoriales/fragmentos/servomotor-con-arduino/
[4] http://ict.udlap.mx/people/oleg/docencia/ASSEMBLER/asm_interrup_21.htmlhttps://www.youtube.com/watch?v=vBcyhk_cUDE
[5] Código Subido a classroom como ejemplo por nuestro auxiliar