

ARQUITECTURA DE COMPUTADORAS

UNIDAD 3

*PROGRAMAS DEL LENGUAJE ENSAMBLADOR CON INSTRUCCIONES DEFINIDAS*

*SEMESTRE 6*

*Ingeniería en Sistemas Computacionales*

*Maestro: Ing. Carlos Alberto Díaz Salgado*

*Alumno: Diego Alejandro Ávila Luna*

**COMPARACION DE 9 NUMEROS**

.model small ; declaramos modelo de memoria

.stack ; declaramos segmento de pila

.data ; define los datos

res db 0, 0dh, 0ah, "$" ;se declara la variable res

.code ;declaramos el segmento del codigo

inicio: ;inicia la programacion

mov ax, @data ;iniciar el programa

mov ds, ax

mov ax,0

mov bl,48 ;mueve el 48 a "bl"

ciclo:

cmp bl,58 ;se tiene que cumplir la condicion

je fin ;cuando la condicion se cumpla saltame a fin

mov res,bl ;mueveme el bl al resultado

lea dx,res ;mete res a dx y leelo

mov ah,09h ;imprimeme el resultado en pantalla

int 21h

inc bl ;esto es incremento de 1 que significa que va incrementar 1 el registro "bl"

jmp ciclo ;saltame a cilo e imprimeme el 1 y asi sucesivamente hasta hasta llegar al 9

fin: ;termina programa

mov ah,04ch

int 21h

end inicio ;finaliza programa

**COMPARACION DE UN NUMERO MAYOR Y MENOR**

.model small ; define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los datos

; se declaran las variables acontinuacion

dato1 db "el numero es mayor $"; declaramos mensaje 1

dato2 db "el numero es menor $"; declaramos mensaje 2

.code ;inicializacion

inicio:

mov ax,@data

mov ds,ax

mov ax,0

mov al,7 ;mover num 3 al registro "al".

cmp al,5 ;comparamos el num 3 con el 5

ja mayor ;declaramos salto condicional mayor

jb menor ;declaramos salto condicional menor

mayor: ;iniciamos la comparacion de mayor

lea dx,dato1 ;mete dato1 al registro "dx" y leelo

mov ah,09h ;imprimelo en pantalla

int 21h

jmp fin ;saltalo a fin

menor: ;iniciamos la comparacion de menor

lea dx,dato2 ;mete dato2 a dx y leelo

mov ah,09h ;impremelo en pantalla

int 21h

fin: ;termina el programa

mov ah,04ch

int 21h

end inicio ;finaliza el programa

**DESPLAZAMIENTO LOGICO A LA IZQUIERDA**

.model small ;define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los datos

res db " $" ;declaramos las variables de que tipo van a ser

.code ;inicializacion

inicio: ;inicia la programacion

mov ax,@data

mov ds,ax

mov ax,0

mov al,10001001b ;137 mueve al registro "al"

shl al,01 ; desplazame el contenido de al hacia la izquierda shl, izq shl

mov res,al ;el bit es enviado a la izquierda es enviado a la bandera de acarreo

;y el bit de mas a la derecha se llena con un cero el valor original se reduce a la mitad en numero decimal

lea dx,res ;mete res a dx y leelo

mov ah,09h ;imprime en pantalla

int 21h

mov ah,4ch ;termina el programa

int 21h

end inicio ;finaliza el programa

**DESPLAZAMIENTO LOGICO A LA DERECHA**

.model small ;define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los datos

res db " $" ;declaramos las variables de que tipo van a ser

.code ;declaramos el segmento del codigo

inicio: ;inicializacion

mov ax,@data

mov ds,ax

mov ax,0

mov al,10001001b ;137 mueve al registro "al"

shr al,01 ; desplazame el contenido de al hacia la derecha shr

mov res,al ;el bit es enviado a la derecha es enviado a la bandera de acarreo

;y el bit de mas a la izq se llena con un cero el valor original se reduce a la mitad en numero decimal

lea dx,res ;mete res a dx y leelo

mov ah,09h

int 21h

mov ah,4ch ;termina el programa

int 21h

end inicio ;finaliza el programa

**ALUMNO REPROBADO O APROBADO**

.model small ; define el modelo de memoria

.stack ;define la directiva de pila

.data ;define los datos

;aqui acontinuacion se declaran las variables

mat db 5

espa db 6

cien db 6

men1 db "alumno aprobado $"

men2 db "alumno reprobado $"

.code ;declaramos el segmento del codigo

inicio: ;inicia la programacion

mov ax,@data ;iniciar el programa

mov ds, ax

mov ax, 0

mov al,mat ;mueve mat a la instruccion al

mov bl,espa ;mueve espa a la instruccion bl

mov cl,cien ;mueve cien a la instruccion cl

add al,bl ;suma bl mas al y guardalo en al

add al,cl; ahora suma lo que se encuentra en al mas cl

mov bl,al ;mueve al a bl

mov cl,3 ;mueve 3 a registro cl

div cl ;divide la suma total entre 3

;el resultado total se almacena en la instruccion al

cmp al,7 ;compara el resultado con el 7

;declaramos saltos condicionales

ja a ;es para indicar que es mayor

jb r ;es para indicar que es menor

a: ;si es mayor a 7 aprobado

lea dx,men1 ;mete men1 a dx y leelo

mov ah, 09h ; imprimelo en pantalla

int 21h

jmp fin ;saltalo a fin

r: ;si es menor a 7 reprobado

lea dx,men2 ;mete men2 a dx y leelo

mov ah, 09h ;imprime en pantalla

int 21h

fin: ;termina el programa

mov ah,04ch

int 21h

end inicio;finaliza el programa

**MULTIPLICACION**

.model small ; define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los datos

res db "",0dh,0ah,"$" ; se declara la variable res

.code ;declaramos el segmento del codigo

inicio: ;inicia la programacion

mov ax,@data

mov ds, ax

mov ax, 0

mov al, 4 ; mueve el 4 registro al

mov bl, 2 ; mueve el 2 al registro bl

mul bl ;realiza la multiplicacion

;el resultado se guarda en el registro al

add al, 48 ;al resultado sumale 48

mov res,al ; mueve al a la variable res

lea dx, res ; lee la variable res y almacenala en el registro dx

mov ah, 09h ; imprime en pantalla

int 21h

mov ah,04ch ;termina el programa

int 21h

end inicio ;finaliza el programa

**DIVISION**

.model small ; define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los dato

res db "",0dh,0ah,"$" ; declaramos la variable res

.code ;declaramos el segmento del codigo

inicio: ;inicia la programacion

mov ax,@data

mov ds, ax

mov ax, 0

mov al, 4 ;el 4 se mueve al registro "al"

mov bl, 2 ;el 2 se mueve al registro "bl"

div bl ; se divide en 2 porque lo que tiene el registro "bl"

add al, 48 ;48 se suma al registro "al" esto se hace para que nos valla aparecer un numero correcto mas adelante

mov res,al ; se mueve el registro "al" ala variable res

lea dx, res ;mete res a dx y leelo

mov ah, 09h ; imprime en pantalla

int 21h

mov ah,04ch ;termina el programa

int 21h

end inicio ;finaliza el programa

**UN MENSAJE DE HOLA MUNDO**

.model small ; define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los datos

mensaje DB " HOLA MUNDO",0dh,0ah, "$" ; se declara la variable mensaje

.code ;declaramos el segmento del codigo

programa: ; inicia la programacion

mov ax,@data

mov ds,ax

mov ax,0

lea dx,mensaje ; mete mensaje a "dx" y leelo

mov ah,09h ; imprime en pantalla

int 21h

mov ah, 4ch ;termina el programa

int 21h

end programa ;finaliza el programa

**INTRODUCCION DE NUMEROS A LA PILA**

.model small ; define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los datos

res db 0, 0dh, 0ah, "$" ; se declara la variable res

.code ;declaramos el segmento del codigo

inicio: ;inicia la programacion

mov cx, 9 ;mueve el 9 al registro cx

add al,48 ;sumale 48 al registro al

ciclo: ;comienza el ciclo

inc al ;es un incremento de uno en uno

mov res,al ;mueve al ala variable res

lea dx,res ; lee lo que se encuentra en la variable res y alamacenalo en el registro dx

mov ah,0eh ;muestra el resultado en pantalla

int 10h

;es para formar la pila con el push y el pop

push cx ; mete datos en la pila

pop cx ; saca datos de la pila

loop ciclo ; es loop es un ciclo significa que mete el 1 y luego 2 y asi sucesivamente

mov ah,04ch ; termina el programa

int 21h

end inicio ; finaliza el codigo

**MUESTRAME UN CARÁCTER**

.model small ; define el modelo de memoria

.stack ; define la directiva de pila

.data ; define los datos

dato1 db 125 ; se declara la variable dato1

res db "", 0dh, 0ah, "$" ; declarar las variables tipo caracter

.code ; indica la codificacion del programa

inicio: ;para comenzar nuestro programa

mov ax,@data ;iniciar un programa

mov ds,ax

mov ax,0

mov al,dato1 ; nueve la variable dato1 al registro "al"

mov res, al ;mueve el registro "al" a la variable de caracter "res"

lea dx, res ;imprime la variable "res"

mov ah, 09h ; imprime en pantalla

int 21h

mov ah, 4ch ; termina el programa

int 21h

end inicio ;fin del programa

**ROTACION A LA DERECHA**

.model small; define el modelo de memoria

.stack; define la directiva de pila

.data; define los datos

res db " $" ;declaramos las variables de que tipo van a ser

.code ;declaramos el segmento del codigo

inicio: ;inicia la programacion

mov ax,@data ; iniciar el programa

mov ds,ax

mov ax,0

mov al,10001001b;137 mueve al registro al

ror al,01 ; ror rotame el bit de mas a la derecha del al a la posicion vacante de mas a la izq

mov res,al ; mueve el registro "al" ala variable res

lea dx,res ;mete res a dx y leelo

mov ah,09h ; imprime en pantalla

int 21h

mov ah,4ch ;termina el programa

int 21h

end inicio ;finaliza el programa

**RESTA**

.model small ;define el modelo de memoria

.stack ;define la directiva de pila

.data ;define los datos

res db "",0dh,0ah,"$" ; declaramos la variable como res

.code ;indica la codificacion del programa

inicio: ;para comenzar el programa

mov ax,@data

mov ds, ax

mov ax, 0

mov al, 4 ;4 muevelo al registro "al"

mov bl, 2 ;2 muevelo al registro "bl"

sub al,bl ; restale 2 que es el registo "bl" a 4 que es el registro "al"

;esto dice restar 2-4

add al, 48 ;ahora suma 48 al registro "al" para que nos llege a mostrar un numero coherente

mov res,al ; mueve el registro "al" a la variable res

lea dx, res ; mete res a dx y leelo

mov ah, 09h ;imprime en pantalla

int 21h

mov ah,04ch ;termina el programa

int 21h

end inicio ;finaliza el programa

**SUMA**

.model small ;define el modelo de memoria

.stack ;define la directiva de pila

.data ;define los datos

res db "",0dh,0ah,"$" ; declaramos la variable como res

.code ;indica la codificacion del programa

inicio: ;para comenzar el programa

mov ax,@data

mov ds, ax

mov ax, 0

mov al, 4 ;4 muevelo al registro "al"

mov bl, 2 ;2 muevelo al registro "bl"

add al,bl ;sumale 2 que es el registo "bl" a 4 que es el registro "al"

;esto dice sumar 2-4

add al, 48 ;ahora suma 48 al registro "al" para que nos llege a mostrar un numero coherente

mov res,al ; mueve el registro "al" a la variable res

lea dx, res ; mete res a dx y leelo

mov ah, 09h ;imprime en pantalla

int 21h

mov ah,04ch ;termina el programa

int 21h

end inicio ;finaliza el programa

**PULSE UNA TECLA PARA CERRAR**

name "mycode"

org 100h

mov ax, 1003h

mov bx, 0

int 10h

mov ax, 1234h

mov dx, 0x1234

mov ax, 1234

mov ax, 0ABCDh ;

mov dx, 0xABCD ; or use 0x prefix.

mov dl, TeSt1

jmp n

test1 db 20h

n:

add dl, dl

inc dl

mov ah, 2

int 21h

mov dl, 1

int 21h

mov al, 11100101b

xor al, 1111\_1110b

mov dx, 0705h

mov bx, 0

mov bl, 10011111b

mov cx, msg\_size

mov al, 01b

mov bp, offset msg

mov ah, 13h

int 10h

mov ah, 0

int 10110b

ret

msg db "press any key..."

msg\_size = $ - offset msg