

Ejercicios Sistema basado en microprocesadores

P1) Suponiendo $CS=0000h$, $DS=1000h$, $ES=FFFFh$, $SS=2000h$, $BX=2222h$, $BP=0000h$ y $SI=0002h$ indican la dirección física a la que se está accediendo en las instrucciones

$MOV AH, ES:[SI]$ $ES \times 16 + SI = FFFFh + 0002h = 00002h$

$MOV AH, [SI]$ $DS \times 16 + SI = 10000h + 0002h = 10002h$

$MOV AL, [BP-2]$ $SS \times 16 + BP - 2 = 20000h + FFFEh = 2FFFEh$

$MOV AL, CS:[FFFFh]$ $CS \times 16 + FFFFh = 00000h + FFFFh = FFFFh$

P2) Suponiendo que $CS=2000h$, $DS=424Eh$, $ES=4240h$, $SS=4240h$, $BX=0$, $BP=3$, $DI=3$, $SP=30$ y $AX=1234h$ indica el valor en hexadecimal de los 16 primeros bytes del segmento DS

$mov 2[BX][DI], AH$ $DS \times 16 + 2 + BX + DI = 424E0h + 2 + 0 + 3 = 424E5h$

$mov DS:[BP][DI], AX$ $DS \times 16 + BP + DI = 424E0h + 3 + 3 = 424E6h$

$mov 22[BP], AX$ $SS \times 16 + 22 + BP = 424D0h + 16h + 3h = 424E9h$

$push AX$ $SS \times 16 + SP = 424D0h + 1E = 424EEh$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				12	34	12			34	12		34	12		

P3) Suponiendo $CS=2000h$, $DS=424Eh$, $ES=4240h$, $SS=424Eh$, $BX=0004h$, $BP=0003h$ y $DI=0002h$ indican el valor de AX, los dígitos decimales = P

424E:0000 4E 42 74 61 20 85 73 20

$MOV AX, 1[BX] + 5h$ $AX = 7365h$

$MOV AX, 0005h$ $AX = 7365h$

$MOV AL, ES:[BP+16]$ $AX = ?P61h$

$MOV AH, [BP+3]$ $AX = 73??h$

$MOV AX, ES:14[BX][DI]$ $AX = 6520h$

$424D0 + 14 + 0004h + 0002h = 424E4$

14) Suponiendo CS=2000h, DS=424Eh, ES=424Dh, SS=424Fh
 BX=0009h, BP=0003h y DI=FFFFh, indicar el valor del registro
 AX tras las siguientes operaciones

424E:0000 4E 42 74 01 20 65 73 20

MOV AX, [BX][DI] $424E0h + 0009h + FFFFh = 424E0h + 0003h = 424E3h$ AX=2061h

MOV AX, [000h] AX=424Eh

MOV AL, DS:[BP] $424E0h + 0003h = 424E3h$ AX=??61h

MOV AH, [BP]13 $424E0h + 0003h + 13 = 424F0$ AX=????h

MOV AX, 2[DI] $424E0h + 2 + FFFFh = 424E0h + 0001h = 424E1h$ AX=7442h

P5) Suponiendo que CS=2000h, DS=1000h, ES=1234h, SS=4321h y
 BX=5432h indicar la dirección física a la que acceden
 en cada una de las siguientes instrucciones

MOV AH, [BX] @ = 1000h + 5432h = 15432h

MOV AX, 3[BX] @ = 1000h + 3h + 5432h = 15435h

MOV AX, ES:[BX+3] @ = 12340h + 5432h + 3h = 17775h

MOV AL, [1000h] @ = 1000h + 1000h = 11000

P6) Escribir una secuencia de instrucciones de ensamblador
 para leer sobre AX una palabra de 16 bits almacenada
 en la dirección física E256h

mov AX, E256h Dirección = DS*16 + []

mov DS, AX

mov AX, [AX]

P7) Suponiendo CS=2000h, DS=193Fh, ES=193Eh, SS=2222h y
 BX=0001h, indicar el valor del registro AX

193F:0000 CD 20 FF 9F 00 9A F0 FE
 $193F0h + 1h = 193F1h$

MOV AX, [BX] AX=FF20h

MOV AH, 3[BX] $193F0h + 3 + 1h = 193F4h$ AX=9A00h

MOV AL, ES:[BX+20] $193E0h + 0001h + 20$ AX=??9Ah

MOV AX, ES:[10h] $193E0h + 10h = 193F0h$ AX=20CDh

P8) Indicar el valor de la constante TMP dado el seg. fragmento

Valores DW 4, 5*9, 10h+2*34, 23h, 'A'

TMP EQU (\$-Valores) TMP=10

P9) Suponiendo que CS=1000h, DS=2000h, ES=4321h, SS=1111h, BX=2222h, BP=3333h y SI=0002h, indicar los direcciones a las que acceden

(1) MOV AH, 4[BX][SI] @ = $CS \times 16 + BX + SI = 2222h$
 MOV AH, 55[BP][SI] @ = $SS \times 16 + BP + SI = 14447h$
 MOV AL, [BP+4] @ = $SS \times 16 + 4 = 14447h$
 MOV AL, [CS:1000h] @ = $CS \times 16 + 1000h = 11000h$

P10) Suponiendo que CS=0000h, DS=1000h, ES=FFFFh, SS=2000h, BX=2222h, BP=0000h y SI=0002h, indicar las direcciones

MOV AH, ES:16[SI] $FFFFh \times 16 + SI = FFFF02h = 00002h$
 MOV AH, 16[SI] $10000h + 16 + SI = 100002h = 10002h$
 MOV AL, [BP-2] $20000h + FFFE = 2FFFFh$
 MOV AL, CS:[FFFFh] $00000h + FFFFh = 0FFFFh$
 MOV AL, DS:[BP-1] $10000h + FFFFh = 1FFFFh$

P11) Suponiendo CS=2000h, DS=204Fh, ES=204Fh, SS=2000h, BX=0001h, BP=04F8h, DI=0007h y SP=04F8h, indicar el valor del registro AX con las siguientes instrucciones independientes entre sí

204F: 0000 73 65 67 20 00 08 01 6E

204F: 0008 12 34 4E 00 FF 00 33 11

MOV AH, [BX][DI] = $DS \times 16 + BX + DI = 204F8$ AX = 12FFh
 MOV AL, 3[DI] = $DS \times 16 + 3 + DI = 204FA$ AX = 004Eh
 MOV AX, [BP-6] $SS \times 16 + BP - 6 = 2000h + 04F8h$ AX = 2067h
 pop AX $SS \times 16 + SP = 20000h + 04F8h = 24F8h$ AX = 3212h
 mov AX, 16[BX] $DS \times 16 + 16 + 0001h = 204F0h$ AX = FFFFh
 $= 204F0h + 10h + 0001h = 20501$

P12) Suponiendo CS=2000h, DS=424Eh, ES=424Eh, SS=424Eh, BX=0, BP=3, DI=3, SP=8 y AX=1234h, indicar el valor hexadecimal de las 16 primeras bytes del registro DS

MOV SS [BX][DI], AH $424Eh \times 16 + BX + DI = 424E3$
 MOV DS: [9], AX $424Eh \times 16 + 9 = 424E09h = 424E9h$
 MOV [BP+1], AX $SS \times 16 + BP + 1 = 424E4$
 push ES $SS \times 16 + BP = 424E0h + 3h = 424E3h$

424E0h 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 12h 40h 42h 34h 12h 34h 12h

- P17) Declarar mediante directivas de ensamblador de 8086 con variables que se describen
- ;(tabla1) Tabla de 12 palabras de 16 bits inicializadas a 0
tabla1 dw 12 dup(0)
 - ;(contador) Entero de 4 bytes sin inicializar
contador dd ?
 - ;(tabla2) Tabla de 255 elementos, donde cada elemento es el carácter 'A' seguido de un entero de 2 bytes inicializado a FFFFh
tabla2 255 dup('A', OFFh, OFFh)
 - ;(mensaje) Cadena "Fichero inexistente" seguida de los valores 10 y 13
mensaje db "Fichero inexistente", 10, 13
 - ;(secontador) Entero de 2 bytes inicializado con el segmento de la variable "contador"
secontador dw SEG contador

P18) Teniendo en cuenta la sección de código de la izquierda, implementar cada una de las operaciones solicitadas en el cuadro de la derecha mediante una única instrucción

datos segment
cadena db "Adiós", 13, 10
longitud db \$-cadena
datos ends

res segment
resultado db 200 dup(?)
contador dw ?
res ends

código segment
assume cs: código, ds: datos
inicio proc far
mov ax, código
mov ds, ax
mov ax, datos
mov cx, ax
...
mov ax, 4C00h
int 21h
inicio endp
código ends
end inicio

; leer en AX la variable "longitud"
mov AX word PTR longitud
AX es word y longitud es byte,
Word PTR es como un casting
para que coincidan los tipos

; leer en BX la variable contador
No es posible en una única instrucción ya que el segmento res no está cargado

Todo lo que sea acceder a valores de dentro del segmento res requieren más de una instrucción
; leer en AX la posición de la tabla "cadena" indicada por SI
MOV AX, word PTR ES:cadena[SI]

P22) Dado el código indicar las instrucciones en ensamblador

```

datos segment
    cadena dw "Hola"
    longitud db ?
datos ends
resultados segment
    resultado db 20 dup (?)
    contador dw 0
resultados ends
codigo segment
    assume cs:codigo, ds:datos
    inicio proc far
        mov ax, resultado
        mov ds, ax
        mov ax, datos
        mov es, ax
        ...
        mov ax, 4000h
        int 21h
    inicio endp
codigo ends
end inicio

```

; leer en AL la variable "longitud"
 MOV AL, es:[longitud]
 ; leer en BX la variable "contador"
 MOV AX, contador
 ; Escribir en la tabla "resultado"
 el código ASCII de la letra X
 en la posición indicada por DI
 MOV resultado[di], 'X'
 ; Escribir en la tabla "resultado"
 el valor 65535 = FFFFh = 2 bytes
 MOV word PTR resultado, 65535
 ; leer en PX la posición de la
 tabla cadena indicada por BX
 MOV DX q:cadena[BX]

P23) Suponiendo que CS=2000h, DS=4000h, ES=424Dh, SS=424Eh, BX=0004h, BP=0000h y PI=24E0h indicar el valor de AX con cada instrucción

424E: 0000 4E 42 24 61 20 65 73 20

MOV AX, [BX] DS*16+BX=40004h AX=?
 MOV AX, DS:[0000h] DS*16+0=40000h AX=?
 MOV AH, [BP+3] SS*16+3=424E3h AX=?
 MOV AL, ES:[BP] ES*16+15=424DFh AX=?
 MOV AX, 2[DI] DS*16+2*DI=40000+2*24E0h=424E2h AX=?

P26) Suponiendo que CS=4200h, DS=424Eh, ES=424Dh, SS=424Eh, BX=0, BP=3, PI=3, SI=04ECh y AX=1234h, indicar el valor de los 16 primeros bytes del segmento DS

MOV CS:04ECh[BX], AX 4200*16+04ECh+0=424E0
 MOV SS:[DI]1, AH SS*16+DI+1=424E4h
 MOV DS:[BP][DI]2, AX DS*16+BP+DI+2=424E8
 MOV CS:[SI], AL CS*16+04EC=424EC
 MOV ES:28[DX][DI], AX ES*16+28d+DX+DI=424EF

424E0h 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

34 42 12 34 39 12 34 39

P27) Declarar mediante directivas de ensamblador de 80x86 en variables que se describen

@(Tabla1) Tabla de 128 bytes sin inicializar

Tabla1 db 128 dup(?)

ⓑ(contador) entero de 2 bytes inicializado a -1
contador dw -1

Ⓒ(Tabla2) Tabla de 100 elementos, donde cada elemento es una tabla de 50 palabras de 16 bits inicializadas a 0

Tabla2 dw 100 dup(50 dup(0))

Ⓓ(Mensaje) Cadena "Archivo inexistente" seguida del carácter '\$'

mensaje db "Archivo...", '\$' == mensaje db "Archivo..."

Ⓔ(pcontador) dirección larga de la variable "contador"
pcontador dd contador

P28) Teniendo en cuenta el código de la izquierda implementa el código de la derecha

datos segment

Tabla db 1, 2, 3, 4, 5

dw ?

datos ends

res segment

resultado db 100

dw ?

res ends

código segment

assume cs:datos, es:res

Inicio proc far

mov ax, datos

mov ds, ax

mov ax, res

mov es, ax

proc

fin

Seen en AX los dos primeros valores de "Tabla"

mov AX, word ptr ds:Tabla

Escribir en "v" el entero contenido a partir del segmento de memoria con offset FFFEH

P29 Suponiendo que CS=2222h, DS=1234h, ES=F000h, SS=3333h, BX=1111h, BP=0006h, SI=0004h y DI=0003h, indicar dir física

mov AH, [DI] CS*16+DI=22223h

mov AX, CS:[DI] CS*16+7+DI=2222Ah

mov AL, [SI][DI] Incorrecto SI y DI no pueden estar juntos

mov AX, CS:[BP][DI] CS*16+BP+DI=22229h

P30) Declarar mediante `directiv` en ensamblador 80x86 las variables descriptas a continuación
;(tabla1) Tabla de 256 enteros de 16 bits sin inicializar
16bits \rightarrow 2 bytes

Tabla 1 dw 256 dup (?)

; contador de 4 bytes inicializado a -1
contador dd -1

;(tabla2) Tabla de 50 elementos, donde cada elemento es una tabla de 50 bytes inicializados a 0
Tabla 2 db 50 dup (50 dup (0))

;(mensaje) cadena "Fichero inexistente" seguida del carácter '\$'
mensaje db "Fichero inexistente \$" == mensaje db "Fichero..."; '\$'

;(contador) Dirección larga de la variable "contador"
pcontador dd contador

(P31) Suponiendo que CS=2000h, DS=424Ch, ES=424Eh, SS=424Fh, BX=0002h, BP=0003h y SI=0003h indican AX

424E:0000 4E 42 74 61 20 65 73 20

MOV AL, 35[BX] $DS \times 16 + 35d + BX = 424E5$ AX = ?? 65 h

MOV AX, SS:[0007h] $SS \times 16 + 7 = 424E7$ AX = ?? 20 h

MOV AH, [BP+35] $DS \times 16 + BP + 35d = DS \times 10 + 26 = 424E6$ AX = 73 ?? h

MOV AX, DS:[BP+35] $DS \times 16 + BP + 35d = 424E6$ AX = 20 73 h

MOV AX, ES:[BX][CS] $ES \times 16 + BX + SI = 424E5$ AX = 73 65 h

P33) Suponiendo que CS=1234h, DS=1234h, ES=FFFFh, SS=1111h, BX=1111h, BP=2222h, SI=0002h y DI=0004h

MOV AH, [SI+16] $DS \times 16 + SI + 16 = 12340h + 0002h + 10d = 12352h$

MOV AX, ES:[SI+16] $ES \times 16 + SI + 10 = FFF12h$

MOV AL, [SI][DI] Incorrecto

MOV AX, CS:[SI][BP] $CS \times 16 + SI + BP = 14564h$

MOV AL, ES:[DI] FFF04h

P39) CS=1234h, DS=1234h, ES=FFFFh, SS=2222h, BX=0010h, BP=0011h, SI=0002h y DI=0004h

MOV AX, [BX] 12340h + 010h = 12350h

MOV AH, ES:[DI+10] FFFF0h + DI + 10 = FFFFEh

MOV AX, [BP][DI] 22220h + 0011h + 0004h = 22235h

MOV AL, DS:[SI][BP] 12340h + 2h + 11h = 12353h

MOV AX, ES:[DI] FFFF0h + 0004h = FFFF4h

P4C) Declarar mediante directivas de ensamblador de 80x86 las variables que se describen a continuación

(tabla 1) Tabla de 1000 enteros de 32 bits inicializados a -1
32 bits = 4 bytes

tabla1 dd 1000 dup (-1)

(contador) Entero de 16 bits inicializado a 0
contador dw 0

(tabla 2) Tabla de 200 elementos, donde cada elemento es un byte inicializado a 0 y una tabla de 20 enteros de 16 bits inicializados a 00FFh

tabla2 db 200 dup (0, 20 dup (0FFh, 0))
directa a 0 ← tabla de 20 enteros

(mensaje) cadena "Error de comunicación" seguida del carácter '\$'

mensaje db "Error de comunicación" '\$'

(pmensaje) Dirección lógica de la variable "mensaje"
pmensaje dd mensaje

Teniendo en cuenta la secuencia de código de la prog. implementar lo que se pide a la derecha

datos segment

w dw ?

datos ends

res segment

resultado db 100 dup (?)

res ends

código segment

assume cs: código, ds: datos,

es: res

tabla db "Hello"

v dw \$-tabla

inicio proc far

mov ax, código

mov ds, ax

mov ax, datos

mov es, ax

...

mov ax, 4C00h

int 21h

inicio endp

código ends

end inicio

; Escribir en AX el tamaño de la "tabla", sin emplear direccionamiento inmediato

MOV AX CS:[v]

; Escribir en "w" el tamaño de "tabla" sin usar direccionamiento inmediato

No se puede pasar de memoria a memoria en una única instrucción mov

; Inicializar "w" a 1234h

MOV ES:[w], 1234h

; Escribir en BP el valor almacenado en w

MOV bp, ES:[w]

; Escribir en el último byte de resultado el valor 0

No se puede, el segmento res no está cargado en registro puntero de segmento

P73) Suponiendo que CS=4000h, DS=424Ah, ES=424Bh, SS=424Ah
 BP=0010h, SI=0006h y DI=24B0h indicar el valor de AX

424A:0000 65 D3 CC 44 DE CS F4 42 37 FF D4 E5 00 0A 23 00
 424A:0010 34 55 A2 23 81 99 AB 1F 12 AA 0A 22 FF FF 98 7E

MOV AX, [SI] $DS \times 16 + SI = 424A6h$ AX = 42F4h
 MOV AH, CS: [DI] $CS \times 16 + DI = 424A9h$ AX = AA??h
 MOV AL, SS: [BP][SI] $SS \times 16 + SI = 424A0h$ AX = ???Ah
 MOV AX, CS: [24AEh] $CS \times 16 + 24AEh = 424AEh$ AX = 0023h
 MOV AX, ES: [BP] $ES \times 16 + BP = 424B0h$ AX = P??Ph

P44) Suponiendo CS=4321h, DS=4321h, ES=0000h, SS=2222h,
 BX=1000h, BP=0010h, SI=0004h y DI=0002h

MOV AX, [SI], [DI] $DS \times 16 + SI + DI = 43216h$ 16 repuede usar DI y SI juntos
 MOV AH, CS: [BX][DI] $CS \times 16 + 1000h + 0002h = 44212h$
 MOV AX, DS: [BP][SI] $DS \times 16 + 0010h + 0004h = 43224h$
 MOV AL, ES: [SI+0FFFFh] $ES \times 16 + 3h = 00003h$
 MOV AX, SS: [DI] $SS \times 16 + 0002h = 22222h$

P46) Declaran mediante directivas de ensamblador

- (tabla1) Tabla de 1000 enteros de 8 bits inicializados a 100
 tabla1 db 1000 dup(100)
- (contador) Entero de 32 bits sin inicializar
 contador dd ?
- (tabla2) Tabla de 100 elementos, donde cada elemento es un entero de 32 bits inicializado a -1, la cadena de caracteres "12345" y un entero de 16 bits sin inicializar
 tabla2 db 100 dup(32 dup(-1), "12345", 16 dup(?))
- (mensaje) Cadena "ERROR" seguida de un byte inicializado a 0
 mensaje db "ERROR", 0
- (tabla2) desplazamiento (offset) de la variable "tabla2"
 ptabla2 dw tabla2