



Examen de Laboratorio de Estructura de Computadores
I.T. Informática de Gestión / Sistemas. 21 de enero de 2008
Departamento de Automática – ATC –

Apellidos, Nombre:

Grupo de laboratorio: Gestión Sistemas Libre Elección

Pregunta correcta=0,5 Pregunta no contestada=0 Pregunta incorrecta (tipo test)=- 0,15

1. Son funciones de un sistema operativo:

- Definir la interfaz de usuario.
- Planificar recursos.
- Las dos respuestas anteriores son ciertas.
- Descodificar las instrucciones.

2. En Windows 2000 los permisos para compartir un directorio:

- se pueden asignar a grupos o a usuarios.
- son los mismos que tiene el ordenador local.
- no se pueden asignar. Este sistema operativo no permite esta operación.
- eliminan todos los ficheros.

3. ¿Cuál de los siguientes nombres de fichero no está permitido en MS-DOS?

- Ficheros.txt
- Libros.exe
- 2archivo.doc
- documents.asm

4. En MS-DOS, la orden *dir *.doc* muestra

- todo el contenido del directorio actual.
- todos los ficheros de texto del directorio actual.
- todos los ficheros con la extensión doc del directorio actual.
- el fichero de nombre *.doc.

5. ¿Cuál de las siguientes secuencias de órdenes permite crear un fichero ejecutable de nombre *programa.exe* a partir del fichero *programa.asm*?

- 1º *masm programa*; 2º *link programa*; 3º *cv programa*
- 1º *masm programa.asm*, 2º *link programa.exe*
- 1º *masm programa.asm*, 2º *link programa.obj*.
- 1º *masm programa.asm*; 2º *link programa.obj*; 3º *cv programa.exe*

6. El microprocesador 8086 tiene:

- 4 registros de 16 bits.
- 13 registros de 16 bits.
- 14 registros de 16 bits.
- 8 registros de 8 bits y 9 de 16 bits.



Apellidos, Nombre:

Grupo de laboratorio: Gestión Sistemas Libre Elección

7. Para modificar el contenido del registro AX dentro de CODEVIEW se debe indicar en la línea de comandos la orden:

- EB AX nuevo_contenido
- EW AX nuevo_contenido
- WW AX nuevo_contenido
- R AX nuevo_contenido**

8. ¿Qué registros del microprocesador 8086 se pueden utilizar para realizar direccionamientos relativos?

- AX, BX, CX, y DX
- SI, DI y AX
- BX, BP y DX
- BX, BP, SI y DI**

Programación en Ensamblador (6 puntos)

1) El código fuente que se muestra a continuación es de un programa que convierte un número decimal introducido por el teclado en su correspondiente representación hexadecimal, presentando el resultado en la pantalla.

```
DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
    MsjEnt      DB '¿Número decimal (0...65535)? : ',10,13,'$'
    Buffer       DB 6
    CarLei      DB ?
    Cadena      DB 6 DUP(?)
    Numero      DW ?
    MsjSal      DB 10,13,'Su representación hexadecimal es: '
    NumSal      DB 4 DUP('0'),'','$'
.CODE
Inicio:    mov ax,@DATA
           mov ds,ax
           ;Entrada del número decimal por el teclado
           Zona 1
           ;Conversión de la cadena ASCII del número decimal a valor
           Zona 2
           ;Conversión del valor a hexadecimal codificado en ASCII
           Zona 3
           ;Presentación del resultado en la pantalla
           Zona 4
           ;Fin del programa
           mov ah,4Ch
           int 21h
           END Inicio
```



Examen de Laboratorio de Estructura de Computadores
I.T. Informática de Gestión / Sistemas. 21 de enero de 2008
Departamento de Automática – ATC –

Apellidos, Nombre:

Grupo de laboratorio: Gestión Sistemas Libre Elección

- a) Para la **zona 1** del programa, escribe el código necesario para realizar la entrada del número por el teclado, utilizando las definiciones de posiciones de memoria: `MsjEnt`, `Buffer`, `CarLei` y `Cadena`.

```
mov ah,9
lea dx,MsjEnt
int 21h
mov ah,0Ah
lea dx,Buffer
int 21h
```

- b) Para la **zona 2** del programa, escribe el código necesario para convertir la cadena ASCII del número decimal leída del teclado almacenada en la posición de memoria `Cadena`, en su correspondiente valor de 16 bits que debe quedar almacenado en la posición de memoria `Numero`.

```
xor ax,ax
lea bx,Cadena
xor ch,ch
mov cl,CarLei
xor dx,dx
mov di,10
Lazo1:   mul di
         mov dl,[bx]
         sub dl,30h
         add ax,dx
         inc bx
Loop lazo1
mov Numero,ax
```

- c) Para la **zona 3** del programa, escribe el código necesario para convertir el valor almacenado en la posición de memoria `Numero` en su valor hexadecimal codificado en ASCII que debe quedar almacenado en la posición `NumSal`.

```
mov cx,4
lea bx,NumSal
Lazo2:   and ax,0Fh
         cmp al,9
         jbe EsNumr
         add al,7
EsNumr:  add al,30h
         mov [bx+3],al
         dec bx
         push cx
         mov cl,4
         mov ax,Numero
         shr ax,cl
         mov Numero,ax
         pop cx
Loop Lazo2
```



Examen de Laboratorio de Estructura de Computadores
I.T. Informática de Gestión / Sistemas. 21 de enero de 2008
Departamento de Automática – ATC –

Apellidos, Nombre:

Grupo de laboratorio: Gestión Sistemas Libre Elección

d) Para la **zona 4** del programa, escribe el código necesario para mostrar en la pantalla el resultado incluyendo el texto previo de presentación almacenado en la posición de memoria `MsjSal`.

```
mov ah,9
lea dx,MsjSal
int 21h
```

2) El programa mostrado a continuación suma dos números de 32 bits almacenados en memoria y devuelve el resultado en la memoria. Para realizar la operación, el programa introduce en la pila los dos números y la dirección del resultado `Res1`.

```
DOSSEG
.MODEL SMALL
.STACK 500h
.DATA
    Num1 DW 1A23h,8FC1h
    Num2 DW 3C12h,4CFAh
    Res1 DW 2 DUP (0)
    Carr DW 0
.CODE

;Procedimiento que suma dos números de 32 bits

    ;Código fuente del procedimiento

;Programa principal
Inicio: mov ax,@DATA
        mov ds,ax
        lea bx,Num1
        mov cx,4
        Lazo: mov ax,[bx]
                push ax
                inc bx
                inc bx
        Loop Lazo
        lea ax,Res1
        push ax
        call Suma32
        mov ah,4Ch
        int 21h
        END Inicio
```

a) Escribe el código fuente del procedimiento `Suma32` que suma los datos pasados a través de la pila y devuelva el resultado en la dirección `Res1` proporcionada por el mismo método. Además debe devolver en la dirección `Carr` el estado del flag de acarreo al finalizar la suma.

```
Suma32 PROC
    push bp
    mov bp,sp
    push ax
    push bx
    push dx
    xor dx,dx
    mov bx,[bp+4] ;Dirección de Res1
```



**Examen de Laboratorio de Estructura de Computadores
I.T. Informática de Gestión / Sistemas. 21 de enero de 2008
Departamento de Automática – ATC –**

Apellidos, Nombre:

Grupo de laboratorio: Gestión Sistemas Libre Elección

```

mov ax, [bp+12]
add ax, [bp+8]
mov [bx], ax
inc bx
inc bx
mov ax, [bp+10]
adc ax, [bp+6]
rcl dx, 1                    ;Posible acarreo a DX
mov [bx], ax
inc bx
inc bx
mov [bx], dx
pop dx
pop bx
pop ax
pop bp
ret 10

```

Suma32 ENDP

- b) Indica el contenido de la pila después de ejecutarse la instrucción anterior a la instrucción **call Suma32**, sabiendo que la variable **Res1** está almacenada en la dirección 4184:0012 y suponiendo que la estructura de la pila al comenzar la ejecución del programa es la siguiente:

SS:SP →4184:0500	
4184:04FF	1A
4184:04FE	23
4184:04FD	8F
4184:04FC	C1
4184:04FB	3C
4184:04FA	12
4184:04F9	4C
4184:04F8	FA
4184:04F7	00
4184:04F6	12
4184:04F5	
4184:04F4	
4184:04F3	
4184:04F2	

PUNTUACIÓN DE LA PROGRAMACIÓN EN ENSAMBLADOR

Ejercicio 1: a) 0,75 puntos b) 1,25 puntos c) 1,75 puntos d) 0,25 puntos
Ejercicio 2: a) 2,5 puntos b) 1 punto