

SISTEMAS BASADOS EN MICROPROCESADOR (SBM) (2º)

Grado en Ingeniería Informática
Escuela Politécnica Superior – UAM

Parcial 1 - Curso 2015/16 MODELO A

NOMBRE : _____ SOLUCION _____ DNI : _____
APELLIDOS : _____

P1. Suponiendo que **CS=2000h**, **DS=3000h**, **ES=4000h**, **SS=5000h**, **BX=1500h** y **BP=1500h** indicar la **dirección física** de memoria (@) a la que se está accediendo en cada una de las siguientes instrucciones, considerando los registros de segmento por defecto. (1 punto)

<code>mov AH, [BX]</code>	@ = 31500h
<code>mov AX, [BP + 2h]</code>	@ = 51502h
<code>mov AX, ES: [BX + Ah]</code>	@ = 4150Ah
<code>mov AL, CS: [3500h]</code>	@ = 23500h

P2. Suponiendo que **CS=2000h**, **DS=250Fh**, **ES=250Eh**, **SS=3000h** y **BX=0004h**, Indicar el valor del **registro AX** tras ejecutar cada una de las instrucciones siguientes (**independientes entre si**), dado el volcado de memoria adjunto. Expresar los **dígitos hexadecimales desconocidos de AX con un '?'**. (1.5 puntos)

250F:0000 10 2A 33 8D 01 EE DF FF A0 B5 67 28 23 F5 78 12

<code>mov AX, [BX]</code>	AX = EE01h
<code>mov AH, 2[BX]</code>	AX = DF??h
<code>mov AL, ES: [BX + 19]</code>	AX = ??FFh
<code>mov AX, ES: [10h]</code>	AX = 2A10h

P3. Si **SP=0006h**, **FLAGS=0200h** y **BX=FA69h** al inicio de la ejecución del código que se adjunta, indicar los valores contenidos en las **primeras seis posiciones de la pila** tras ejecutar la primera instrucción (**push bx**) del procedimiento **SumarDatos**, tanto cuando todos los procedimientos del programa son cercanos (**NEAR**), como cuando son lejanos (**FAR**). No escriba nada en aquellas direcciones de la pila que no se vean afectadas. (1.5 puntos).

```
5000:025E E8A8FD    call SumarDatos
5000:0261 89161000    mov word ptr Tabla[5], ax
```

0	1	2	3	4	5
		69h	FAh	61h	02h

Caso NEAR

0	1	2	3	4	5
69h	FAh	61h	02h	00h	50h

Caso FAR

P4. Si **SP=0008h**, **FLAGS=0200h**, **AX=5476h** al inicio de la ejecución del código que se adjunta, indicar los valores contenidos en las **primeras ocho posiciones de la pila** en el momento de ejecutar la primera instrucción (sin ejecutarla) de la rutina de servicio de la interrupción 51h. No escriba nada en aquellas direcciones de la pila que no se vean afectadas. (1.5 puntos)

0	1	2	3	4	5	6	7
		60h	03h	00h	50h	00h	02h

5000:035E CD61 int 51h
5000:0360 89161000 mov Datos[0], ax

P5. Indicar el contenido (expresado como segmento:offset) del vector de interrupción 51h dado el siguiente volcado de memoria. (1.5 puntos)

0000:0130 54 02 CF 15 CE 01 CF 15 04 00 70 00 D7 01 CF 15
0000:0140 04 00 70 00 30 00 00 50 30 00 00 C8 30 00 00 C8

Segmento = 5000h Offset = 0030h

¿Qué es ese valor?

Dirección de memoria donde se encuentra la (primera instrucción) de la rutina de servicio de vinculada a ese vector de interrupción.

P6. Escribir en ensamblador una rutina **Multiplicar** (NEAR) que lleve a cabo la multiplicación de 2 operandos de 16 bits sin signo (BX y CX) devolviendo el resultado en DX_AX. La operación de multiplicación debe llevarse a cabo mediante instrucciones suma (ADD, ADC). (3 puntos)

```

MULTIPLICAR    PROC
                MOV AX, #0
                MOV DX, #0
                CMP CL, #0
                JZ SAVERES
LOOPCL         ADD AX, BL
                ADD DX, BH
                DEC CL
                CMP CL, #0
                JNZ LOOPCL
                ADD DX, AH
                MOV AH, DL
                ASR DX, #8
SAVERES        PUSH AX DX
                MOV AX, #0
                MOV DX, #0
                CMP CH, #0
                JZ FIN
LOOPCH         ADD AX, BL
                ADD DX, BH
                DEC CH
                CMP CH, #0
                JNZ LOOPCH
                ADD DX, AH
                MOV AH, DL
                ASR DX, #8
                POP CX BX
                ASL DX, #8
                MOV DL, AH

```

```
ASL AX, #8
ADD AX, BX
ADC DX, CX
RET
MULTIPLICAR ENDP
```