

FACULTAD:	<b>Tecnología Informática</b>		
CARRERA:			
ALUMNO/A:			
SEDE:		LOCALIZACIÓN:	<b>Buenos Aires</b>
ASIGNATURA:	<b>Sistemas de Computación II</b>		
COMISIÓN:		TURNO:	
PROFESOR:		FECHA:	
TIEMPO DE RESOLUCIÓN:	<b>120 min</b>	EXAMEN PARCIAL N°:	<b>2 (dos)</b>
MODALIDAD DE RESOLUCIÓN:	A Distancia Sincrónico - Escrito		
CALIFICACIÓN:			
RESULTADOS DE APRENDIZAJE ESPERADO: <b>Asegurar la comprensión de Interrupciones, manejo de la pila y programación en Assembler.</b>			

Dado el siguiente estado de la UCP (los registros que no aparecen no son usados por el programa que se está ejecutando):

AX, IP, RE, SP, CS, DS, SS (tomar los valores de la planilla provista por el profesor)

Alumno	AX	IP	RE	SP	CS	DS	SS	CALL	INT	Vec: IP	Vec: CS
--------	----	----	----	----	----	----	----	------	-----	------------	------------

### PUNTO 1:

Si la siguiente instrucción a ejecutar es CALL nnnn que ocupa 3 bytes, **indicar:**

**1.a** Mediante las componentes XXXX:YYYY de la memoria, en qué dirección de memoria está la instrucción CALL nnnn y cuál es la dirección que aparecerá en el bus de direcciones al buscar esa instrucción.

**1.b** Qué movimientos de información ocurren durante la ejecución de CALL nnnn y con qué valores quedan los registros involucrados.

**1.c** Cómo queda la pila luego de la ejecución de CALL nnnn.

### PUNTO 2:

Si la subrutina empieza con PUSH AX, **indicar:**

**2.a** Mediante las componentes XXXX:YYYY, donde se encuentra la instrucción PUSH AX y cual es la dirección que aparecerá en el bus de direcciones al buscarse la instrucción PUSH AX.

**2.b** Qué movimientos de información ocurren durante la ejecución de PUSH AX y con qué valores quedan los registros involucrados.

**2.c** Cómo queda la pila luego de la ejecución de PUSH AX

### PUNTO 3:

Si a PUSH AX (de un byte de longitud) sigue la instrucción INT nn (de dos bytes de longitud), cuyo vector de interrupciones contiene las componentes CS=CSIN e IP=IPIN, **indicar:**

**3.a** Mediante las componentes XXXX:YYYY, en que posición de memoria se encuentra INT66 y cómo aparecerá la misma en el bus de direcciones.

**3.b** Qué movimientos de información se realizan durante su ejecución para poder resguardar información en la Pila.

**3.c** Cómo queda la pila luego de los movimientos de información que dispara la ejecución de INT64.

**3.d** Cuáles son las acciones que faltan realizar para concluir con la ejecución de INT nn en relación con el flag I y la localización de la subrutina del SO que atiende INTnn. Describir la ubicación del vector de interrupción dentro de la Tabla de vectores.

**3.e** Con qué instrucción termina esta rutina que atiende la Interrupción por software, y qué movimientos de información genera su ejecución.

**3.f** Cómo queda la pila cuando finaliza la ejecución de esta subrutina del SO.

**3.g** Se puede enmascarar este tipo de interrupciones?

#### **PUNTO 4:**

Puesto que la subrutina empezó con PUSH AX e INTnn, **indicar:**

**4.a** Con qué instrucciones debe terminar.

**4.b** Qué movimientos ocurren cuando se ejecuta cada una, y cómo queda la pila luego de cada ejecución.

#### **PUNTO 5:**

Realizar un esquema de la memoria, graficando el proceso llevado a cabo entre los Puntos 1 y 4 de este examen, arrancando en la posición de memoria donde se encuentra el llamado a subrutina del Punto 1, y mostrando qué recorrido realiza el procesador buscando instrucciones en la memoria, hasta volver a ese mismo punto.

#### **PUNTO 6:**

Realizar un programa en lenguaje ensamblador, que tome como dato un vector de números NATURALES/ENTEROS de 8/16 bits cuya cantidad de elementos se proporciona en la planilla (elem). El vector dato inicia en (Orig). A cada elemento se le deben sumar (sumar), para formar un vector resultado que se guarda a partir de la dirección (Dest). En caso de que la operación de error se deberá interrumpir la operación y guardar el Código de error en la dirección de error.

Tomar los datos de este programa de la planilla provista por el profesor

Alumno	Vec:Orig	Vec:Dest	Elem	bits	TIPO	Sumar	Codigo Error	Dir Error
--------	----------	----------	------	------	------	-------	-----------------	-----------