

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

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
	AB0	7088	7404	1345	6000	6000	A700	1284	66	2843	8000

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 cuál 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:

Punto 1:

- a) Los componentes son CS: $IP = 6000:2843$, en el BUS aparecerá $6000H:2843 = 62843$
- b) $IP = IP + 3 = 2843 + 3 = 2846$ (IP apunta brevemente a la instrucción Int1 que sigue a CALL (que ocupa 3 bytes).

$SP \leftarrow SP - 2$ ($SP = 1345 - 2 = 1343$). Osea que la nueva cima de la pila está en 1343

$[SP] \leftarrow IP$ [$1343/9$] = 2846 EN LAS DIRECCIONES 1343/9 DE LA PILA SE GUARDÓ EL VALOR DE IP

$IP \leftarrow 1284$ $IP = 1284$ APUNTA A LA DIRECCIÓN 1284 DONDE COMIENZA LA SUBROUTINA.

- c) SS: SP = A700:1343 8B(IP)
:1344 70(IP)

Punto 2:

- a) CS: IP = 6000:1284. LA DIRECCIÓN EN BUS DE DIRECCIONES: $60000 + 1284 = 61284$

- b) $SP \leftarrow SP - 2$ ($SP = 1343 - 2 = 1341$) LA NUEVA CIMA DE LA PILA ESTÁ EN 1341
- $[SP] \leftarrow RE$ [$1341/7$] = 7404

EN LAS DIRECCIONES 1341/7 DE LA PILA SE GUARDÓ EL VALOR DE RE QUE CONTIENE LOS VALORES DE LOS FLAGS.

- c) SS: SP = A700:1341 04(RE)
:1342 74(RE)
:1343 8B(IP)
:1344 70(IP)

$SP \leftarrow SP - 2$ ($SP = 1341 - 2 = 133F$) LA NUEVA CIMA DE LA PILA ESTÁ EN 133F

$[SP] \leftarrow AX$ [$133F/5$] = ABCD EN LAS DIRECCIONES 133F/5 DE LA PILA SE GUARDÓ EL VALOR DE AX

- SS: SP = A700:133F CD(AX)
:1340 AB(AX)
:1341 04(RE)
:1342 74(RE)
:1343 8B(IP)
:1344 70(IP)

PUNTO 3:

a) ~~TODAS LAS INSTRUCCIONES PUSH E INTXX OCUPAN 2 bytes en memoria. Nuestra pila PUSH AX comienza en A700:~~

Puesto que PUSH AX comienza en A700:1284, INT66 comenzará en A700:1286.

b) $SP \leftarrow SP - 2$ ($SP = 133F - 2 = 133D$) la nueva cima de la pila está en 133D

$[SP] \leftarrow RE$ $[133D/2] = 0474$ (como hasta ahora la UAL no hizo ninguna operación, el valor de RE se mantiene).

$SP \leftarrow SP - 2$ ($SP = 133D - 2 = 133B$) la nueva cima de la pila está en 133B

$[SP] \leftarrow CS$ $[133B/0] = 6000$.

$IP \leftarrow IP + 2$ ($IP = 1286 + 2 = 1288$)

$SP \leftarrow SP - 2$ ($SP = 133B - 2 = 1339$) la nueva cima de la pila está en 1339.

$[SP] \leftarrow IP$ $[1339/F] = 1288$

c) la pila quedará:

SS:SP = A700:	1339	88 (IP)
	133A	12 (IP)
	133B	00 (CS)
	133C	60 (CS)
	133D	04 (RE)
	133E	74 (RE)
	133F	CD (AX)
	1340	AB (AX)
	1341	04 (RE)
	1342	74 (RE)
	1343	8B (IP)
	1344	70 (IP)

d) A los efectos de localizar la subrutina que atiende a INT66, la componente izquierda de la dirección de su vector es 0000; y la componente derecha vale $21 \times 4 = 84h$. O sea que la dirección del vector será 0000:0084. En la zona de vectores se tendrá:

0000:	0084	43 IN (IP)
0000:	0085	2B IP (IP)
0000:	0086	00 IN (CS)
0000:	0087	80 CS (CS)

Entonces $CS:IP = 0000:2843$ ES LA DIRECCIÓN DONDE EMPIEZA LA SUBROUTINA DEL SO QUE ATIENDE A INT66.

→ e) LA SUBROUTINA DEL SO DEBE TERMINAR CON IRET. SU EJECUCIÓN COMPRENDE LOS SIGUIENTES MOVIMIENTOS:

$IP \leftarrow [ESP]$ $IP = 128B$

$SP \leftarrow SP + 2$ ($SP = 1339 + 2 = 133B$) LA NUEVA CIMA DE LA PILA ESTÁ EN 133B

$CS \leftarrow [ESP]$ $CS = 6000$

$SP \leftarrow SP + 2$ ($SP = 133B + 2 = 133D$) LA NUEVA CIMA DE LA PILA ESTÁ EN 133D

$RE \leftarrow [ESP]$ $CS = 7404$

$SP \leftarrow SP + 2$ ($SP = 133D + 2 = 133F$; ASÍ SP APUNTA AL CONTENIDO DE AX GUARDADO EN LA PILA).

f) LA PILA QUEDA COMO EN EL PUNTO (2-C)

g) INT66 COMO CUALQUIER OTRA INTERRUPTIÓN POR SOFTWARE NO PUEDE SER ENMASCARADA, O SEA DEMORADA, PUES SE TRATA DE UNA INSTRUCCIÓN QUE EJECUTA LA UC, Y NO HAY FORMA DE DEMORARLA.

PUNTO 4

a) LAS INSTRUCCIONES DEBEN SER:

- POP AX
- RET.

b) EJECUCIÓN POP AX

$AX \leftarrow [ESP]$ ($AX = ABCD$)

$SP \leftarrow SP + 2$ ($SP = 133F + 2 = 1341$)

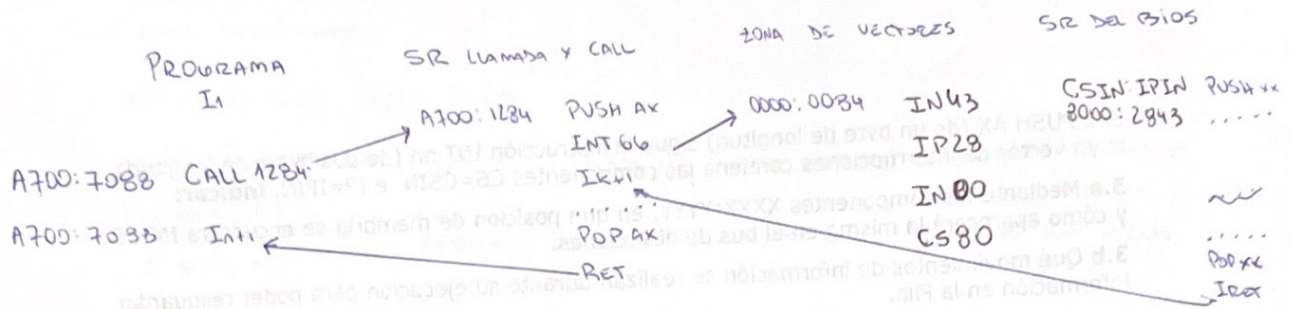
LA NUEVA CIMA ESTÁ EN 1341 Y LA PILA QUEDA COMO EN EL PUNTO (2C)

EJECUCIÓN RET

$IP \leftarrow [ESP]$ ($IP = 2407$)

$SP \leftarrow SP + 2$ ($SP = 1341 + 2 = 1343$) LA NUEVA CIMA ESTÁ EN 1343 DESDE DONDE COMENZÓ ESTE EJERCICIO

PUNTO 5:



PUNTO 4:

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

4.2. Con qué instrucciones debe terminar.

4.3. Puesto que la subrutina empieza con PUSH AX e INTIM, indicar:

PUNTO 5:

Realizar un resumen de la memoria, graficando el proceso llevado a cabo entre los puntos 1 y 4 de este examen, anotando en la posición de memoria donde se encuentra el llamado a subrutina del Punto 1, y mostrando qué registro 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 cuyos contenidos se proporcionan en la planilla (leem). El vector debe iniciar en (0). A cada elemento se le debe sumar (sumar), para formar un vector resultado que se guarda a partir de la dirección (dest). En caso de que la dirección de error se deba 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	Vector	VecDest	Elem	bits	Tipo	Suma	Error	Dir Error