

FACULTAD:	Tecnología Informática		
CARRERA:	ANALISTA PROGRAMADOR		
ALUMNO/A:	GERARDO TORDOYA		
SEDE:	ULTRA	LOCALIZACIÓN:	Buenos Aires
ASIGNATURA:	Sistemas de Computación II		
COMISIÓN:	1-O-N	TURNO:	NOCHE
PROFESOR:	RAUL ROMERO	FECHA:	18/07/2021
TIEMPO DE RESOLUCIÓN:	90 min	EXAMEN PARCIAL N°:	2 (dos)
MODALIDAD DE RESOLUCIÓN:		A Distancia Asincrónico - Escrito	
CALIFICACIÓN:			
RESULTADOS DE APRENDIZAJE ESPERADO: Asegurar la comprensión de la Segmentación de Memoria y la Administración de la Pila			

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
15	CDEF	5600	7676	1A34	A000	A000	B400	4425	60	4821	7000

A A000:5600

A000:5600 CALL 4425

E B400:1A32

A A000:4425

A000:4425 PUSH AX

A000:4426 INT 60

A000:4428 POP AX

A000:4429 RET

A000:442A ----

PUNTO 1:

Si la siguiente instrucción a ejecutar es CALL 4425 que ocupa 3 bytes, **indicar:**

1.a Mediante las componentes XXXX:YYYY de la memoria, en qué dirección de memoria está la instrucción CALL 4425 y cuál es la dirección que aparecerá en el bus de direcciones al buscar esa instrucción.

CALL 4425 se encuentra en A000:5600

En el bus de direcciones aparecerá:	A0000	
	+ 5600	

	A5600	(1010 0101 0110 0000 0000)

1.b Qué movimientos de información ocurren durante la ejecución de CALL 4425 y con qué valores quedan los registros involucrados.

Los movimientos que realiza CALL 4425 son:

- $SP \leftarrow SP - 2$ { $SP=1A34-2=1A32$ }
- $[SP] \leftarrow IP$ {[$1A32$]= $5600+3=5603$ }
- $IP \leftarrow (IP \leftarrow IP + DA)$ { $IP=4425$ }

SP=1A32

IP=4425

1.c Cómo queda la pila luego de la ejecución de CALL 4425.

B400:1A32	03	← CIMA DE LA PILA
B400:1A33	56	
B400:1A34	XX	

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-RSS
SS 0792
:B400
-A A000:5600
A000:5600 CALL 4425
A000:5603
-R
AX=CDEF BX=0000 CX=0000 DX=0000 SP=1A34 BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=5600 NU UP EI PL NZ NA PO NC
A000:5600 EB22EE CALL 4425
-T
AX=CDEF BX=0000 CX=0000 DX=0000 SP=1A32 BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=4425 NU UP EI PL NZ NA PO NC
A000:4425 0000 ADD [BX+SI],AL DS:0000=00
-E
-E B400:1A32
B400:1A32 03. 56.

```

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.

CS:IP

PUSH AX se encuentra en A000:4425

En el bus de direcciones aparecerá:

A0000	
+ 4425	

A4425	(1010 0100 0100 0010 0101)

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

PUSH AX

- $SP \leftarrow SP - 2$ $\{SP = 1A32 - 2 = 1A30\}$
- $[SP] \leftarrow AX$ $\{CDEF\} \leftarrow AX$ NO CAMBIA

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

[SS:SP]

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
A000:4429 RET
A000:442A
-
-
-R
AX=CDEF BX=0000 CX=0000 DX=0000 SP=1A32 BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=4425  NU UP EI PL NZ NA PO NC
A000:4425 50          PUSH    AX
-T
AX=CDEF BX=0000 CX=0000 DX=0000 SP=1A30 BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=4426  NU UP EI PL NZ NA PO NC
A000:4426 CD60          INT     60
-
-E B400:1A30
B400:1A30 EF.   CD.   03.   56._

```

```

B400:1A30      EF   ← CIMA DE LA PILA
B400:1A31      CD
B400:1A32      03
B400:1A33      56
B400:1A34      XX

```

PUNTO 3:

Puesto que la subrutina empezó con PUSH AX e INT **60**, **indicar:**

3.a Con qué instrucciones debe terminar.

POP AX

RET

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

NOTA 1:

SEGÚN EL EJERCICIO DEL LIBRO, ANTERIORMENTE HUBO OPERACIONES QUE AFECTARON LOS VALORES PEDIDOS AHORA (PUNTO 3 DEL LIBRO). ESOS VALORES, DESPUÉS DE LA INSTRUCCIÓN INT 60, SE LISTAN A CONTINUACIÓN EN FORMA RESUMIDA:

SP=1A30-2=1A2E

SP=1A2E-2=1A2C

SP=1A2C-2=1A2A

A PARTIR DE AQUÍ, CONTINÚA LA RESPUESTA DE LA PRESENTE CONSIGNA.

POP AX

- AX \leftarrow [SP] {SP=1A2A}
- SP \leftarrow SP+2 {SP=1A2A+2=1A2C}

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
:4426
-R
AX=CDEF BX=0000 CX=0000 DX=0000 SP=1A2A BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=4426 NU UP DI PL NZ NA PO NC
A000:4426 CD60 INT 60
-RIP
IP 4426
:4428
-R
AX=CDEF BX=0000 CX=0000 DX=0000 SP=1A2A BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=4428 NU UP DI PL NZ NA PO NC
A000:4428 5B POP AX
-T
AX=4428 BX=0000 CX=0000 DX=0000 SP=1A2C BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=4429 NU UP DI PL NZ NA PO NC
A000:4429 C3 RET
-T
AX=4428 BX=0000 CX=0000 DX=0000 SP=1A2E BP=0000 SI=0000 DI=0000
DS=A000 ES=0792 SS=B400 CS=A000 IP=A000 NU UP DI PL NZ NA PO NC
A000:A000 0000 ADD [BX+SI],AL DS:0000=00
-E B400:1A2C
B400:1A2C CC. 01. 02. 70.
B400:1A30 EF. CD. 03. 56. _
```

TENIENDO EN CUENTA QUE NO SE ESTÁ ESCRIBIENDO UN PROGRAMA SINO ANALIZANDO LO QUE SUCEDE CON CADA INSTRUCCIÓN, SI NOS LIMITAMOS A CALCULAR TEÓRICAMENTE LOS VALORES (YA SIN RECURRIR AL DEBUG), TENDRÍAMOS QUE RETOMAR LOS VALORES ASENTADOS EN EL PUNTO 2.B, Y ESO DEJARÍA:

SP \leftarrow 1A30

[SP] \leftarrow CDEF

SI APLICAMOS LO PEDIDO EN ESTA CONSIGNA RETOMANDO DESDE LOS VALORES EN 2.B:

POP AX

- AX \leftarrow [SP] {SP=1A30}
- SP \leftarrow SP+2 {SP=1A30+2=1A32}

SEGÚN EL LIBRO (P84), LA PILA QUEDA COMO EN EL PASO 2.C

B400:1A30 EF
B400:1A31 CD
B400:1A32 03
B400:1A33 56
B400:1A34 XX

RET

- $IP \leftarrow [SP] \quad \{1A2C\}$
- $SP \leftarrow SP+2 \quad \{1A2C+2=1A2E\}$

SI APLICAMOS LO PEDIDO EN ESTA CONSIGNA RETOMANDO DESDE LOS VALORES EN 2.B Y TRABAJADOS EN EL PUNTO ANTERIOR:

$IP \leftarrow 1A30$
 $SP \leftarrow 1A32$

SEGÚN EL LIBRO (P84), LA NUEVA CIMA DE LA PILA QUEDA COMO AL INICIO DEL EJERCICIO.

NOTA 2:

EN LA CAPTURA ANTERIOR, SI SE OBSERVAN LOS VALORES LUEGO DE RET, SE OBSERVARÁN QUE SE PRODUCE UN RESETEO, EL CUAL (SEGÚN LA DOCUMENTACIÓN QUE PUDE ENCONTRAR) SE EXPLICA PORQUE, POR ESTAR "CERCENADO" EL PROGRAMA (NOTA 1 SOBRE LA AUSENCIA DE PARTE DEL EJERCICIO) EL RETORNO SE HACE HACIA LA NADA. EN ESOS CASOS, EL DEBUG PUEDE ACTUAR COMO PARECE HABER HECHO: "RESETEANDO" VALORES O PARÁMETROS COMO EL SEGMENTO.

PUNTO 4:

Realizar un esquema de la memoria, graficando el proceso llevado a cabo entre los Puntos 1 y 3 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.

