|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FACULTAD: | **Tecnología Informática** | | | | |
| CARRERA: | **Analista Programador** | | | | |
| ALUMNO/A: | **Gastón Alejandro Galeano** | | | | |
| SEDE: | **Online** | | LOCALIZACIÓN: | **Buenos Aires** | |
| ASIGNATURA: | **Sistemas de Computación II** | | | | |
| COMISIÓN: | **L** | | TURNO: | **N** | |
| PROFESOR: | **Enrique Emilio Douce** | | FECHA: | **17/07/2021** | |
| 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** |
| 18 | 1200 | 8900 | 7778 | 4A56 | 3000 | 3000 | B700 | 4451 | 63 | 6666 | 2000 |

**PUNTO 1:**

Si la siguiente instrucción a ejecutar es CALL **4451** que ocupa 3 bytes, **indicar:**

**1.a** Mediante las componentes XXXX:YYYY de la memoria, en qué dirección de memoria está la instrucción CALL 4451 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 4451 y con qué valores quedan los registros involucrados.

**1.c** Cómo queda la pila luego de la ejecución de CALL 4451.

**1.d** Represente usando DEBUG esta ejecución mostrando con el comando R y T como se produce el salto y como cambia SP, y con E como ha cambiado la pila.

**RESPUESTA**:

**1.a** El CALL 4451, se encuentra en 3000:8900

En el bus de operaciones aparecerá:

|  |  |
| --- | --- |
| + | 30000 |
| 8900 |
| **38900** | |

(0011 1000 1001 0000 0000)

**1.b** Los movimientos que realiza CALL 4451 son:

CALL XXXX

* SP SP-2
* [SP]  IP
* IP  DA + IP

Con mis datos, quedaría de la siguiente manera:

CALL **4451**

• SP = ***4A54*** 🡨 SP = ***4A56* -**2

• [SP] = ***[4A54]*** 🡨 (IP = ***8900 +3****) =* ***8903***

• IP 🡨 **4451**

Los datos SP y IP quedarían de la siguiente manera:

SP = **4A54**

IP = **4451**

**1.c** La pila luego de la ejecución CALL 4451 queda de la siguiente manera, tomando el SS:SP:

***Cima de la pila*→** B700:4A54 **03**

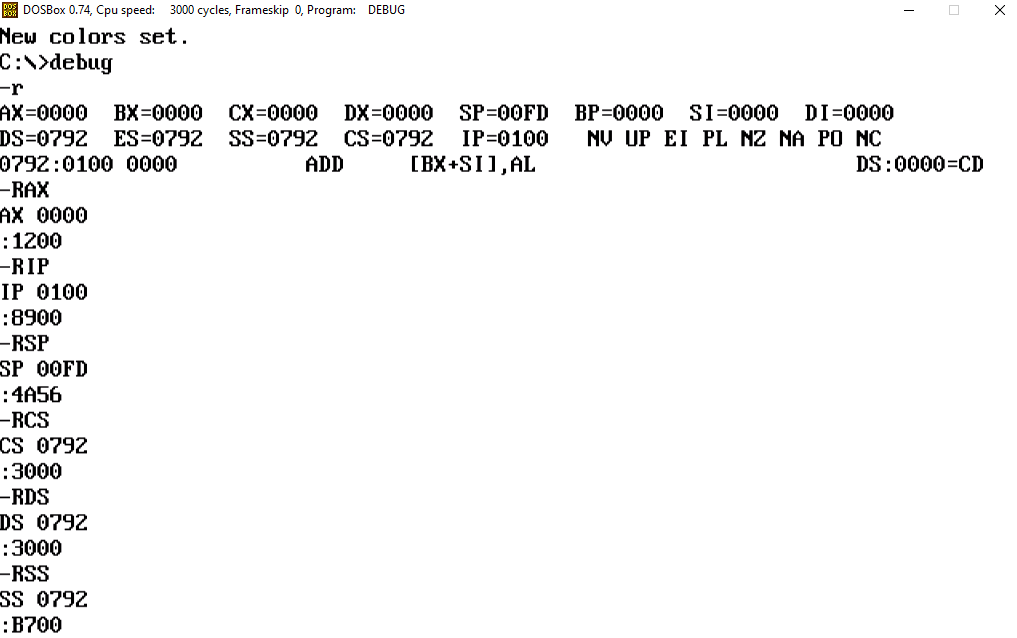
B700:4A55 **89**

B700:4A56 **XX**

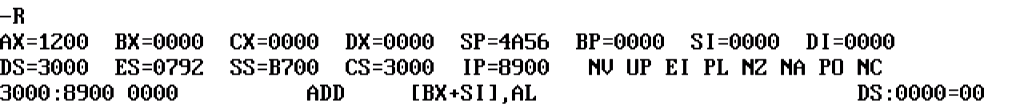
Siendo el **8903** el valor de la IP.

**1.d** A continuación se representa la ejecución correspondiente por DEBUG:

* **1er imagen**: Inicializo todos los datos, con excepción del RE que no es posible manipularlo, según los datos proporcionados en el documento **DatosAsignados**.



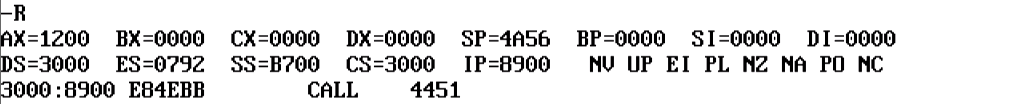
* **2da Imagen**: Chequeo todos los datos que se cargaron previamente:



* **3er Imagen**: Modifico con el comando A el segmento CS:IP y cargo CALL 4451 la cual ocupa 3 posiciones.

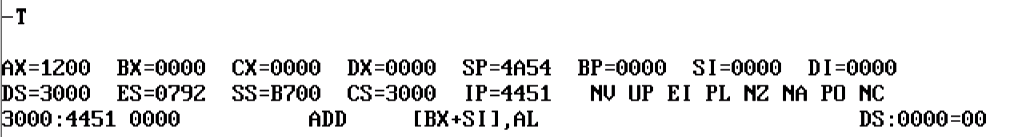
****

* **4ta Imagen**: Controlos los valores a modificar con el comando R.





* **5ta Imagen**: Ejecuto con comando T y valido que los datos SP = 4A54 e IP = 4451 correspondientes al punto 1.b son los correctos.





* **6ta Imagen**: Utilizo el comando E para controlar los valores de la pila correspondientes al punto 1.c



**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

**RESPUESTA:**

**2.a** La instrucción PUSH AX se encuentra en 3000:4451

|  |  |
| --- | --- |
| + | 30000 |
| 4451 |
| **34451** | |

(0011 0100 0100 0101 0001)

**2.b** Los movimientos que realiza durante el PUSH AX son los siguientes:

PUSH Ax

SP ßSP-2

[SP] ß Ax (o cualquier otro registro)

Con mis datos, quedaría de la siguiente manera:

PUSH **1200**

SP = **4A52** 🡨 SP= **4A54** – 2

[SP] = **[4A52]** 🡨 AX = **1200**

Los datos SP y AX quedarían de la siguiente manera:

SP=**4A52**

AX=**1200**

**2.c** La pila luego de la ejecución queda de la siguiente manera:

***Cima de la pila*→** B700:4A52 **00**

B700:4A53 **12**

B700:4A54 **03**

B700:4A55 **89**

B700:4A56 **XX**

Siendo el **8903** el valor de la IP y el **1200** el valor de la AX.

**PUNTO 3:**

Si a PUSH AX (de un byte de longitud) sigue la instrucción INT **63** (de dos bytes de longitud), cuyo vector de interrupciones contiene las componentes CS=**2000** e IP=**6666**, **indicar:**

**3.a** Mediante las componentes XXXX:YYYY, en que posición de memoria se encuentra INT **63** 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 INTnn.

**3.d** Cuáles son las acciones que faltan realizar para concluir con la ejecución de INT 63 en relación con el flag I y la localización de la subrutina del SO que atiende INT 63. 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?

**3.h** Represente usando DEBUG esta ejecución mostrando con el comando R y T como se produce el salto y como cambia SP, y con E como ha cambiado la pila y el contenido del vector de interrupción

**RESPUESTA:**

**3.a** INT 63 se encuentra en la posición de memoria en 3000:4452

|  |  |
| --- | --- |
| + | 30000 |
| 4452 |
| **34452** | |

(0011 0100 0100 0101 0010)

**3.b** Los movimientos que realiza durante el INT 63 son los siguientes:

INT 63

* SP SP-2
* [SP]  RE
* SP SP-2
* [SP]  CS
* SP SP-2
* [SP]  IP

Con mis datos, quedaría de la siguiente manera:

INT 63

SP= **4A50** 🡨 SP= **4A52** – 2

[SP] = [**4A50**] 🡨 RE = **7778**

SP = **4A4E** 🡨 SP = **4A50** – 2

[SP] = [**4A4E**] 🡨 CS = **3000**

SP = **4A4C** 🡨 SP = **4A4E** – 2

[SP] = [**4A4C**] 🡨 IP = **4454**

Los datos SP y IP quedarían de la siguiente manera:

SP= **4A4C**

IP= **4454**

**3.c** La nueva configuración de la pila será:

***Cima de la pila*→**B700:4A4C **54**

B700:4A4D **44**

B700:4A4E **00**

B700:4A4F **30**

B700:4A50 **78**

B700:4A51 **77**

B700:4A52 **00**

B700:4A53 **12**

B700:4A54 **03**

B700:4A55 **89**

B700:4A56 **XX**

Siendo el **8903** el valor de la IP, el **1200** el valor de la AX, el **7778** el valor del RE, el **3000** el valor de la CS y el **4454** el valor de la IP.

**3.d** Los movimientos que realizan son:

INT 63

* IP  [63\*4]
* CS  [63\*4+2]
* DI  Flag I = 0

IP = 6666

CS = 2000

Vector de Interrupción:

0000:018C 66

0000:018D 66

0000:018E 00

0000:018F 20

**3.e** La rutina que atiende INT 63 termina con IRET:

IRET

* IP  [SP]
* SP  SP+2
* CS  [SP]
* SP  SP+2
* RE  [SP]
* SP  SP+2
* EI  Flag I

Con mis datos, quedaría de la siguiente manera:

IRET

IP= **4A4C** 🡨 [SP] = [**4A4C**]

SP= **4A4E** 🡨 SP = **4A4C** + 2

CS= **4A4E** 🡨 [SP] = [**4A4E**]

SP= **4A50** 🡨 SP = **4A4E** + 2

RE= **4A50** 🡨 [SP] = [**4A50**]

SP= **4A52** 🡨 SP = **4A50**+ 2

EI  Flag I = 1

**3.f** La pila queda con la situación previa a ejecutar INT 63.

***Cima de la pila*→** B700:4A52 **00**

B700:4A53 **12**

B700:4A54 **03**

B700:4A55 **89**

B700:4A56 **XX**

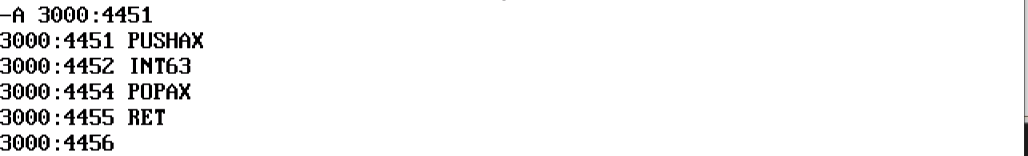
**3.g** Si, se puede enmascarar este tipo de interrupciones por medio del Flag I.

**3.h** PREGUNTAR

1er imagen: Utilizo el código e, para cargar valor de vectores



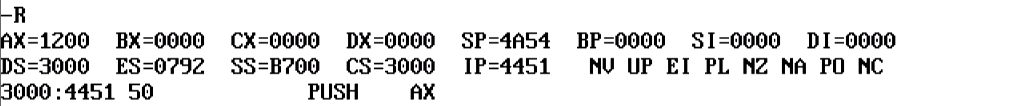
2er imagen: Utilizo el código A, para cargar instrucciones en SS:IP



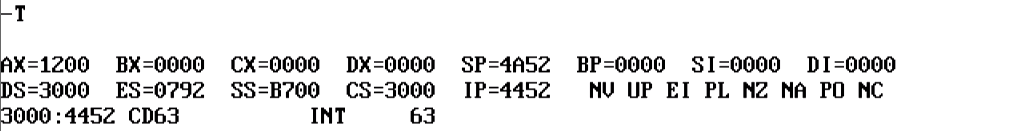
3er imagen:



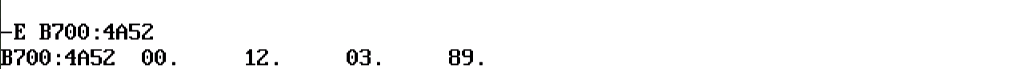
4ta imagen:



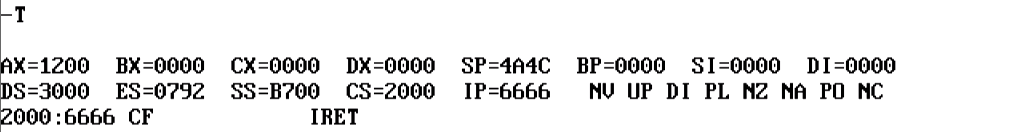
5ta imagen:



6ta imagen:



7ma imagen:



**PUNTO 4:**

Puesto que la subrutina empezó con PUSH AX e INT 63, **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.

**RESPUESTA:**

**4.a** La subrutina debe terminar con POP AX y luego RET.

**4.b Los movimientos que debe ejecutar son los siguientes:**

POP Ax

* Ax  [SP]
* SP  SP+2

RET

* IP  [SP]
* SP  SP+2

Con mis datos, quedaría de la siguiente manera:

POP 1200

AX = **4A52** 🡨 [SP] = [**4A52**]

SP = **4A54** 🡨 SP = **4A52** + 2

RET

IP= **4A54** 🡨 [**4A54**]

SP = **4A56**  🡨 SP = **4A54 + 2**