## Ejercitación manual

El Trabajo Práctico es personalizado, razón por la cual las direcciones y datos con los que usted trabajará serán las ocho cifras de su Documento Nacional de Identidad (DNI). En caso de números particulares, no dude en consultar con su tutor/a. Se presentarán las consignas indicando 8 cifras de forma genérica (12.345.678) las que luego, para realizar la actividad, usted deberá sustituir por las personales.

En un lenguaje de alto nivel se ha tipeado lo siguiente:

MAGNITUDES ↵

R = P + Q – T ↵

**P** = 123D **Q** = 456 D **T** = 78 D

**PRECISIONES SOBRE LOS DATOS SOBRE LOS QUE SE TRABAJARÁN**

MAGNITUDES ↵

R = P + Q – T ↵

**P** = 227D (00E316) **Q** = 774D (030616) **T** = 20D (001416)

**R**=981 (03D516)

* + 1. Complete la tabla que comienza en 4D escribiendo en HEXADECIMAL cómo quedan en memoria codificados en ASCII los caracteres arriba tipiados (se indica en 0100 el código ASCII 4D de M). Para codificar utilice la tabla ASCII. (Se elige necesariamente la primera dirección 0100 por razones de compatibilidad con el Debug, que usaremos en la segunda parte de este ejercicio). Complete también las direcciones. Puede tomar como referencia, del texto La PC por dentro, Ginzburg: 2006, lo desarrollado en la página 1‐149, pero debe completar las celdas de memoria en hexa y no en binario.

|  |  |  |  |
| --- | --- | --- | --- |
| **MEMORIA** | **PROGRAMA** | **ASCII** | **HEX** |
| 0100 | M | 77 | 4D |
| 0101 | A | 65 | 41 |
| 0102 | G | 71 | 47 |
| 0103 | N | 78 | 4E |
| 0104 | I | 73 | 49 |
| 0105 | T | 84 | 54 |
| 0106 | U | 85 | 55 |
| 0107 | D | 68 | 44 |
| 0108 | E | 69 | 45 |
| 0109 | S | 83 | 53 |
| 0110 |  | 13 | D |
| 0111 | R | 82 | 52 |
| 0112 | = | 61 | 3D |
| 0113 | P | 80 | 50 |
| 0114 | + | 43 | 2B |
| 0115 | Q | 81 | 51 |
| 0116 | - | 45 | 2D |
| 0117 | T | 84 | 54 |
| 0118 |  | 13 | D |
| 0119 | P | 80 | 50 |
| 0120 | = | 61 | 3D |
| 0121 | 2 | 50 | 32 |
| 0122 | 2 | 50 | 32 |
| 0123 | 7 | 55 | 37 |
| 0124 |  | 32 | 20 |
| 0125 | Q | 81 | 51 |
| 0126 | = | 61 | 3D |
| 0127 | 7 | 55 | 37 |
| 0128 | 7 | 55 | 37 |
| 0129 | 4 | 52 | 34 |
| 0130 |  | 32 | 20 |
| 0131 | T | 84 | 54 |
| 0132 | = | 61 | 3D |
| 0133 | 2 | 50 | 32 |
| 0134 | 0 | 48 | 30 |
| 0135 |  | 13 | D |
| 0136 |  | 10 | A |

* + 1. Indique cómo deja el compilador en memoria la zona de datos e instrucciones. Para ello:

**2a)** Reserve para cada variable **dos** posiciones de memoria, a partir de la dirección dada por las primeras 4 cifras de su DNI (suponiendo que es un número en hexa, aunque en realidad sea decimal) indicando luego las direcciones siguientes. Se ejemplifica para el DNI 28482422. (No utilice estas direcciones para realizar el trabajo práctico sino las 4 primeras cifras que surjan de su DNI, considerándolas como un número hexadecimal). Los valores P, Q y T se determinarán en el apartado 2b).

|  |  |
| --- | --- |
| **MEM** | **VAR[[1]](#footnote-1)** |
| 2277 | R |
| 2278 | R |
| 2279 | P |
| 2280 | P |
| 2281 | Q |
| 2282 | Q |
| 2283 | T |
| 2284 | T |

**2b**) Pasar a binario natural los valores de **P**, **Q** y **T** tomados de su DNI y considerados como números en base diez (utilizando el método de las pesas). A cada número binario así obtenido agregarle ceros a la izquierda hasta completar el formato de 16 bits correspondiente a dos posiciones consecutivas de memoria. Luego, pasarlo a hexa, de forma que resulte un número de 4 símbolos. Completar las siguientes líneas. Recuerde realizar el ejercicio con los números de su DNI.

|  |  |  |
| --- | --- | --- |
| **DECIMAL A BINARIO** | | |
| **PESA** | **VALOR** | **BINARIO** |
| 2^7 | 128 | 1 |
| 2^6 | 64 | 1 |
| 2^5 | 32 | 1 |
| 2^4 | 16 | 0 |
| 2^3 | 8 | 0 |
| 2^2 | 4 | 0 |
| 2^1 | 2 | 1 |
| 2^0 | 1 | 1 |
| **DECIMAL:** | | **227** |
| **BINARIO:** | | **0000 0000 1110 0011** |
| **HEXADECIMAL:** | | **00E3** |

|  |  |  |
| --- | --- | --- |
| **DECIMAL A BINARIO** | | |
| **PESA** | **VALOR** | **BINARIO** |
| 2^9 | 512 | 1 |
| 2^8 | 256 | 1 |
| 2^7 | 128 | 0 |
| 2^6 | 64 | 0 |
| 2^5 | 32 | 0 |
| 2^4 | 16 | 0 |
| 2^3 | 8 | 0 |
| 2^2 | 4 | 1 |
| 2^1 | 2 | 1 |
| 2^0 | 1 | 0 |
| **DECIMAL:** | | **774** |
| **BINARIO:** | | **0000 0011 0000 0110** |
| **HEXADECIMAL:** | | **0306** |

|  |  |  |
| --- | --- | --- |
| **DECIMAL A BINARIO** | | |
| **PESA** | **VALOR** | **BINARIO** |
| 2^4 | 16 | 1 |
| 2^3 | 8 | 0 |
| 2^2 | 4 | 1 |
| 2^1 | 2 | 0 |
| 2^0 | 1 | 0 |
| **DECIMAL:** | | **20** |
| **BINARIO:** | | **0000 0000 0001 0100** |
| **HEXADECIMAL:** | | **0014** |

**DNI: 22.777.420[[2]](#footnote-2)**

P=227 BIN: 0000 0000 1110 0011 HEX: 00E3

Q=774 BIN: 0000 0011 0000 0110 HEX: 0306

T=20 BIN: 0000 0000 0001 0100 HEX: 0014

**2c**) Escribir los valores en hexadecimal hallados en el punto 2b para P, Q y T en las posiciones de memoria reservadas en el punto 2a, teniendo presente que para Intel se necesita escribir los dos símbolos menos significativos en la dirección más baja y los dos restantes en la dirección siguiente. Por ejemplo, si P = 032A, escribirá primero 2A y abajo 03, como se ejemplifica en la página 149 antes citada.

|  |  |
| --- | --- |
| **MEM** | **VAR[[3]](#footnote-3)** |
| 2277 |  |
| 2278 |  |
| 2279 | E3 |
| 2280 | 00 |
| 2281 | 06 |
| 2282 | 03 |
| 2283 | 14 |
| 2284 | 00 |

**2d**) En función de las **direcciones** determinadas en el punto 2a, y a partir de la dirección dada por las últimas 4 cifras del DNI, **completar en XX que aparecen las direcciones de P, Q y T en la zona de instrucciones de la tabla siguiente**, de modo que la sentencia R = P + Q – T quede traducida como lo haría un supuesto compilador. Se indican los códigos de operación de dichas instrucciones correspondientes a Intel.

La dirección de la primera instrucción (de código de operación A1 y dirección XX de la variable P) siguiendo con el supuesto DNI = 28482422, es 2422. Usted las debe sustituir por las últimas 4 cifras de su DNI, considerándolas como un número hexadecimal.

|  |  |
| --- | --- |
| **MEM** | **INS[[4]](#footnote-4)** |
| 7420 | A1 |
| 7421 | 79 |
| 7422 | 22 |
| 7423 | 03 |
| 7424 | 06 |
| 7425 | 81 |
| 7426 | 22 |
| 7427 | 2B |
| 7428 | 06 |
| 7429 | 83 |
| 7430 | 22 |
| 7431 | A3 |
| 7432 | 77 |
| 7433 | 22 |

* + 1. Suponiendo que se ejecutan las instrucciones codificadas en el punto 2d, indicar en hexa cómo quedaría el registro AX, luego de cada instrucción. En caso de que una instrucción ordene una suma o resta, efectuarla en código binario, como lo realiza la UAL, en 16 bits y verificar que el resultado sea el esperado. Esto significa, pasar el resultado binario a decimal y comprobar que el resultado de la cuenta efectuada en decimal coincida con el resultado binario pasado a decimal.

|  |  |
| --- | --- |
| **INS** | **VAL** |
| P | 0000 0000 1110 0011 |
| + |  |
| Q | 0000 0011 0000 0110 |
| AX | 0000 0011 1110 1001 |
| - |  |
| T | 0000 0000 0001 0100 |
| AX | 0000 0011 1110 1001 |
| T-CONV | 10 1011 |
| UNI | 1 |
| AX | 0000 0011 1101 0101 |
| AX-HEX | 03D5[[5]](#footnote-5) |

Luego de realizar la operación, pasar el resultado de P+Q de binario a decimal y verificar que es el valor esperado.

|  |  |
| --- | --- |
| P+Q | 0000 0011 1110 1001 |

Comparando los valores de P+Q de la tabla anterior y la siguiente, los resultados coinciden.

|  |  |  |
| --- | --- | --- |
| **BINARIO A DECIMAL** | | |
| **PESA** | **VALOR** | **DECIMAL** |
| 0010 0000 0000 | 2^9 | 512 |
| 0001 0000 0000 | 2^8 | 256 |
| 0000 1000 0000 | 2^7 | 128 |
| 0000 0100 0000 | 2^6 | 64 |
| 0000 0010 0000 | 2^5 | 32 |
| 0000 0001 0000 | 2^4 |  |
| 0000 0000 1000 | 2^3 | 8 |
| 0000 0000 0100 | 2^2 |  |
| 0000 0000 0010 | 2^1 |  |
| 0000 0000 0001 | 2^0 | 1 |
| **BINARIO:** | | **0000 0011 1101 0101** |
| **DECIMAL** | | **1001** |

* + 1. Indicar en hexa como queda en memoria (en dos posiciones sucesivas) el valor de la variable **R**.

|  |  |
| --- | --- |
| **MEM** | **VAL** |
| 2277 | D5 |
| 2278 | 03 |

* + 1. Indicar cómo debe quedar en memoria en **código ASCII** la información que debe llegar a la impresora para que imprima el valor de **R** en base diez (R = XXX). Complete la siguiente lista, escribiendo al lado de cada símbolo y dígito decimal su codificación en ASCII (en hexadecimal):

|  |  |  |
| --- | --- | --- |
| **CHAR** | **HEX** | **DEC** |
| R | 52 | 82 |
| = | 3D | 61 |
| 9 | 39 | 57 |
| 8 | 38 | 56 |
| 1 | 31 | 49 |

## Ejecución experimental

Para la resolución de esta tercera instancia usted deberá utilizar el programa Debug y el teclado. Cada vez que complete una pantalla usando el Debug deberá guardarla, caso contrario perderá la información que se encuentra en la parte superior de la pantalla al ingresar nueva información. Usted deberá adjuntar es tos archivos al trabajo práctico y enviarlos a su profesor tutor.

Una forma de realizar la impresión es achicar (mediante Alt ↵) la ventana del Debug. Luego cliquear en el borde azul de la misma con el botón derecho del mouse para que aparezca una ventanita con la opción **edit** y desde ésta a la opción **mark** de modo que aparezca en el borde superior izquierdo de la ventana del debug un pequeño rectángulo blanco. Este se puede agrandar con el botón izquierdo del mouse, hasta seleccionar la zona que se quiere enviar al portapapeles, para lo cual se debe pulsar Enter. Después se debe abrir el Word y, en un archivo nuevo, se debe hacer Ctrl V, para que la zona antes seleccionada aparezca en pantalla, de modo de poder guardarla y luego imprimirla desde el Word.

Otra forma de hacerlo podría ser la siguiente: cada vez que usted complete una pantalla del Debug deberá capturarla (con la tecla *print screen* o imprimir pantalla) y copiarla en un archivo que deberá guardar y entregar luego como adjunto con el Trabajo Práctico.

* + 1. Verificación de la coincidencia entre los códigos ASCII hallados en el punto 1, y los tipiados mediante el teclado.

**6a**) Entrar al DOS y tipiar lo que se encuentra en letra itálica:

C > *COPY CON MARIO. TXT* ↵ (o cualquier nombre seguido por un punto y tres letras).

*MAGNITUDES* ↵

*R = P + Q – T* ↵

*P = 123 Q = 456 T = 78* ↵

Ctrl Z

1 ARCHIVO COPIADO (responde el DOS)

Graphical user interface, text

Description automatically generated

**6b**) En el DOS tipiar lo que se encuentra en letra itálica. C > *DEBUG MARIO. TXT* ↵

**6c**) Al lado del guión que aparecerá luego de tipiar el comando anterior, con lo cual debe aparecer el código 4D de la M en ASCII, de la palabra MAGNITUDES:

*‐E 0100* ↵ 4D

Luego, pulsar sucesivamente la barra espaciadora, de modo que vayan apareciendo los códigos ASCII en hexa de los caracteres tipeados en 6a, que son los mismos que los que supuestamente se tipearon en el paso 1.

A picture containing clock, dark, sitting, screen

Description automatically generated

Verificar con una tilde que dichos códigos coincidan con los determinados mediante tablas en el paso 1. En caso afirmativo, marcar las coincidencias usando un resaltador.

|  |  |  |  |
| --- | --- | --- | --- |
| **MEM** | **HEX** | **ASCII** | **CHAR** |
| 100 | 4D | 77 | M |
| 101 | 41 | 65 | A |
| 102 | 47 | 71 | G |
| 103 | 4E | 78 | N |
| 104 | 49 | 73 | I |
| 105 | 54 | 84 | T |
| 106 | 55 | 85 | U |
| 107 | 44 | 68 | D |
| 108 | 45 | 69 | E |
| 109 | 53 | 83 | S |
| 110 | 0D | 13 |  |
| 111 | 0A | 10 |  |
| 112 | 52 | 82 | R |
| 113 | 3D | 61 | = |
| 114 | 50 | 80 | P |
| 115 | 2B | 43 | + |
| 116 | 51 | 81 | Q |
| 117 | 2D | 45 | - |
| 118 | 54 | 84 | T |
| 119 | 0D | 13 |  |
| 120 | 0A | 10 |  |
| 121 | 50 | 80 | P |
| 122 | 3D | 61 | = |
| 123 | 31 | 49 | 1 |
| 124 | 32 | 50 | 2 |
| 125 | 33 | 51 | 3 |
| 126 | 20 | 32 |  |
| 127 | 51 | 81 | Q |
| 128 | 3D | 61 | = |
| 129 | 34 | 52 | 4 |
| 130 | 35 | 53 | 5 |
| 131 | 36 | 54 | 6 |
| 132 | 20 | 32 |  |
| 133 | 54 | 84 | T |
| 134 | 3D | 61 | = |
| 135 | 37 | 55 | 7 |
| 136 | 38 | 56 | 8 |
| 137 | 0D | 13 |  |
| 138 | 0A | 10 |  |

Analizar qué pasa cada vez que se tipea “Enter” (↵) en relación con lo supuesto en el paso 1.

Se disparan dos códigos de control: el CR (por “carriage return, retorno de carro”, código 13 del ASCII) seguido del LF (por “line feed, salto de línea”, código 10 del ASCII).

Los sistemas basados en ASCII o en una codificación de caracteres compatible usan LF (salto de línea), CR (retorno de carro), o CRLF (CR seguido de LF):

* LF: Unix y sistemas tipo Unix, Linux, AIX, Xenix, Mac OS X, BeOS, Amiga, RISC OS y otros
* CR+LF: CP/M, MP/M, DOS, Microsoft Windows
* CR: familia Apple II y Mac OS hasta la versión 9
* LF+CR: Las antiguas máquinas de escribir (por establecer la analogía con las computadoras)

El término viene de las máquinas de escribir, que tienen un cilindro llamado carro que es donde se apoya el papel. Cuando se ha escrito una línea completa, se acciona una palanca o mecanismo (llamada retorno de carro) que lo mueve hacia la izquierda. Las máquinas de escribir eléctricas sustituyeron esta palanca por una tecla, a la que llamaron retorno de carro o sólo retorno (return en inglés). Para describir mejor el significado de esta tecla, se la etiquetó con el símbolo **↵**, que ilustra la acción que hace sin necesitar palabras.

Las primeras impresoras eran muy parecidas a las máquinas de escribir, y cada vez que les llegaba el carácter de retorno de carro, activaban el proceso físico de retorno del carro. En realidad, más que las máquinas de escribir, el término “carriage return” y “line feed” son originarios de los teletipos ya que las máquinas de escribir mecánicas, al accionar la palanca para retornar el carro, también producían la acción de alzar la línea.

* + 1. Escritura en memoria, mediante el comando **E** del Debug, de las zona de da tos y la de instrucciones determinadas en 2a, 2c y 2d y verificación mediante lectura (nuevamente con el comando E) de que la escritura fue correcta. Esto implica los siguientes pasos:

**7a) Escritura de la zona de datos:** luego de haber realizado el paso 6c en el último renglón de la pantalla se verá un guión indicador que el Debug está esperando un comando. Al lado de este guión se escribirá:

E 1234 ↵ (E 2848 para el DNI ejemplificado)

Las primeras 4 cifras del DNI indicaban la dirección del comienzo de la zona de datos (paso 2a).

Luego de pulsar Enter (↵) el Debug responderá indicando el contenido de la dirección que acompaña al comando E. Dicho contenido va seguido de un punto. A la derecha de este punto el Debug está esperando que se escriba (si se desea) el nuevo contenido con que se quiere reemplazar el contenido indicado.

El nuevo contenido a escribir es el que aparece en la dirección simbolizada 1234 en la tabla determinada en el punto 2a. Después de escribirlo, se debe pulsar la barra espaciadora para evitar tipear nuevamente el comando E con la dirección siguiente**.** Así el Debug mostrará automáticamente el contenido de la dirección siguiente a la modificada a continuación del último contenido modificado.

Nuevamente aparecerá un valor que no interesa, acompañado por un punto. A la derecha de este punto, como se hizo anteriormente, hay que escribir el nuevo contenido que reemplazará al que se muestra, de acuerdo con el segundo renglón de la tabla determinada en el punto 2a. Luego, se debe pulsar la barra espaciadora. El proceso anterior se deberá repetir para cada dirección de memoria hasta escribir todos los datos de la tabla del paso 2a, en cuyo caso pulsar Enter (↵).1

**7b)** Volver a tipiar E 1234↵ y luego pulsar sucesivamente la barra espaciadora de modo que vayan apareciendo todos los contenidos escritos en el punto 7a. Verificar con una tilde que sean los mismos que figuran en la tabla determinada en 2a.

A close up of a screen

Description automatically generated

**7c) Escritura zona de instrucciones:** escribir al lado del guión del Debug:

‐E 5678 ↵

Dado que las últimas 4 cifras del DNI indicaban la dirección del **comienzo de la zona de instrucciones** (paso 2d). Luego proceder como en **7a**) pero ahora para escribir en memoria la tabla del paso 2d).

**7d)** Volver a tipear E 5678↵, y luego pulsar sucesivamente la barra espaciadora de modo que vayan apareciendo todos los contenidos escritos en 7c.

Verificar con una tilde que sean los mismos que figuran en la tabla determinada en 2d.

A screen shot of a computer

Description automatically generated

**7e)** Escribir al lado del guión del Debug el comando R IP↵. Debajo de este comando aparecerá el valor actual del IP seguido de dos puntos, debiendo escribirse en el tercer renglón la dirección donde comienza la primera instrucción, establecida en el paso 2d., seguida de ↵.

A close up of a screen

Description automatically generated

**7f) Escribir al lado del guión el comando R** ↵. Aparecerán en pantalla 3 renglones con el estado de distintos registros de la UCP. De esta información en este paso sólo interesa el valor del registro IP y en el tercer renglón a la izquierda el código de máquina de la primera instrucción de la secuencia.

# Anotar a continuación estos valores:

# IP =7420

# CODIGO DE LA PROXIMA INSTRUCCION A EJECUTAR = A17922

Verificar, indicando en el debug con una tilde o con un resaltador:

**7f.1)** Si el valor del IP coincide con la dirección donde comienza la primera instrucción (I1) establecida en el punto 2d.

**7f.2)** Si el código de máquina de la primera instrucción a ejecutar es el mismo que el anotado más arriba.

*En caso de que alguna de las verificaciones no coincidan, implica que hay un error en la realización de los puntos 7b o 7c.*

**A close up of a screen

Description automatically generated**

**7g)** Escribir al lado del guión del comando T ↵. Se ejecutará la instrucción I1 que apareció en pantalla al realizar el paso 7d. El Debug, como en el paso 7d, mostrará en pantalla 3 renglones de información.

*(El comando T siempre ejecuta el código de máquina de la instrucción que está arriba del mismo).*

Anotar a continuación los siguientes valores de la información hallada:

**AX** =00E3

**IP** =7423

**PROX INSTR A EJECUTAR** = **0306**8122

Verificar indicando en el debug con una tilde o con resaltador:

**7g.1)** Si el valor de **AX** coincide con el calculado en el punto 3 después de ejecutar I1.

**7g.2)** Si el valor de **IP** coincide con la dirección de comienzo de I2, establecida en 2d.

**7g.3)** Si el código de la próxima instrucción a ejecutar (I2) es el que aparece a partir del tercer renglón, observando la tabla del punto 2d.

IMAGEN CON TILDES EN EL PUNTO **7J**.

**7h)** Ídem 7g, con lo cual se ejecutará I2 volcando los siguientes valores:

**AX** =00E3

**IP** =7427

**PROX INSTR A EJECUTAR** = 2B068322

Verificar indicando en el debug con una tilde o con resaltador si AX coincide con el valor calculado y si IP y la próxima instrucción a ejecutar coinciden con lo establecido en 2d.

IMAGEN CON TILDES EN EL PUNTO **7J**.

**7i)** Ídem 7g, con lo cual se ejecutará I3 volcando los siguientes valores:

**AX** =00E3

**IP** =742B

**PROX INSTR A EJECUTAR** = A37722

Verificar indicando en el debug con una tilde o con resaltador si AX coincide con el valor calculado y si IP y la próxima instrucción a ejecutar coinciden con lo establecido en 2d.

IMAGEN CON TILDES EN EL PUNTO **7J**.

**7j)** Ídem 7g, con lo cual se ejecutará I4 volcando los siguientes valores:

**AX** =00E3

**IP** =742E

**PROX INSTR A EJECUTAR** = 0000

Verificar indicando en el debug con una tilde o con resaltador si AX coincide con el valor calculado (también indicado en el paso 4) y si IP y la próxima instrucción a ejecutar coinciden con lo establecido en 2d.

Después de este paso, en la pantalla deberán verse cuatro comandos T, en correspondencia con la ejecución de las 4 instrucciones que componen la secuencia.

A screen shot of a monitor

Description automatically generated

**7k)** La instrucción I4 de código de operación A3 ordena pasar el contenido de AX a la dirección de memoria que acompaña a A3. Para **verificar que se ejecutó esta instrucción** se debe examinar esta dirección de memoria y la siguiente, correspondientes a la ubicación de la variable R. Esto implica que en el Debug debe hacerse, de acuerdo con el paso 2a:

**‐E** dirección de la variable **R** (primeras 4 cifras del DNI) ↵

A screen shot of a computer

Description automatically generated

El valor de esta dirección y el de la siguiente **debe coincidir con el indicado en el paso 4**, en cuyo caso **se debe resaltar en el Debug esta coincidencia**. En caso contrario, debe buscarse el error y subsanarlo.

## SOBRE LOS 2 ÚLTIMOS PUNTOS

He intentado subsanar el resultado inesperado de especialmente el último punto (que es el que más me llamó la atención). Para esto, se ha revisado detalladamente el práctico de principio a fin y fui haciendo anotaciones en este mismo práctico para señalar con claridad cada uno de los pasos del total del procedimiento.

En la segunda revisión, trabajé con calculadora corroborando cada respuesta. En la tercera revisión completa, hice todo el proceso “a mano” a través de una prueba de escritorio y con la bibliografía al lado, también consultando información que podría esclarecer algún punto que se hubiera pasado por alto.

Del resultado de esa investigación, se adoptaron distintas formas de visar la información que el DEBUG puede entregar. A continuación, mostraré las pantallas que la última prueba que se realizó en el día de la fecha en una de las tres (3) computadoras en las que se estuvo repitiendo integralmente toda la experiencia. Se hace esto a fin de señalar algunos hechos remarcables:

Graphical user interface

Description automatically generated

Como puede verse en esta pantalla, al hacer un DUMP, puede apreciarse que los datos esperados coinciden plenamente a partir de la 3ra dirección de memoria (la 2279) y que el registro indica con claridad que 7420 es la dirección de la primera sentencia a ejecutar.

A picture containing graphical user interface

Description automatically generated

Si se ejecuta el UNASSEMBLE en 7420, puede apreciarse que las instrucciones están correctamente cargadas y que las direcciones de memoria a las que apunta cada instrucción son las esperadas (puede verificarse esto en las tablas pedidas, punto **2D**). Lo que particularmente llama la atención es que la dirección a la que debe copiarse el resultado (la 2277) está claramente definida en la última sentencia.

Text

Description automatically generated

Como se puede apreciar en esta pantalla, (aunque innecesario) se vuelve a establecer la IP en la dirección donde se encuentra la dirección de la primera instrucción.

Graphical user interface, text

Description automatically generated

Volcamos el registro, y puede verse con toda claridad que la IP es la esperada y que la dirección de la sentencia a ejecutar coincide con lo mostrado por el UNASSEMBLER. AX está en cero (honestamente, no sé si este valor inicial es el esperado, en este mismo instante estoy en duda con respecto a eso).

Graphical user interface, text

Description automatically generated

Empezamos con la prueba. Como puede verse, AX acompaña las ejecuciones y las próximas instrucciones anticipadas son las definidas.

A picture containing sign, holding, screen, street

Description automatically generated

Como puede verse, AX deja de cambiar en las últimas dos ejecuciones. No he encontrado razones claras para este comportamiento, pero como se hizo en la mayoría de las experiencia y repeticiones del procedimiento, termino con la ejecución de todas las sentencias para LUEGO poder hacer alguna evaluación de lo sucedido.

Graphical user interface, text

Description automatically generated

Y ahí está. Tal como pudo anticiparse del comportamiento de AX, el resultado no es el esperado en la dirección de memoria 2277.

**CONCLUSIONES**

Son difíciles. En todo lo que he leído para poder explicar esto (mayormente la documentación de Microsoft y blogs que han permanecido a pesar del tiempo) la información es verdaderamente escasa.

Un análisis honesto arroja algunas posibilidades:

* Que a pesar de todas mis revisiones y de todas las pruebas que hice, no llegué a ver si he cometido un error de procedimiento.
* La otra que he considerado es que la anomalía se presenta NO cuando YO escribo un valor en la memoria, sino que el error se produce cuando el DEBUG intenta hacerlo. Esto implicaría alguna especie de protección con la que cuenta la memoria y de la que no estoy anoticiado (¿tal vez el HIMEN?).

Dejo a usted profesor la decisión sobre este asunto.

Que tenga buenas noches.

1. VARIABLES: DIRECCIÓN Y CONTENIDO [↑](#footnote-ref-1)
2. **R=981 BIN: 0000 0011 1101 0101 HEX: 03D5** [↑](#footnote-ref-2)
3. **REFERENCIA 1 (VARIABLES)** [↑](#footnote-ref-3)
4. **REFERENCIA 2 (INSTRUCCIONES)** [↑](#footnote-ref-4)
5. QUE CORRESPONDE A LA VARIABLE **R**. [↑](#footnote-ref-5)