|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **IP** | **DL** | **CALL 0200** | **RET** | **SP** | **Stack** |
| 0102 | 41 |  |  | FFFE | 00 00 |
| 0105 | 41 | X |  | FFFE | 00 00 |
| 0200 | 41 |  |  | FFFC | 08 01 00 |
| 0202 | 41 |  |  | FFFC | 08 01 00 |
| 0204 | 41 |  |  | FFFC | 08 01 00 |
| 0206 | 42 |  | X | FFFC | 08 01 00 |
| 0108 | 42 |  |  | FFFE | 00 00 |
| 0105 | 42 | X |  | FFFE | 00 00 |
| 0200 | 42 |  |  | FFFC | 08 01 00 |
| 0202 | 42 |  |  | FFFC | 08 01 00 |
| 0204 | 42 |  |  | FFFC | 08 01 00 |
| 0206 | 43 |  | X | FFFC | 08 01 00 |
| 0108 | 43 |  |  | FFFE | 00 00 |
| 0105 | 43 | X |  | FFFE | 00 00 |
| 0200 | 43 |  |  | FFFC | 08 01 00 |
| 0202 | 43 |  |  | FFFC | 08 01 00 |
| 0204 | 43 |  |  | FFFC | 08 01 00 |
| 0206 | 44 |  | X | FFFC | 08 01 00 |
| 0108 | 44 |  |  | FFFE | 00 00 |
| 0105 | 44 | X |  | FFFE | 00 00 |
| 0200 | 44 |  |  | FFFC | 08 01 00 |
| 0202 | 44 |  |  | FFFC | 08 01 00 |
| 0204 | 44 |  |  | FFFC | 08 01 00 |
| 0206 | 45 |  | X | FFFC | 08 01 00 |
| 0108 | 45 |  |  | FFFE | 00 00 |
| 0105 | 45 | X |  | FFFE | 00 00 |
| 010A | 45 |  |  | FFFE | 00 00 |

-a 100  
0D41:0100 mov dl, 41  
0D41:0102 mov cx, 0004  
0D41:0105 call 0200  
0D41:0108 loop 0105  
0D41:010A int 20  
0D41:010C  
-a 200  
0D41:0200 mov ah, 02  
0D41:0202 int 21  
0D41:0204 inc dl  
0D41:0206 ret  
0D41:0207  
-t  
  
AX=0000  BX=0000  CX=0000  DX=0041  SP=FFEE  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0102   NV UP EI PL NZ NA PO NC  
0D41:0102 B90400        MOV     CX,0004  
-t  
  
AX=0000  BX=0000  CX=0004  DX=0041  SP=FFEE  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0105   NV UP EI PL NZ NA PO NC  
0D41:0105 E8F800        CALL    0200  
-t  
  
AX=0000  BX=0000  CX=0004  DX=0041  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0200   NV UP EI PL NZ NA PO NC  
0D41:0200 B402          MOV     AH,02  
-t  
  
AX=0200  BX=0000  CX=0004  DX=0041  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0202   NV UP EI PL NZ NA PO NC  
0D41:0202 CD21          INT     21

1. La instrucción CALL llama a los procedimientos. A esto lo realiza copiando en IP la dirección que contiene la instrucción y almacenando en la pila la dirección de la instrucción a realizar cuando finalice el procedimiento llamado.  
   La instrucción RET es de retorno. Esta instrucción copia en el IP la dirección que se encuentra en la cima de la pila.
2. La instrucción INT aumenta en uno al registro DL.

-a 100  
0D41:0100 call 200  
0D41:0103 int 20  
0D41:0105  
-a 200  
0D41:0200 call 300  
0D41:0203 ret  
0D41:0204  
-a 300  
0D41:0300 call 400  
0D41:0303 ret  
0D41:0304  
-a 400  
0D41:0400 ret  
0D41:0401  
-r ip  
IP 0100  
:100

-n Hola.com

-r bx

BX 0003

:0000

-r cx

CX 0001

:301

-w

Writing 00301 bytes  
-t  
  
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0200   NV UP EI PL NZ NA PO NC  
0D41:0200 E8FD00        CALL    0300  
-t  
  
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEA  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0300   NV UP EI PL NZ NA PO NC  
0D41:0300 E8FD00        CALL    0400  
-t  
  
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0400   NV UP EI PL NZ NA PO NC  
0D41:0400 C3            RET  
-t  
  
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEA  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0303   NV UP EI PL NZ NA PO NC  
0D41:0303 C3            RET  
-t  
  
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0203   NV UP EI PL NZ NA PO NC  
0D41:0203 C3            RET  
-t  
  
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0103   NV UP EI PL NZ NA PO NC  
0D41:0103 CD20          INT     20  
-p  
  
El programa ha terminado de forma normal  
  
-d ffe0  
0D41:FFE0  4F 03 6A 07 00 00 00 00-00 00 00 00 41 0D 7B 07  
0D41:FFF0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

1. Para "ver" los valores de la pila se utiliza la instrucción D, del Debug (y a continuación el valor del puntero de pila (su dirección), desde donde queremos ver la pila).

(OTRA VERCION) :Se visualizan los valores de la pila realizando un volcado de memoria en la dirección indicada por el registro de pila (SP).

1. Para saber en qué dirección se encuentra guardado, debemos observar  
   el puntero de pila (SP).

|  |  |  |  |
| --- | --- | --- | --- |
| **IP** | **Instrucción** | **SP** | **Stack** |
| 0100 | CALL 0200 | FFFF | 00 |
| 0200 | CALL 0300 | FFFD | 03 01 00 |
| 0300 | CALL 0400 | FFFB | 03 02 03 01 00 |
| 0400 | RET | FFF9 | 03 03 03 02 03 01 00 |
| 0303 | RET | FFFB | 03 02 03 01 00 |
| 0203 | RET | FFFD | 03 01 00 |
| 0103 | INT 20 | FFFF | 00 |

-a 100  
0D41:0100 mov cx, ffff  
0D41:0103 mov dx, eeee  
0D41:0106 call 200  
0D41:0109  
-a 200  
0D41:0200 push cx  
0D41:0201 push dx  
0D41:0202 mov cx, 0003  
0D41:0205 inc dl  
0D41:0207 loop 0205  
0D41:0209 pop dx  
0D41:020A pop cx  
0D41:020B ret  
0D41:020C  
-t  
  
AX=0000  BX=0000  CX=FFFF  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0103   NV UP EI PL NZ NA PO NC  
0D41:0103 BAEEEE        MOV     DX,EEEE  
-t  
  
AX=0000  BX=0000  CX=FFFF  DX=EEEE  SP=FFEE  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0106   NV UP EI PL NZ NA PO NC  
0D41:0106 E8F700        CALL    0200  
-t  
  
AX=0000  BX=0000  CX=FFFF  DX=EEEE  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0200   NV UP EI PL NZ NA PO NC  
0D41:0200 51            PUSH    CX  
  
-t  
  
AX=0000  BX=0000  CX=FFFF  DX=EEEE  SP=FFEA  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0201   NV UP EI PL NZ NA PO NC  
0D41:0201 52            PUSH    DX  
-t  
  
AX=0000  BX=0000  CX=FFFF  DX=EEEE  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0202   NV UP EI PL NZ NA PO NC  
0D41:0202 B90300        MOV     CX,0003  
-t  
  
AX=0000  BX=0000  CX=0003  DX=EEEE  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0205   NV UP EI PL NZ NA PO NC  
0D41:0205 FEC2          INC     DL  
-t  
  
AX=0000  BX=0000  CX=0003  DX=EEEF  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0207   NV UP EI NG NZ NA PO NC  
0D41:0207 E2FC          LOOP    0205  
-t  
  
AX=0000  BX=0000  CX=0002  DX=EEEF  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0205   NV UP EI NG NZ NA PO NC  
0D41:0205 FEC2          INC     DL  
-t  
  
AX=0000  BX=0000  CX=0002  DX=EEF0  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0207   NV UP EI NG NZ AC PE NC  
0D41:0207 E2FC          LOOP    0205  
-t  
  
AX=0000  BX=0000  CX=0001  DX=EEF0  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0205   NV UP EI NG NZ AC PE NC  
0D41:0205 FEC2          INC     DL  
-t  
  
AX=0000  BX=0000  CX=0001  DX=EEF1  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0207   NV UP EI NG NZ NA PO NC  
0D41:0207 E2FC          LOOP    0205  
-t  
  
AX=0000  BX=0000  CX=0000  DX=EEF1  SP=FFE8  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0209   NV UP EI NG NZ NA PO NC  
0D41:0209 5A            POP     DX  
-t  
  
AX=0000  BX=0000  CX=0000  DX=EEEE  SP=FFEA  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=020A   NV UP EI NG NZ NA PO NC  
0D41:020A 59            POP     CX  
-t  
  
AX=0000  BX=0000  CX=FFFF  DX=EEEE  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=020B   NV UP EI NG NZ NA PO NC  
0D41:020B C3            RET  
-t  
  
AX=0000  BX=0000  CX=FFFF  DX=EEEE  SP=FFF0  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0000   NV UP EI NG NZ NA PO NC  
0D41:0000 CD20          INT     20  
-p  
  
El programa ha terminado de forma normal

6-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IP** | **SP** | **Pila** | **PUSH** | **POP** | **CALL** | **RET** |
| 0100 | FFFE |  |  |  |  |  |
| 0103 | FFFE |  |  |  |  |  |
| 0106 | FFFE |  |  |  | X |  |
| 0200 | FFFC | 09 01 00 00 | X |  |  |  |
| 0201 | FFFA | FF FF 09 01 00 00 | X |  |  |  |
| 0202 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 0205 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 0207 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 0205 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 0207 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 0205 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 0207 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 0209 | FFF8 | EE EE FF FF 09 01 00 00 |  |  |  |  |
| 020A | FFFA | FF FF 09 01 00 00 |  | X |  |  |
| 020B | FFFC | FF FF 09 01 00 00 |  | X |  |  |
| 0109 | FFFE |  |  |  |  | X |
|  |  |  |  |  |  |  |

-a 100  
0D41:0100 mov ah, 08  
0D41:0102 int 21  
0D41:0104  
-g 104  
  
AX=0861  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0104   NV UP EI PL NZ NA PO NC  
0D41:0104 EE            OUT     DX,AL  
  
  
7) Lo que ocurre es al tener 08 asignado en AH y a continuación la instrucción INT 21, lee un carácter, pero no imprime el mismo en la pantalla.

-a 100  
0D41:0100 call 200  
0D41:0103 int 20  
0D41:0105  
-a 200  
0D41:0200 push dx  
0D41:0201 mov ah, 08  
0D41:0203 int 21  
0D41:0205 cmp al,30  
0D41:0207 jb 0203  
0D41:0209 cmp al, 46  
0D41:020B ja 0203  
0D41:020D cmp al,39  
0D41:020F ja 021b  
0D41:0211 mov ah, 02  
0D41:0213 mov dl,al  
0D41:0215 int 21  
0D41:0217 sub al, 30  
0D41:0219 pop dx  
0D41:021A ret  
0D41:021B cmp al, 41  
0D41:021D jb 0203  
0D41:021F mov ah, 02  
0D41:0221 mov dl,al  
0D41:0223 int 21  
0D41:0225 sub al, 37  
0D41:0227 pop dx  
0D41:0228 ret  
  
  
-r ip  
IP 0103  
:100  
-p  
0  
AX=0200  BX=0000  CX=0000  DX=0000  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0103   NV UP EI PL ZR NA PE NC  
0D41:0103 CD20          INT     20  
-r ip  
IP 0103  
:100  
-p  
9  
AX=0209  BX=0000  CX=0000  DX=0000  SP=FFEC  BP=0000  SI=0000  DI=0000  
DS=0D41  ES=0D41  SS=0D41  CS=0D41  IP=0103   NV UP EI PL NZ NA PE NC  
0D41:0103 CD20          INT     20

8-

Se lee un carácter sin imprimirlo, luego compara dicho carácter con los valores ASSI: “0” y “F”, volviendo a leer el carácter si se encuentra fuera de este rango. Luego vuelve a comparar el carácter con el valor en ASSI: “9”, si dicho carácter se encuentra entre “0” y “9” lo convierte en un numero hexadecimal, pasándolo como parámetro de retorno al programa principal. Por lo contrario, lo compara con el valor “A”, y si el carácter está comprendido entre “A” y “F” lo convierte a hexadecimal, pasándolo como parámetro de retorno. De lo contrario vuelve a leer el carácter.

9-

Se comprueba que el programa únicamente acepta los caracteres entre 0 y 9, no funcionando al ingresar “/” o “:”.

10-

-a 100

138F:0100 call 200

138F:0103 mov dl, al

138F:0105 mov cl, 04

138F:0107 shl dl, cl

138F:0109 call 200

138F:010C add dl, al

138F:010E mov ah, 02

138F:0110 int 21

138F:0112 int 20

-a 200

138F:0200 push dx

138F:0201 mov ah, 08

138F:0203 int 21

138F:0205 cmp al, 30

138F:0207 jb 0203

138F:0209 cmp al, 46

138F:020B ja 0203

138F:020D cmp al, 39

138F:020F ja 021b

138F:0211 mov ah, 02

138F:0213 mov dl, al

138F:0215 int 21

138F:0217 sub al, 30

138F:0219 pop dx

138F:021A ret

138F:021B cmp al, 41

138F:021D jb 0203

138F:021F mov ah, 02

138F:0221 mov dl, al

138F:0223 int 21

138F:0225 sub al, 37

138F:0227 pop dx

138F:0228 ret

-n Guardar.com

-r bx

BX 0000

:0000

-r cx

CX FFF1

:0129

-w

Writing 00129 bytes

11-

-a 100

138F:0100 mov dx, ax

138F:0102 push dx

138F:0103 call 300

138F:0106 int 21

138F:0108 pop dx

-a 300

138F:0300 mov dh, 25

138F:0302 int 21

138F:0304 ret

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **IP** | **SP** | **Valores en la pila** | | | | | |
| **FFEE** | **FFEC** | **FFEA** | **FFE8** | **FFE6** | **FFE4** |
| 0103 | FFEC | xxxx | 3B87 |  |  |  |  |
| 0300 | FFEA | xxxx | 3B87 | 0106 |  |  |  |
|  |  | xxxx | 3B87 | 0106 | 7234 | 1F52 | 0304 |
| 0304 | FFEA | xxxx | 3B87 | 0106 |  |  |  |
| 0106 | FFEC | xxxx | 3B87 |  |  |  |  |
|  |  | xxxx | 3B87 | 7234 | 1F52 | 0108 |  |
| 0108 | FFEC | Xxxx | 3B87 |  |  |  |  |
| 0109 | FFEE | xxxx |  |  |  |  |  |

12-

-a 100

138F:0100 push ax

138F:0101 call 300

138F:0104 int 21

138F:0106 pop dx

138F:0107 int 20

138F:0109

-a 300

138F:0300 mov dh, 20

138F:0302 int 21

138F:0304 ret

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **IP** | **SP** | **Valores en la pila** | | | | | |
| **FFAE** | **FFAC** | **FFAA** | **FFA8** | **FFA6** | **FFA4** |
| 0102 | FFAC | xxxx | 0211 |  |  |  |  |
| 0300 | FFAA | xxxx | 0211 | 0104 |  |  |  |
|  |  | xxxx | 0211 | 0104 | 7223 | 23F1 | 0304 |
| 0304 | FFAA | xxxx | 0211 | 0104 |  |  |  |
| 0104 | FFAC | xxxx | 0211 |  |  |  |  |
|  |  | xxxx | 0211 | 7223 | 23F1 | 0106 |  |
| 0106 | FFAC | xxxx | 0211 |  |  |  |  |
| 0108 | FFAE | xxxx |  |  |  |  |  |

13-   
-a 100  
0D41:0100 mov cx, fffe  
0D41:0103 mov ax, ffff  
0D41:0106 push ax  
0D41:0107 loop 106  
0D41:0109 int 20  
0D41:010B  
  
El programa no funciona, ya que repetirá las instrucciones PUSH y LOOP, FFFE veces (porque ese valor le hemos ingresado en CX: indicador de bucles), y terminará sobrescribiendo al programa, por lo que dará error.

El ultimo valor del SP es 0108.  
  
14-  
-a 100  
0D41:0100 mov cx, fffe  
0D41:0103 pop ax  
0D41:0104 loop 103  
0D41:0106 int 20  
0D41:0108  
  
Se produce un acarreo en el puntero de pila y la instrucción pop sobreescribe la memoria desde su comienzo hasta llegar al propio programa, donde se detiene su ejecución al volverse corrupto.  
  
15- FATLATLRS

El programa ubicado a la izquierda emula el funcionamiento de las instrucciones pop, sin embargo no altera el puntero de pila (SP).

16-a)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IP** | **SP** | **Valores en la pila** | | | | | | | |
| **FFEE** | **FFEC** | **FFEA** | **FFE8** | **FFE6** | **FFE4** | **FFE2** | **FFE0** |
| 0100 | FFEE | xxxx |  |  |  |  |  |  |  |
| 010A | FFEC | xxxx | 2002 |  |  |  |  |  |  |
| 010B | FFEA | xxxx | 2002 | 2362 |  |  |  |  |  |
| 0210 | FFE8 | xxxx | 2002 | 2362 | 0118 |  |  |  |  |
| 0211 | FFE6 | xxxx | 2002 | 2362 | 0118 | 6262 |  |  |  |
|  |  | xxxx | 2002 | 2362 | 0118 | 6262 | 7897 | 197D | 0217 |
| 0217 | FFE6 | xxxx | 2002 | 2362 | 0118 | 6262 |  |  |  |
| 6262 | FFEE | xxxx | 2002 | 2362 | 0118 |  |  |  |  |

16-b)

0109 Push bx

7F32 \* 10 + FFEC = 8F30C → bl: 02h

7F32 \* 10 + FFEB = 8F30B → bh: 20h

010A Push ax

7F32 \* 10 + FFEA = 8F30A → al: 62h

7F32 \* 10 + FFE9 = 8F309 → ah: 23h

010D Mov [3A00], CH

A320 \* 10 + 3A00 = A6C00 → ch: 62h

0111 Mov ah, [SI + 2B00]

A320 \* 10 + 3A00 = A6C00 → el valor que hay en la dirección 3A00.

0115 Call 0210

7F32 \* 10 + FFE8 = 8F308 → 18h

7F32 \* 10 + FFE7 = 8F307 → 01h

En las direcciones 8F308 y 8F307 de la pila se encuentra la dirección de retorno del Call.

0210 Push ax

7F32 \* 10 + FFE6 = 8F306 → al: 62h

7F32 \* 10 + FFE5 = 8F305 → ah: 62h

0215 INT 21

7F32 \* 10 + FFE4 = 8F304 → 97h Registro de

7F32 \* 10 + FFE3 = 8F303 → 78h Estado

7F32 \* 10 + FFE2 = 8F302 → 7Dh Segmento de

7F32 \* 10 + FFE1 = 8F301 → 19h Código

7F32 \* 10 + FFE0 = 8F300 → 17h Dirección de

7F32 \* 10 + FFDE = 8F2FE→ 02h Retorno

17)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IP** | **SP** | **Valores en la pila** | | | | | | | |
| **FFEE** | **FFEC** | **FFEA** | **FFE8** | **FFE6** | **FFE4** | **FFE2** | **FFE0** |
| 0100 | FFEE | xxxx |  |  |  |  |  |  |  |
| 010A | FFEC | xxxx | 0985 |  |  |  |  |  |  |
| 010B | FFEA | xxxx | 0985 | 6223 |  |  |  |  |  |
| 0310 | FFE8 | xxxx | 0985 | 6223 | 0118 |  |  |  |  |
| 0311 | FFE6 | xxxx | 0985 | 6223 | 0118 | 6223 |  |  |  |
|  |  | xxxx | 0985 | 6223 | 0118 | 6223 | 3100 | 0262 | 0317 |
| 0317 | FFE6 | xxxx | 0985 | 6223 | 0118 | 6223 |  |  |  |
| 6263 | FFEE | xxxx | 0985 | 6223 | 0118 |  |  |  |  |

0109 Push bx

60F0 \* 10 + FFEC = 70EEC→ bl: 85h

60F0 \* 10 + FFEB = 70EEB→ bh: 09h

010A Push ax

60F0 \* 10 + FFEA = 70EEA → al: 23h

60F0 \* 10 + FFE9 = 70EE9 → ah: 62h

010D Mov [5B00], CL

5FF0 \* 10 + 5B00 = 65A00→ cl: 23h

0111 Mov DL, [DI + 5A10]

5FF0 \* 10 + 5B00 = 65A00 → 23h, almacenado en la dirección 5B00.

0115 Call 0310

60F0 \* 10 + FFE8 = 70EE8→ 18h

60F0 \* 10 + FFE7 = 70EE7→ 01h

En las direcciones 70EE8 y 70EE7 de la pila se encuentra la dirección de retorno del Call.

0310 Push ax

60F0 \* 10 + FFE6 = 70EE6 → al: 23h

60F0 \* 10 + FFE5 = 70EE5 → ah: 62h

0315 INT 21

60F0 \* 10 + FFE4 = 70EE4 → 00h Registro de

60F0 \* 10 + FFE3 = 70EE3 → 31h Estado

60F0 \* 10 + FFE2 = 70EE2 → 62h Segmento de

60F0 \* 10 + FFE1 = 70EE1 → 02h Código

60F0 \* 10 + FFE0 = 70EE0→ 17h Dirección de

60F0 \* 10 + FFDE = 70EDE→ 03h Retorno

18)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IP** | **SP** | **Valores en la pila** | | | | | | | |
| **E9DE** | **E9DC** | **E9DA** | **E9D8** | **E9D6** | **E9D4** | **E9D2** | **E9D0** |
| 0100 | E9DE | xxxx |  |  |  |  |  |  |  |
| 010A | E9DC | xxxx | 040B |  |  |  |  |  |  |
| 0116 | E9DA | xxxx | 040B | 000A |  |  |  |  |  |
| 0120 | E9D8 | xxxx | 040B | 000A | 0119 |  |  |  |  |
| 0121 | E9DA | xxxx | 040B | 000A |  |  |  |  |  |
| 0124 | E9DC | xxxx | 040B |  |  |  |  |  |  |
|  |  | xxxx | 040B | 7898 | A723 | 0126 |  |  |  |
| 0126 | E9DC | xxxx | 040B |  |  |  |  |  |  |
| 040B | E9DE | xxxx |  |  |  |  |  |  |  |

0109 Push cx

A723 \* 10 + E9DC = B5C0C → bl: 0Bh

A723 \* 10 + E9DB = B5C0B → bh: 04h

010A Push dx

A723 \* 10 + E9DA = B5C0A → al: 0Ah

A723 \* 10 + E9D9 = B5C09 → ah: 00h

010D Mov DS:[1B34], dh

B520 \* 10 + 1B34 = B6D34 → 00h

0112 Mov [SI + 0500], cx

B520 \* 10 + 0A00 = A7C30 → cl: 0Bh

B520 \* 10 + 0A01 = A7C31 → ch: 04h

0116 Call 0120

7A81 \* 10 + E9D8 = 891E8 → 19h

7A81 \* 10 + E9D7 = 891E7 → 01h

0124 INT 21

7A81 \* 10 + E9DA = 891EA → 98h Registro de

7A81 \* 10 + E9D9 = 891E9 → 78h Estado

7A81 \* 10 + E9D8 = 891E8 → 23h Segmento de

7A81 \* 10 + E9D7 = 891E7 → A7h Código

7A81 \* 10 + E9D6 = 891E6 → 26h Dirección de

7A81 \* 10 + E9D5 = 891E5 → 01h Retorno

19) A)

0107 Push ax

167D \*10 + FFCC = 2679C → al: 80h

167D \*10 + FFCB = 2679B → ah: 09h

0108 Push bx

167D \*10 + FFCA = 2679A → bl: A1h

167D \*10 + FFC9 = 26799 → bh: 01h

010B Mov [1B34], ax

167D \* 10 + 1B34 = 18304 → al: 21h

167D \* 10 + 1B35 = 18305 → ah: 0Bh

010E Mov [bx + 100], bl

167D \* 10 + 02A1 = 16A71→ bl: A1h

19)B)

167D \* 10 + 0114 = 168E4 → Dirección de la próxima instrucción.

20)

022A INT 21

ED61 \* 10 + FFE6 = FD5F6 Registro de

ED61 \* 10 + FFE5 = FD5F5 Estado

ED61 \* 10 + FFE4 = FD5F4 Segmento de

ED61 \* 10 + FFE3 = FD5F3 Código

ED61 \* 10 + FFE2 = FD5F2 Dirección de

ED61 \* 10 + FFE1 = FD5F1 Retorno

022C RET

ED61 \* 10 + FFE8 = FD5F8 → 10h

ED61 \* 10 + FFE7 = FD5F7 → 01h

0109 Add ax, F3D3

152A \* 10 + 010A = 153AA →D3h

152A \* 10 + 010B = 153AB → F3h

0106 Mov [5161], ax

152A \* 10 + 5161 = 1A401 → 10h

152A \* 10 + 5162 = 1A402 → 01h

El valor del IP al finalizar el programa será 0113, ya que la INT ocupa 2 bytes.