19

Debug

Cómo:

Capturar un programa, Ejecutarlo, Salvarlo, Recargarlo, Desensamblarlo y Volcarlo

20

Captura del programa

- Iniciar el debug
  - Acceder a una consola de msdos desde:
    - Menú Inicio/ejecutar/**cmd** o **command**
- Capturar un programa que "no hace nada":

-a (*solicita el ensamblado de un programa*)
0CD3:0100 mov ah,4c ↵
0CD3:0102 int 21 ↵
0CD3:0104 ↵

21

Ejecución del programa

- La ejecución se puede hacer vía tres comandos:
  - t (trazar) o p (proceder) o g (ejecutar)
  - Trazar ejecuta instrucciones paso a paso. No se debe usar con las interrupciones (int).
  - Proceder ejecutar grupos de instrucciones como el caso de llamadas a subrutinas o interrupciones.
- Ejemplo:

```
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0CD3  ES=0CD3  SS=0CD3  CS=0CD3  IP=0100   NV UP EI PL NZ NA PO NC
0CD3:0100 B44C          MOV     AH,4C
-t
AX=4C00  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0CD3  ES=0CD3  SS=0CD3  CS=0CD3  IP=0102   NV UP EI PL NZ NA PO NC
0CD3:0102 CD21          INT     21
-p
```


22

Salvar el programa

- Antes de almacenar el código primero debemos conocer su longitud, para ello usamos el comando de hexaritmética h.
- Se obtiene la diferencia de la última línea menos la primera:
  - h 104 100
  - 0204 0004

Última línea donde ↵ esta en blanco
Primera línea
Diferencia

- Se modifica el contenido de CX, con el comando r
  - r cx ↵

CX 0000
:4 ↵

23

Salvar el programa

- Se asigna un nombre al programa
  - n primero.com
- Se pide al debug que guarde el programa con el comando w (escribir)
  - w

Writting 00004 bytes

24

Recargarlo

- Existen dos métodos para volver a abrir un programa:
  - Escribir desde la línea de comandos:
    - Debug programa.xxx (programa es el archivo con extensión exe o com)
  - Invocarlo internamente desde el debug:
    - n programa.xxx (indicar el nombre del archivo)
    - L programa.xxx (solicitar se cargue a memoria)
- Nota: solo obtendremos mensaje si: el archivo no se encontró o hubo error.

25

Desensamblar un programa

- El desensamblar un programa es útil cuando queremos ver el código de un programa que no escribimos o bien que deseamos modificar.
- Para ello se usa el comando u (desensamblar):
  - u <dirección> <rango>

-u 100 L5 (*solicita desensamblar de la dirección 100, 5 bytes de Longitud*)

```
0CD3:0100 6E            DB      6E
0CD3:0101 206172        AND     [BX+DI+72],AH
0CD3:0104 63            DB      63
```


26

Volcarlo

- El volcado de memoria permite visualizar los bytes tal cual en formato hexadecimal y ascii, para esto se usa el comando d (volcar)
  - d 100 (*se pide volcar a partir de la dirección 100, por omisión muestra 128 bytes: 16 columnas por 8 renglones*):

```
-d 100
0CD3:0100  6E 20 61 72 63 68 69 76-6F 20 73 6F 72 65 20   n archivo sobre
0CD3:0110  A3 A1 20 6D 69 73 6D 6F-0D 0A 1F 45 34 00 C2 0C   s. mismo...E4...
0CD3:0120  69 6F 20 69 6E 73 75 66-69 63 69 65 6E 74 65 20   io insuficiente
0CD3:0130  65 6E 20 64 69 73 63 6F-0D 0A 1D 50 A0 67 69 6E   en disco...P.gin
0CD3:0140  61 20 64 65 20 63 A2 64-69 67 6F 73 20 6E 6F 20   a de c.digos no
0CD3:0150  76 A0 6C 69 64 61 0D 0A-11 46 65 63 68 61 20 6E   v.lida...Fecha n
0CD3:0160  6F 20 76 A0 6C 69 64 61-0D 0A 10 48 6F 72 61 20   o v.lida...Hora
0CD3:0170  6E 6F 20 76 A0 6C 69 64-61 0D 0A 10 52 75 74 61   no v.lida...Ruta
```


27

Modificando el reg. Flags

- Las banderas indican el estado del CPU tras alguna operación a continuación se muestra su nombre cuando están activas y no activas:

Flag Name	Set	Clear
Overflow (yes/no)	OV	NV
Direction (decrease/increase)	DN	UP
Interrupt (enable/disable)	EI	DI
Sign (negative/positive)	NG	PL
Zero (yes/no)	ZR	NZ
Auxiliary carry (yes/no)	AC	NA
Parity (even/odd)	PE	PO
Carry (yes/no)	CY	NC

28

Archivos

- ✔ Exe
- ✓ Com

29

Introducción

- En éste curso se estudian los programas del tipo exe y com. Tales archivos los conocemos como archivos ejecutables.
- La diferencia entre estos archivos esta en la forma de cómo se escribe el código.
- El com respecto al exe, no tiene mayor ventaja que ser unos 200 bytes más compacto.

30

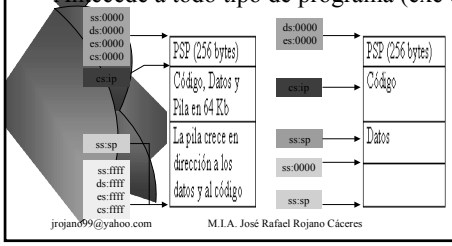
Diferencias .com & .exe

- Los .com proceden del CP/M, en tiempos de poca memoria, estos no rebasaban los 64 kb.
- Los .exe pueden ser del tamaño de la Ram, son inclusive mayores que ésta.
- Los .com incluyes en un solo segmento: datos, instrucciones y pila.
- Los .exe disponen de segmentos diferentes para las áreas mencionadas en el punto anterior
- En ambos casos el DOS define un área llamada PSP para la cuando éstos archivos se están ejecutando.

31

PSP

- El Program Segment Prefix es una estructura de datos en memoria de 256 bytes de longitud.
- Antecede a todo tipo de programa (exe o com).



32

✓ Com

- Los programas .com son copias directas de memoria (datos y código no cambian), además no deben tener llamadas externas.
- Los programas .com tienen máximo 64 kb como se dijo anteriormente, menos los 256b del psp y al menos 1 palabra para la pila
- El .com inicia en la dirección 100h su ejecución
- El puntero de pila esta en FFFEh al final del segmento
- El programador debe cuidar que la pila no sobre escriba al código.
- La pila crece a razón de 2 bytes por llamadas en dirección del programa principal.

33

✔ Exe

- Éstos no están limitados a 64 kb
- Son más complejos ya que guardan mucha información que les permite adaptarse a diferentes ambientes
- Los programas .exe no se **cargan** directamente a memoria. En lugar de ello deben ser **cargados** a través de una función de **exec** del dos.
- Los exe al igual que los com, se cargan en posiciones de memoria divisibles por 16.

34

✔ Exe

- Los .exe pueden tener llamadas FAR, para ello el .exe cuenta con una **cabecera de archivo** que es como una tabla de relocalización creada por **el ligador**.