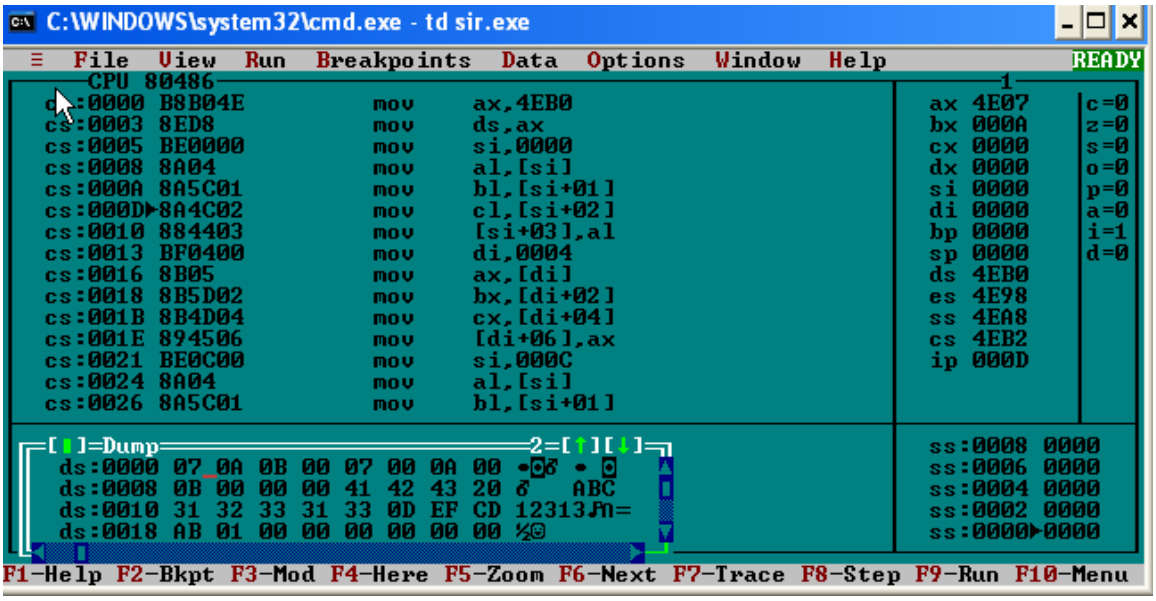
EXEMPLU DE UTILIZARE ŞI EXPLOATARE A MEMORIEI INTERNE

Să se determine harta memoriei pentru datele de mai jos şi offset-urile (deplasările) şirurilor

```
segm_stiva  SEGMENT
        stiva      DB 128 DUP (?)
segm_stiva  ENDS

segm_date  SEGMENT
        sir_date    DB      7,10,11, ?
        sir_date2   DW       7,10,11, ?
        sir_car     DB      'ABC 123'
        car         DB      '13'
        nr          DB       13
        nr_mare     DD       1abcdefh
segm_date  ENDS

segm_cod    SEGMENT
        ASSUME CS:segm_cod, DS:segm_date, SS:segm_stiva

        et:     MOV AX, segm_date
                MOV DS, AX

                MOV  SI, offset sir_date        ;  SI = 00 00
                MOV  AL, [SI]                   ; AX= ....07
                MOV  BL, [SI+1]                 ; BX= ....0A
                MOV  CL, [SI+2]                 ; CX= ....0B
                MOV  [SI+3], AL                 ; valoarea 07 se mută la adresa 0003

                MOV  DI, OFFSET sir_date2       ;  DI = 00 04
                MOV  AX, [DI]                   ; AX= 00 07
                MOV  BX, [DI+2]                 ; BX= 00 0A
                MOV  CX, [DI+4]                 ; CX= 00 0B
                MOV  [DI+6], AX                 ; valoarea 00 07 se mută la adresa
000A

                MOV   SI, OFFSET sir_car        ;  SI = 00 0C
                MOV  AL, [SI]                   ; AX= ....41
                MOV  BL, [SI+1]                 ; BX=.... 42
                MOV  CL, [SI+2]                 ; CX=.... 43

                MOV   DI, OFFSET car            ;  DI = 00 13
                MOV  AL,car                     ; AX= .....31
                MOV  AH,nr                      ; AX= 0D ...

                MOV SI, OFFSET nr_mare                  ; SI = 00 16
                MOV AX, WORD PTR [nr_mare]              ; AX= CD EF
                MOV BX, WORD PTR [nr_mare+2]            ; BX= 01  AB
```

```
                MOV AX, 4C00H
                INT 21H

segm_cod   ENDS

END et
```

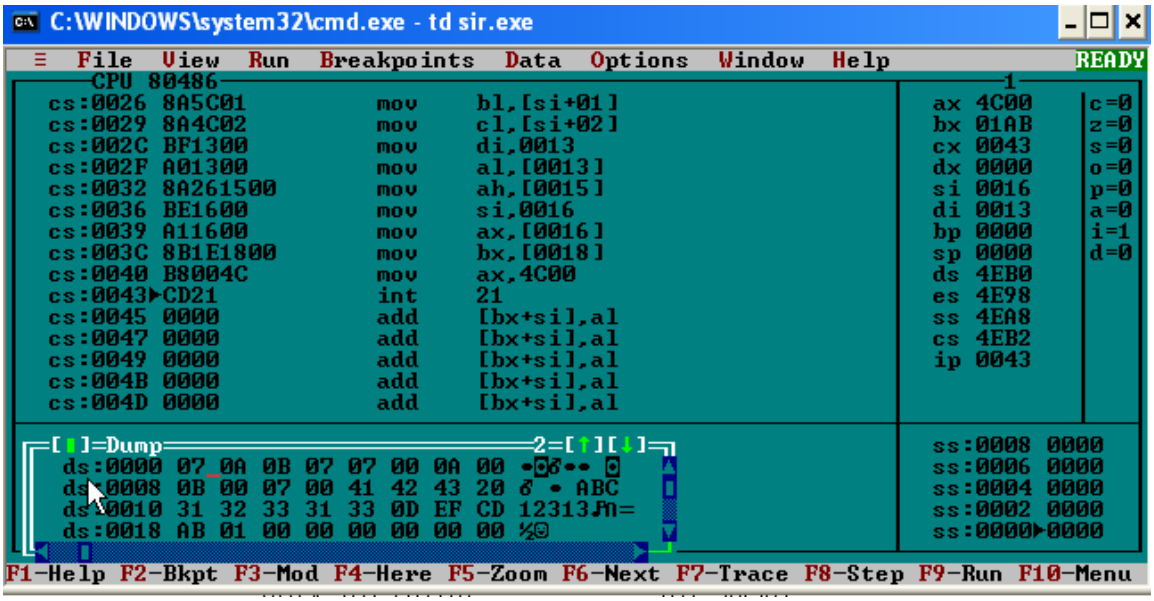


Harta memoriei la începutul programului:

sir_date	*sir_date2*	*sir_car*	*car*	*nr*
07 0A 0B 00	07 00 0A 00 0B 00 00 00	41 42 43 20 31 32 33	31 33	0D

EF CD AB 01
---
nr_mare

OBS.: În memorie, numerele la nivel de B (byte) şi caracterele (şirurile de caractere) sunt reprezentate în ordinea normală, iar numere la nivel de W (word) şi DD (double word) sunt reprezentate invers (se inversează byte-s între ei) – reprezentarea „little endian”



Harta memoriei la sfârşitul programului:

sir_date	*sir_date2*	*sir_car*	*car*	*nr*
07 0A 0B 07	07 00 0A 00 0B 00	07 00 41 42 43 20 31 32 33	31 33	0D

EF CD AB 01

---

nr_mare

Moduri de reprezentare nr

- negative – in CC,
- nr reale – in VMSP => pe 4B folosind DD (Define Double Word)

```
segm_stiva   SEGMENT
   stiva     DB 128 DUP (?)
segm_stiva   ENDS

segm_date    SEGMENT
        rez1    DB      ?
        rez2    DB      ?
        rez3    DB      ?
        nr1     DB      -128    ;(80)H
        nr2     DW      -128    ;(80 FF)H
        nr3     DB      -130    ;(7E)H
        nr4     DW      -130    ;(7E FF)H
        nr5     DB      +255    ;(FF)H
;       nr6     DB      +256  -da eroare la reprez -"value out of range"
;       nr6_6   DB      -257  -da eroare la reprez -"value out of range"
        nr6_7   DB      -256    ;(00)H
        nr7     DW      +256    ;(00 01)H

        sir_date   DB  7,-13,130, ?  ;(07, F3, 82, 00)H
        sir_date2 DW  7,-13,+130,?  ;(07 00, F3 FF, 82 00, 00 00)H
        sir_date3 DD  7.0, -13.0, +130.0, ?    ; (00 00 E0 40, 00 00 50 C1, 00 00 02 43)H
segm_date    ENDS
```

```
segm_cod    SEGMENT
   ASSUME CS:segm_cod, DS:segm_date, SS:segm_stiva
   et: MOV AX, segm_date
      MOV DS, AX

         MOV AL, -1
         ADD AL, -127
         MOV   rez1, AL

         MOV AH, -128
         ADD AH, 1
         MOV   rez2, AH

         MOV BL, -128
         ADD BL, -2   ; depasire interval reprez pt 8b; ramane rez  inainte de adunare, si O=1
         MOV   rez3,BL

      MOV  SI, offset sir_date
      MOV  AL, [SI]
      ADD  AL, [SI+1]
      SUB  AL, [SI+2]
      MOV [SI+3], AL

      MOV DI, offset sir_date2
      MOV  AX, [DI]
      ADD  AX, [DI+2]
      SUB  AX, [DI+4]
      MOV [DI+6], AX

      MOV AX, 4C00H
      INT 21H
segm_cod   ENDS
END et
```