

A5. Dezvoltarea programelor în limbaj de asamblare (II)

A5.1 Definirea simplificată a segmentelor

Variantele mai noi ale asamblelor au introdus o modalitate de definire simplificată a segmentelor, cu avantajul că se respectă același format (structura programului obiect) cu programele scrise în limbaje de nivel înalt. Utilizând definirea simplificată se vor genera segmente cu nume și atribute identice cu cele obținute cu compilatoarele pentru limbaje de nivel înalt.

Pentru specificarea tipului de memorie folosit se va folosi directiva **.MODEL** cu sintaxa dată mai jos:

.MODEL tip

unde tip poate fi: *tiny*, *small*, *medium*, *large*, *huge*. Semnificația lor este următoarea :

- *tiny* -toate segmentele (date, cod, stiva) se pot genera într-o zonă de 64Ko și alcătuiesc un singur grup de segmente; -salturile, apelurile și definițiile de proceduri sunt toate NEAR -este utilizat la programele .COM;
- *small* -datele+stiva formează un segment iar codul un alt segment. Segmentele sunt <64Ko; -salturile, apelurile și definițiile de proceduri sunt toate NEAR
- *medium* -datele+stiva formează un segment <64Ko, iar codul poate fi în mai multe segmente separate (negrupate), deci >64Ko; -salturile și apelurile și definițiile de proceduri sunt implicit de tip FAR;
- *compact* -datele și stiva se găsesc în segmente separate (deci > 64Ko) iar codul grupat <64Ko; -salturile și apelurile sunt de tip NEAR, iar datele sunt cu referință de tip FAR;
- *large* -datele și codul >64Ko; nici o structură de date nu va depăși 64Ko ;
- *huge* -datele și codul >64Ko; nu sunt restricții asupra structurilor de date; -datele, codul și pointerii vor fi cu referință îndepărtată;

Sintaxele directivelor simplificate de definire a segmentelor sunt prezentate mai jos:

```
. stack dimensiune ; implicit 512 octeți pentru TASM
. code [nume]
. data
. data? ; date neinițializate în limbaj de nivel înalt
```

```
. fardata [nume] ; seg. de date utilizate prin adrese complete
                  ; in LNI
. fardata [nume]
. const           ; definire constante în LNI (HLL)
```

Dacă parametrul [nume] lipsește se vor atribui nume implicite segmentelor generate:

```
TEXT             pentru .code (modele de cod mic)
Nume_fis_sursa_TEXT pentru (modele de cod mare)
DATA             pentru .data
_BSS             pentru .data?
CONST            pentru .const
STACK            pentru .stack
_FAR_DATA        pentru .fardata
_FAR_BSS         pentru .fardata?
```

O parte din segmente se grupează în mod implicit într-un grup numit DGROUP, după cum urmează:

- la modelul tiny intră toate segmentele
- la restul modelelor intră segmentele DATA, DATA?, CONST, STACK.

Există o directivă ASSUME implicită, de forma:

- la modelele small/compact:

```
ASSUME cs:TEXT, ds:DGROUP, ss:DGROUP
```

- la modelele large/huge:

```
ASSUME cs:nume_TEXT, ds:DGROUP, ss:DGROUP
;nume apare în directiva .code sau numele fis_sursa.asm
```

- la modelul tiny:

```
ASSUME cs:DGROUP, ds:DGROUP, ss:DGROUP.
```

Folosirea directivelor .fardata/ .fardata? cere utilizarea explicită a directivei ASSUME.

Accesul la adresele de început ale segmentelor sau grupurilor de segmente se face prin simbolurile globale: @data, @data?, @fardata, @fardata?, dgroup.

Inițializarea regiștrilor segment DS și ES la începutul unui program se poate face cu simbolul @data sau dgroup.

```
MOV AX,DGROUP ; SAU @DATA
MOV DS,AX
MOV ES,AX.
```

A5.2 Aplicație

Folosind definirea simplificată a segmentelor, în aplicația următoare este determinată suma a două numere

```
.MODEL SMALL
.STACK 200H
.DATA
    A DB 3
    B DB 5
    SUMA DB ?
.CODE
MAIN LABEL
    MOV AX,@DATA
    MOV DS,AX
    MOV BX, OFFSET A
    MOV AL, [BX]
    MOV BX, OFFSET B
    ADD AL,[BX]
    MOV BX, OFFSET SUMA
    MOV [BX],AL
    MOV AX,4C00h
    INT 21h
END MAIN
```

Listingul aplicației:

```
Turbo Assembler          Version 4.1          04/22/04 10:14:02
Page 1
first1.ASM
1 0000                .MODEL SMALL
2 0000                .STACK 200H
3 0000                .DATA
4 0000 03                A DB 3
5 0001 05                B DB 5
6 0002 ??                SUMA DB ?
7 0003                .CODE
8 0000                MAIN LABEL
9 0000 B8 0000S          MOV AX,@DATA
10 0003 8E D8           MOV DS,AX
11 0005 BB 0000R        MOV BX, OFFSET A
12 0008 8A 07           MOV AL, [BX]
13 000A BB 0001R        MOV BX, OFFSET B
14 000D 02 07           ADD AL,[BX]
15 000F BB 0002R        MOV BX, OFFSET SUMA
16 0012 88 07           MOV [BX],AL
17
18                                END MAIN

Turbo Assembler          Version 4.1          04/22/04 10:14:02
Page 2
Symbol Table
```

| Symbol Name | Type | Value |
|-------------|--------|-------------|
| ??DATE | Text | "04/22/04" |
| ??FILENAME | Text | "first1 " |
| ??TIME | Text | "10:14:02" |
| ??VERSION | Number | 040A |
| @32BIT | Text | 0 |
| @CODE | Text | _TEXT |
| @CODESIZE | Text | 0 |
| @CPU | Text | 0101H |
| @CURSEG | Text | _TEXT |
| @DATA | Text | DGROUP |
| @DATASIZE | Text | 0 |
| @FILENAME | Text | FIRST1 |
| @INTERFACE | Text | 000H |
| @MODEL | Text | 2 |
| @STACK | Text | DGROUP |
| @WORDSIZE | Text | 2 |
| A | Byte | DGROUP:0000 |
| B | Byte | DGROUP:0001 |
| MAIN | Word | _TEXT:0000 |
| SUMA | Byte | DGROUP:0002 |

| Groups & Segments | Bit | Size | Align | Combine | Class |
|-------------------|-----|------|-------|---------|-------|
| DGROUP | | | | | |
| GROUP | | | | | |
| STACK | 16 | 0200 | PARA | STACK | STACK |
| _DATA | 16 | 0003 | WORD | PUBLIC | DATA |
| _TEXT | 16 | 0014 | WORD | PUBLIC | CODE |