UNIUNEA EUROPEANĂ
GUVERNUL ROMÂNIEI
Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Programare în limbaj de asamblare

21. Inițializarea registrelor segment, asocierea segmentelor cu registrele.

Inițializarea registrelor segment

Registrele CS:IP sunt inițializate de SO (sistemul de operare) la încărcarea programului executabil în memorie pentru execuție astfel încât ele să conțină adresa primei instrucțiuni ce trebuie executată din program. Această instrucțiune este etichetată şi se specifică prin intermediul directivei END:

```
END   [adresa_de_start]
```

<adresa_de_start> este o adresă sau expresie care specifică prima instrucțiune executabilă şi care va inițializa conținutul registrului IP; acesta poate fi inițializat şi cu pseudoinstrucțiunea ORG.

Adresa de start este opțională deoarece poate lipsi pentru unele module de program (cele care conțin date sau proceduri). CS va fi inițializat de SO cu prima adresă de segment disponibilă, iar IP cu offsetul, din cadrul segmentului, al primei instrucțiuni executabile.

Pentru perechea de registre SS:SP care face referirea la stivă, inițializarea va fi executată de utilizator, dacă se folosesc directive complete de segmentare şi nu se specifică tipul şi clasa segmentului de stivă. Dacă, însă, se utilizează directive simplificate sau directive complete, cu precizarea tipului şi clasei stack, atunci cele două registre vor fi inițializate automat (SP va fi inițializat cu dimensiunea segmentului de stivă, declarată în directiva *.stack*, sau rezervată în declararea completă a segmentului de stivă).

Dacă segmentul este declarat astfel:

```
stiva   segment
        dw      100     dup (?)                 ; se rezerva 100 cuvinte pt. stiva
        varf_stiva      label           word
stiva   ends
```

atunci utilizatorul trebuie sa inițializeze cele două registre:

```
        assume cs: seg_cod, ds: seg_date, ss: stiva
                .............
                mov     ax, seg_date            ; initializare DS
                mov     ds, ax
                mov     ax, stiva               ; initializare SS
                mov     ss, ax
                mov     sp, offset varf_stiva   ; initializare SP
        .............
```

În ceea ce priveşte celelalte registre segment (DS, ES etc.), acestea trebuie inițializate explicit şi întotdeauna de către programator, conform directivei ASSUME (ca în exemplele anterioare).

Asocierea segmentelor cu registrele
Pseudoinstrucțiunea ASSUME

Deoarece într-un program în limbaj de asamblare pot exista mai multe segmente, de diferite tipuri, asamblorul trebuie informat în legătură cu cele 4 segmente logice curente. Acest lucru se realizează cu pseudoinstrucțiunea ASSUME:

```
ASSUME          <reg_segment>:<segment>
```

unde <reg_segment> poate fi unul dintre registrele CS, DS, ES, sau SS, iar este numele unui segment sau grup de segmente care va fi adresat de registrul segment respectiv; segmentul mai poate fi precizat şi de operatorul SEG, urmat de numele unei variabile sau numele unei etichete din segmentul respectiv:

```
assume cs: code, ds: dgrup, es: SEG var_a, ss: SEG varf_stiva
```

Pentru anularea unei astfel de asocieri se poate utiliza una din formele:

```
assume      <reg_seg>     : NOTHING    ; care anuleaza asocierea

; reg_seg la orice segment, facuta anterior
assume NOTHING          ; sau care anuleaza asocierile facute pentru
                        ; toate registrele.
```

De remarcat că ASSUME este o pseudoinstrucțiune pentru care nu se generează cod, rolul său fiind numai de a informa asamblorul cu privire la intenția programatorului. Asamblorul nu are nici un mecanism prin care să verifice dacă într-adevăr conținutul registrului segment este cel specificat în pseudoinstrucțiunea ASSUME. În schimb această pseudoinstrucțiune este utilizată de asamblor pentru a face referire la date şi instrucțiuni. Când o instrucțiune face o referire la o variabilă, fără prefix segment, asamblorul determină segmentul ce conține variabila şi apoi examinează pseudoinstrucțiunea ASSUME, pentru a determina registrul segment ce adresează acel segment.

Asocierea dintre segmentul logic şi registrul segment trebuie făcută înainte de a face referire la vreo informație din segmentul logic; deoarece aceste referiri se fac, de regulă, în segmentul de cod, directiva este plasată la începutul segmentului de cod sau, dacă se modifică aceste asocieri, înainte de prima referire la date din segmentul respectiv.

Să considerăm următoarele declarații:

```
data1   segment
                alfa            db      5
                beta            dw      25
data1   ends
data2   segment
                var1            dw      0
                var2            dw      100
data2   ends
cod             segment
                assume      cs:cod, ds:data1, es:data2
;               initializarea registrelor segment conform lui ASSUME
                start:  mov     ax, data1
                        mov     ds, ax
                        mov     ax, data2
                        mov     es, ax
;                       referiri la date
                        mov     bx, beta        ; se va utiliza DS, beta  data1  DS
                        mov     alfa, al        ; alfa  data1  din ASSUME  DS
```

```
                    add         var1, bx        ; var1  data2  ES
;                       modificam continutul lui DS
                    mov         ax, cs
                    mov         ds, ax
                    assume ds:cod
                    mov         data_cs, bx     ; se va utiliza DS
                    mov         cs:data_cs, cx  ; utilizare explicita a lui CS
                    ..............
        data_cs             dw          1000
cod         ends
end         start
```

Prefix segment

Dacă o referință la o variabilă nu este acoperită de o directivă ASSUME, asamblorul poate fi informat în mod explicit asupra registrului segment al variabilei prin codificarea unui prefix segment de forma:

reg_seg : instructiune (referinta)

anterior variabilei respective. Sarcina inițializării acestui registru segment revine, ca şi în cazul anterior, utilizatorului.

Cu toate că această construcție prezintă avantajul că nu necesită directiva ASSUME, ea are două dezavantaje:

- este valabilă doar pentru o instrucțiune; în loc de ASSUME, prefixul segment trebuie precizat pentru fiecare referință la o variabilă;
- erorile sunt uşor de comis, deoarece se referă la un registru segment şi nu la un nume de segment, şi deci este mai uşor de omis.

Exemplu de utilizare:

```
exemplu         segment
        assume      cs:     exemplu
        data_cs     dw      1000
        start:      mov     ax, seg rez     ; initializare registre segment
                    mov     es, ax          ; ES si DS cu segmentele respective
                    mov     ax, seg val1
                    mov     ds, ax
                    mov     ax, data_cs             ; data_cs exemplu CS
                    add     ax, ds: val1            ; utilizare explicita a lui DS
                    mov     es: rez, ax             ; utilizare explicita a lui ES
                                .......
exemplu             ends
end                 start
```

Referințe anonime

Variabile la care se face referire sub forma:

[bx],

[bp],
word ptr [si],
byte ptr [di],
[bx].camp,

sunt denumite referințe anonime, deoarece nu este precizat numele variabilei din care să se poată determina segmentul căruia îi aparțin. În astfel de situații, registrele segment utilizate pentru adresare sunt determinate în mod implicit pe baza unor reguli.

9.6. Reguli pentru determinarea registrului segment implicit

Aceste reguli se referă la cazurile în care în instrucțiune nu apar referiri explicite la registrele segment (prefix de segment) sau nu apar nume de variabile din care să se poată deduce registrul segment care trebuie utilizat.

Registrele segment utilizate implicit sunt

- pentru registrele de bază:
  - SS pentru BP,
  - DS pentru BX;
- pentru registrele index:
  - DS pentru ambele registre index (DI,SI).

În cazul în care instrucțiunea folosește și un registru de bază și unul index, atunci se va utiliza regula corespunzătoare registrului de bază. Deci, în concluzie, ori de câte ori se utilizează registrul de bază BP, fără nume de variabilă sau prefix segment, se va utiliza pentru adresare segmentul referit de SS; în caz contrar, se va utiliza segmentul referit de DS.

Exemple de adresări:

```
seg_date        segment
        d1      dw      3 dup (3)
        d2      db      5 dup (5)
seg_date        ends
cod     segment
        assume cs:cod, ds:seg_date
        start:  .....
                mov     bx, offset d2
                mov     al, [bx]        ; utilizare implicita a registrului. DS
                mov     bp, sp
                mov     dx, [bp]        ; utilizare implicita a registrului SS
                mov     ax, es:[bp]     ; utilizare explicita a registrului ES
                mov     ax, d1[2]       ; utilizare explicita a registrului DS
                mov     ds:[bp+si], ax  ; explicit DS
                mov     si, offset d1
                mov     [si], dx        ; utilizare implicita a registrului DS
                        .....
cod             ends
end             start
```

Instrucțiunile care lucrează cu stiva (PUSH, POP, CALL, RET, INT, IRET) utilizează întotdeauna SS:SP şi nu pot fi prefixate.

Instrucțiunile pe şiruri utilizează întotdeauna pentru şirul destinație registrul segment ES. Referințele anonime, care nu utilizează registrul segment DS pentru şirul sursă, considerat implicit de asamblor, necesită specificarea explicită nu numai a registrului segment utilizat, dar şi tipul operanzilor:

```
movs    es:byte ptr [di], ss:[si]
movs    word    ptr es:[di], ss:[si]
```