



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

## Programare în limbaj de asamblare

### 53. Comunicarea între module.

## Comunicarea între module

Este posibil însă să dezvoltăm un program care constă dintr-un modul principal, legat (editate legăturile) împreună cu unul sau mai multe programe asamblate separat. Motivele organizării programelor în subprograme sunt următoarele:

- pentru a face legătura între diferite limbaje, pentru a combina ușurința programării în LNI cu eficiența de prelucrare a LA;
- pentru a facilita dezvoltarea de proiecte mari, în care diferite echipe elaborează separat modulele lor;
- pentru a suprapune părți ale unui program pe durata execuției sale, datorită dimensiunii mari a programului.

Programele de dimensiuni mai mari se împart în mai multe module. Vom denumi modul, un program compus din unul sau mai multe segmente, cuprins într-un fișier. Fiecare modul este asamblat separat, generând propriul său modul obiect. Editorul de legături realizează apoi legarea modulelor obiect într-un singur modul executabil. Modulele trebuie să comunice între ele.

În cazul simbolurilor, definite în cadrul unui modul, ce trebuie referite din alte module, acestea trebuie declarate 'PUBLIC', în modulul unde sunt definite, și declarate 'EXTRN' :

BYTE, WORD, DWORD, FWORD, QWORD,

TBYTE - ptr. variabile

ABS - pentru constante,

NEAR, FAR, PROC - pentru etichete sau proceduri; pentru ultimul tip

(PROC) asamblorul determină tipul procedurii.

Deci la definirea simbolurilor se declară 'PUBLIC', iar la referire, din alt modul, se declară 'EXTRN'.

## Directivile PUBLIC, EXTRN, GLOBAL, COMM, INCLUDE

În locul directivelor PUBLIC și EXTRN se poate utiliza directiva GLOBAL; ea se substituie directivelor EXTRN și PUBLIC.

O altă directivă des utilizată este INCLUDE:

### PUSH.ASM

*push ax*

*push bx*

....

*push sp*

*push bp*

*procedura*

*include*

.....

*include*

*ret*

*procedura*

### POP.ASM

*pop bp*

*pop sp*

....

*pop bx*

*pop ax*

*proc*

*push.asm*

*pop.asm*

*endp*

Pentru rezervarea de spațiu de memorie la declararea unor variabile neinițializate (ce fac parte din segmentul BSS), se poate folosi directiva **COMM**:

**COMM [ NEAR/FAR ] identificator : tip\_element [ : dim ]**

Atributul **NEAR/FAR**, dacă lipsește se poate determina din modelul de implementare ales, pentru directive simplificate de segmentare, sau se presupune **NEAR**, pentru directivele complete. Dacă aceeași declarație este în mai multe module, variabila respectivă va fi alocată o singură dată. Utilizând această directivă, nu mai este necesar să se definească variabila ca **PUBLIC** într-un modul și **EXTRN** în celălalt, întrucât atributele asociate de cele două directive sunt furnizate de **.COMM**. Directiva va fi plasată în segmentul de date. Atributul *tip\_element* : **BYTE**, **WORD**, **DWORD**, **QWORD**, și **TBYTE**, sau un număr întreg, care specifică numărul de octeți. Parametrul *dim* indică numărul de elemente pentru variabilă, implicit fiind 1.

De exemplu declarația:

**COMM FAR VARIABILA : BYTE : 25**

definește **VARIABILA** de 25 de octeți.