

LUCRAREA Nr. 1

ASAMBLORUL INTEGRAT TASMB

1. Scopul lucrării

Familiarizarea cu posibilitățile de editare, asamblare și rulare a programelor scrise în limbajul de asamblare al microprocesoarelor Intel funcționând “în modul real”, cu ajutorul asamblorului integrat.

2. Prezentare generală

După lansarea programului TASMB, pe ecranul calculatorului apare meniul principal al asamblorului. În acest meniu sunt afișate:

- ora curentă;
- comenzile asamblorului integrat;
- mărimea memoriei disponibile (inițial 65278 octeți);
- mărimea memoriei utilizate (inițial 0 octeți).

Executarea comenzilor se face prin tastarea literei corespunzătoare, afișate mai intens pe ecran (mod "highlight"). Programul fie execută direct comanda primită, fie răspunde cu un mesaj și așteaptă introducerea unor parametri urmați de tasta **ENTER**. Dacă în locul parametrilor așteptați se introduce doar **ENTER**, comanda nu este executată.

3. Comenzile asamblorului integrat TASMB

3.1. Editare (E)

În acest mod se pot crea/modifica programe sursă scrise în limbajul de asamblare al microprocesorului INTEL 8086, sau orice alte texte ASCII.

Principalele comenzi de editare :

- **←→↑↓** deplasează cursorul cu un caracter sau cu o linie în sensul indicat;
- **PgUp** sau **CTRL/R** deplasează cursorul cu o pagină în sus;
- **PgDn** sau **CTRL/C** deplasează cursorul cu o pagină în jos;
- **Home** sau **CTRL/QS** poziționează cursorul la începutul liniei curente;
- **End** sau **CTRL/QD** poziționează cursorul la sfârșitul liniei curente;
- **BS** (Backspace) șterge un caracter la stânga cursorului;
- **Del** (Delete) șterge caracterul de sub cursor;
- **Ins** (Insert) comută modurile inserare/suprascriere;

- **CTRL/Y** șterge linia curentă;
- **CTRL/T** șterge cuvântul din dreapta cursorului;
- **CTRL/A** poziționează cursorul la începutul cuvântului din stânga sa;
- **CTRL/F** poziționează cursorul la începutul cuvântului din dreapta sa;
- **CTRL/QR** poziționează cursorul pe prima linie a textului;
- **CTRL/QC** poziționează cursorul pe ultima linie a textului;
- **CTRL/KB** marchează începutul unui bloc text;
- **CTRL/KK** marchează sfârșitul unui bloc text;
- **CTRL/KC** copiază blocul text marcat;
- **CTRL/KV** mută blocul text marcat;
- **CTRL/KY** șterge blocul text marcat;
- **CTRL/KH** anulează marcarea blocului;
- **CTRL/KR** citește bloc text dintr-un fișier de pe disc;
- **CTRL/KW** scrie bloc text într-un fișier pe disc.

Ieșirea din modul editare se face cu tasta **ESC** sau **CTRL/KD**, programul revenind în meniul principal.

3.2. Salvare fișier sursă (W)

Permite scrierea pe disc a fișierului aflat în editare. Programul solicită numele sub care va fi salvat fișierul text. Dacă nu se introduce extensia în mod explicit, fișierului i se va atribui implicit extensia **ASM**.

Exemplu: **Write Source File: PROG1[.ASM]**

3.3. Citire fișier sursă (G)

Incarcă un fișier de pe disc, introducându-l în buffer-ul de editare. După citire, se poate intra în modul editare (**E**) pentru a vizualiza sau modifica textul încărcat.

Exemplu: **Read Source File: PROG1[.ASM]**

3.4. Stergere fișier (K)

Permite ștergerea unui fișier aflat pe disc. Numele fișierului trebuie introdus complet (**NUME.EXTENSIE**).

Exemplu: **Kill File : PROG2.ASM**

3.5. Listare fișier text (L)

Listează un fișier pe ecran sau la imprimantă, conform opțiunilor asamblorului (vezi 3.10.1). Numele fișierului trebuie introdus complet. Întreruperea listării se face cu tasta **SPACE**, reluarea făcându-se cu **SPACE** sau **ENTER**. Abandonarea listării se face cu **ESC** sau **CTRL/C**.

Exemplu: **Text File: PROG1.ASM**

3.6. Afișare conținut fișier în hexazecimal (H)

Permite vizualizarea conținutului unui fișier, octet cu octet, atât în valori hexazecimale cât și în caractere ASCII. Comenzile de control al listării sunt aceleași ca și la comanda (**L**).

Exemplu: **Hexdump File: PROG1.ASM**

3.7. Afișarea fișierelor din directorul curent (**D**)

Este echivalentul comenzii **DIR** din sistemul de operare. La introducerea măștii este permisă folosirea simbolurilor "*" și "?". Dacă nu se introduce nici o mască, se vor afișa toate fișierele existente în directorul curent.

Exemplu: **Directory Mask: *.ASM**

3.8. Schimbarea discului/directorului curent (**N**)

Este echivalentă unei comenzi **A:**, **B:** etc. din sistemul de operare. După schimbarea discului curent este posibilă și modificarea directorului curent, echivalentă cu comanda **CHDIR** din sistem. Regulile de parcurgere a arborelui director sunt aceleași ca și în DOS.

Exemplu: **New Drive: F**
 New Directory: \DIR1

3.9. Conversia unei valori numerice (**V**)

Permite conversia dintr-o bază de numerație într-alta. Bazele permise sunt: hexazecimală (**H**), zecimală (implicită sau **D**), octală (**O**), binară (**B**). Se afișează și caracterul ASCII corespunzător valorii introduse, dacă acesta există.

Exemplu: **Value: 66**
 HEX = 00000042, DEC = 66, OCT = 000000000102
 BIN = 0000 0000 0100 0010, CHAR = B

3.10. Asamblarea (**A**)

Prin operația de asamblare se convertește un program sursă în codul obiect corespunzător microprocesorului. În funcție de opțiunile de asamblare selectate, se pot genera fișiere **OBJ**, **COM**, liste de simboluri, etc. Tot în faza de asamblare se detectează și eventualele erori sintactice ale programului sursă.

O linie sursă are în general următoarea sintaxă :

[eticheta[:]] cod operatie [operand[,operand]] [;comentariu]

Se recomandă ca la scrierea programului etichetele să înceapă din prima coloană a liniei iar între etichetă și codul operației, precum și între cod și primul operand, să se folosească separatorul **TAB** pentru a facilita citirea textului sursă.

Directivele de asamblare sunt comenzi date asamblorului care indică modul în care acesta va prelucra textul sursă. Cele mai uzuale directive de asamblare sunt :

- **ORG expresie** - indică adresa de încărcare a programului asamblat, la lansarea acestuia în execuție. Pentru programele de tip **COM**, această adresă trebuie să fie 100H.
- **[eticheta] DB [expresie[,expresie]]** - rezervă unul sau mai mulți octeți în memorie și îi inițializează cu expresiile respective, dacă acestea există.
- **[eticheta] DB 'text'** - rezervă o zonă de memorie și o inițializează cu un text ASCII.
- **[eticheta] DW [expresie[,expresie]]** - rezervă unul sau mai multe cuvinte de câte doi octeți și le inițializează cu expresiile respective, dacă acestea există.
- **simbol EQU expresie** - echivalează simbolul din zona de etichetă cu expresia din zona de operand.
- **END** - indică sfârșitul programului sursă.

3.10.1. Opțiuni de asamblare (O)

Înainte de a trece la asamblarea unui program, este utilă activarea opțiunilor de asamblare dorite, prin apăsarea tastelor funcționale corespunzătoare, după ce s-a intrat în meniul de opțiuni. Dezactivarea se face în același mod. O opțiune este activată cu ON și dezactivată cu OFF. Așa cum se va vedea mai jos, unele opțiuni sunt alternative (nu pot fi activate simultan).

- F1 (Screen)** - conținutul listei de asamblare, listele de simboluri și de referințe încrucișate, precum și efectele comenzilor (**L**) și (**H**) se afișează pe ecran. Inițial ON.
- F2 (Printer)** - toate efectele comenzilor de mai sus se trimit la imprimant. Inițial OFF. Opțiunile **F1** și **F2** sunt alternative.
- F3 (Symbols)** - generează lista simbolurilor folosite, afișând pentru fiecare simbol numele, tipul, valoarea, linia în care a fost definit, lungimea. Inițial OFF.
- F4 (Xrefer.)** - generează lista referințelor încrucișate, afișând numele simbolurilor și liniile de program în care acestea au fost definite și referite. Inițial OFF.
- F5 (Memory)** - asamblarea programului se face în memorie, permițând rularea acestuia în cadrul asamblorului integrat. Adresa implicită de asamblare este 8000H, nefiind permisă folosirea unei directive **ORG** cu altă adresă. Inițial OFF.
- F6 (ErrWait)** - asamblarea se oprește la prima eroare întâlnită. Cu **ENTER** se continuă asamblarea, cu orice altă tastă se intră în modul editare, permițându-se corectarea erorii. Inițial OFF.
- F7 (OBJ File)** - programul sursă este asamblat în modul obiect (cu extensia **OBJ**), urmând ca acesta să fie supus unei operații de editare a legăturilor (link-editare), eventual împreună cu alte module obiect, pentru a fi transformat în program executabil (cu extensia **EXE** sau **COM**). La asamblare, fișierul obiect este scris pe disc, cerându-se confirmarea pentru numele acestuia. Inițial OFF.

F8 (COM File) - programul sursă este asamblat în program direct executabil, de tip COM, asamblorul integrat executând și operațiunea de editare a legăturilor. Inițial OFF. Opțiunile **F5**, **F7** și **F8** sunt alternative.

F9 (LST File) - asamblorul generează un fișier listing, în care introduce lista de asamblare conținând numărul liniei sursă, codul instrucțiunii asamblate și textul liniei. Se introduc de asemenea lista de simboluri și de referințe încrucișate, dacă la asamblare opțiunile respective au fost selectate. Inițial OFF.

Ieșirea din meniul de opțiuni și revenirea în meniul principal se face prin apăsarea tastei **ENTER**.

Asamblarea se face în două treceri, la a doua trecere generându-se lista de asamblare. Defilarea textului pe ecran poate fi oprită cu **SPACE**, reluarea făcându-se cu **SPACE** sau cu **ENTER**. Abandonarea asamblării se poate face cu **ESC** sau **CTRL/C**.

După terminarea operațiunii de asamblare, programul afișează numărul de linii conținute în programul sursă, numărul de erori întâlnite la asamblare, numărul de octeți folosiți pentru codul obiect, numărul de octeți rămași liberi și timpul de asamblare.

3.11. Afișarea listei de simboluri (S)

Permite vizualizarea simbolurilor utilizate în program. Afișează de asemenea numărul total de simboluri folosite și spațiul de memorie alocat.

3.12. Afișarea listei de referințe încrucișate (X)

Este similară comenzii (S), afișând lista de referințe dacă această opțiune a fost activată la asamblare. În caz contrar, se afișează numărul de simboluri utilizate și spațiul de memorie alocat.

3.13. Rularea unui program asamblat în memorie (R)

Permite lansarea în execuție a unui program care a fost asamblat cu opțiunea **F5** activată. După rulare se afișează registrele microprocesorului așa cum sunt la ieșirea din program.

OBSERVAȚIE: pentru ca, după rulare programului, să fie posibilă întoarcerea corectă în asamblorul integrat, este necesar ca programul utilizatorului să se încheie cu instrucțiunea **INT 20H**.

3.14 Ieșirea din asamblorul integrat (Q)

Cu această comandă se părăsește programul și se revine în sistemul de operare. Dacă textul existent în buffer-ul de editare nu a fost salvat pe disc, se cere confirmarea salvării acestuia.

4. Modul de lucru recomandat

- Se lansează programul TASMB.
- Se intră în modul editare (**E**) și se scrie programul dorit folosindu-se comenzile de editare; în cazul în care programul sursă a fost generat anterior și se află pe disc, acesta se încarcă de pe disc (**G**).
- Se activează opțiunile de asamblare dorite (**O**).
- Se assemblează programul (**A**).
- Dacă s-au întâlnit erori la asamblare, se revine la editare și se corectează erorile.
- Programul sursă se salvează pe disc (**W**).
- După asamblarea corectă a programului sursă se poate încerca rularea acestuia, fie în memorie (**R**), fie de pe disc.
- În cazul în care rularea programului nu dă rezultatele scontate, se încearcă modificarea acestuia, revenindu-se la editare, sau se folosește un program de depanare ("debugger").

5. Desfășurarea lucrării

- 5.1. Se editează (**E**) textul sursă din Anexa 1.
- 5.2. Se salvează (**W**) textul pe disc cu numele **P1.ASM**.
- 5.3. Se reîncarcă (**G**) textul de pe disc.
- 5.4. Se verifică comenzile (**D**), (**H**), (**L**), (**V**).
- 5.5. Se activează (**O**) opțiunile **F1**, **F3**, **F4**, **F5**, **F6**.
- 5.6. Se assemblează (**A**) programul sursă.
- 5.7. Se verifică comenzile (**S**), (**X**).
- 5.8. Se rulează (**R**) programul asamblat în memorie.

- 5.9. Se introduce (**E**) deasupra primei linii a programului sursă, o nouă linie conținând directiva **ORG 100H**.
- 5.10. Se selectează (**O**) opțiunea **F8**.
- 5.11. Se assemblează (**A**) din nou programul sursă.
- 5.12. Se părăsește (**Q**) asamblorul integrat.
- 5.13. Se rulează programul asamblat pe disc (**P1.COM**).

ANEXA

Programul sursă P1.ASM

```

start    mov  ax,cs
         mov  ds,ax
         mov  ax,0003
         int  10h           ; sterge ecran
         mov  ax,1818h
         mov  pozit,ax
         mov  ah,02
         mov  bh,00
         mov  dx,pozit
         int  10h           ; pozitioneaza cursor
         mov  bx,offset mesaj
         mov  cx,lmes-mesaj
adr1     push bx
         push cx
         mov  al,[bx]
         mov  bx,0038h
         mov  cx,01
         mov  ah,09
         int  10h           ; scrie un caracter
         call delay
         inc  word ptr pozit
         mov  ah,02
         mov  bh,00
         mov  dx,pozit
         int  10h           ; avanseaza cursor
         pop  cx
         pop  bx
         inc  bx
         loop adr1
         mov  bx,offset crlf
         mov  cx,length crlf
         shl  cx,1
adr2     push bx
         push cx
         mov  al,[bx]
         mov  ah,0eh
         int  10h           ; avanseaza linie
         call delay
         pop  cx
         pop  bx
         inc  bx
         loop adr2
         int  20h

```

```
delay    proc
          mov  cx,0ffffh
repl     nop
          loop repl
          ret
delay    endp

pozit    dw   0000
mesaj    db   ' ACESTA ESTE UN PROGRAM DE TEST '
lmes     equ  $
crlf     db   25 dup (10,13)
end
```