

LUCRAREA Nr. 2

INSTRUCȚIUNI DE TRANSFER DE DATE ȘI OPERAȚII CU ȘIRURI PENTRU MICROPROCESOARELE COMPATIBILE INTEL x86 (IA-32) ÎN MODUL REAL

1. Scopul lucrării

Scopul prezentei lucrări este familiarizarea cu modurile de adresare, cu instrucțiunile de transfer de date elementare și structurate specifice microprocesoarelor compatibile Intel (IA-32) funcționând „în modul real” (particularizând pentru Intel 8086), precum și prezentarea unor directive de asamblare.

2. Memoriu de instrucțiuni

Convenții:

s:	sursă;
d:	destinație;
AL AX:	acumulatorul implicit de 8 sau de 16 biți;
mem:	conținutul unei locații de memorie sau conținutul a două locații de memorie succesive, adresate cu unul dintre modurile de adresare permise pentru memoria de date, cu excepția adresării imediate;
mem16:	conținutul a două locații de memorie succesive adresate cu unul dintre modurile de adresare permise pentru memoria de date, cu excepția adresării imediate;
mem32:	conținutul a patru locații de memorie succesive adresate cu unul dintre modurile de adresare permise pentru memoria de date, cu excepția adresării imediate;
r r_i r_j:	un registru oarecare de 8 sau de 16 biți, exceptând registrele segment;
r8:	un registru de 8 biți;
r16:	un registru de 16 biți, exceptând registrele segment;
rs:	un registru segment (CS, SS, DS, ES);
data:	un operand de 8 sau 16 biți care face parte din formatul instrucțiunii (adresare imediată);
data8:	un operand de 8 biți care face parte din formatul instrucțiunii (adresare imediată);

data16:	un operand de 16 biți care face parte din formatul instrucțiunii (adresare imediată);
disp8:	deplasament pe 8 biți (face parte din formatul instrucțiunii);
disp16:	deplasament pe 16 biți (face parte din formatul instrucțiunii);
adr:	o adresă completă (pe 16 biți);
adr8:	o adresă scurtă (pe 8 biți);
adr32:	o adresă logică exprimată pe patru octeți succesivi;
port:	adresa (numărul de ordine) unui port de intrare/ieșire, pe 8 biți;
AE:	adresa efectivă.

Pentru fanioane:

x :	fanionul se schimbă în conformitate cu rezultatul operațiunii;
1 :	fanionul este setat necondiționat;
0 :	fanionul este resetat necondiționat;
? :	fanionul este afectat impredictibil;
blanc:	fanionul nu este afectat.

Pentru calculul numărului de stări :

cAE - timpul de calcul al adresei efective, și anume	
- adresare directă: AE=disp8 disp16	6 stări;
- adresare indexată: AE=(SI) (DI)+ disp8 disp16	9 stări;
- adresare indirectă implicită: AE=(SI) (DI)	5 stări;
- adresare relativă la bază directă, fără deplasament: AE=(BX)	5 stări;
- adresare relativă la bază directă, cu deplasament: AE=(BX)+disp8 disp16	9 stări;
- adresare relativă la bază indexată: AE=(BX)+(SI) (DI)+disp8 disp16	12 stări;
- adresare relativă la bază implicită: AE=(BX)+(SI) (DI)	8 stări;
- adresare în stivă directă, fără deplasament: AE=(BP)	5 stări;
- adresare în stivă directă, cu deplasament: AE=(BP)+disp8 disp16	9 stări;
- adresare în stivă indexată: AE=(BP)+(SI) (DI)+disp8 disp16	12 stări;
- adresare în stivă implicită: AE=(BP)+(SI) (DI)	8 stări;
- pentru redirecționarea segmentului se mai adaugă 2 stări.	

2.1. Instrucțiuni de transfer de date

MOV d,s	Copiază sursa	OF DF IF TF SF ZF AF PF CF
	la destinație	

Descrierea formală a semanticii, în general: $(d) \leftarrow (s)$.

Operanzi	Exemple	Descrierea formală a semanticii
r, data	MOV BX,ALFA	$(BX) \leftarrow ALFA$
mem, data	MOV [BP+DI],55H	$((SS) \uparrow 0H + (BP) + (DI)) \leftarrow 55H$
AL AX, mem	MOV AX,[SI]	$(AL) \leftarrow ((DS) \uparrow 0H + (SI)),$ $(AH) \leftarrow ((DS) \uparrow 0H + (SI) + 1)$
mem, AL AX	MOV [BX+SI+10H],AL	$((DS) \uparrow 0H + (BX) + (SI) + 10H) \leftarrow (AL)$
r1, r2	MOV AX,SP	$(AX) \leftarrow (SP)$
r, mem	MOV BH,[BX+1000H]	$(BH) \leftarrow ((DS) \uparrow 0H + (BX) + 1000H)$
mem, r	MOV [2000H],DL	$((DS) \uparrow 0H + 2000H) \leftarrow (DL)$
rs*, r16	MOV ES,CX	$(ES) \leftarrow (CX)$
rs*, mem16	MOV SS,[DI]	$(SS) \leftarrow ((DS) \uparrow 0H + (DI) + 1) \uparrow ((DS) \uparrow 0H + (DI))$
r16, rs	MOV CX,CS	$(CX) \leftarrow (CS)$
mem16, rs	MOV [SI],DS	$((DS) \uparrow 0H + (SI) + 1) \uparrow ((DS) \uparrow 0H + (SI)) \leftarrow (DS)$

* registrul segment nu poate fi CS

PUSH s	Transferă în stivă de la sursă	OF DF IF TF SF ZF AF PF CF

Descrierea formală a semanticii, în general:

$$(SP) \leftarrow (SP) - 2$$

$$((SS) \uparrow 0H + (SP) + 1) \leftarrow (s)_h$$

$$((SS) \uparrow 0H + (SP)) \leftarrow (s)_l$$

Operanzi	Exemple	Descrierea formală a semanticii
r16	PUSH BP	$(SP) \leftarrow (SP) - 2$ $((SS) \uparrow 0H + (SP) + 1) \uparrow ((SS) \uparrow 0H + (SP)) \leftarrow (BP)$
mem16	PUSH [DI+100H]	$(SP) \leftarrow (SP) - 2$ $((SS) \uparrow 0H + (SP) + 1) \leftarrow ((DS) \uparrow 0H + (DI) + 101H)$ $((SS) \uparrow 0H + (SP)) \leftarrow ((DS) \uparrow 0H + (DI) + 100H)$
rs	PUSH SS	$(SP) \leftarrow (SP) - 2$ $((SS) \uparrow 0H + (SP) + 1) \uparrow ((SS) \uparrow 0H + (SP)) \leftarrow (SS)$

PUSHF	Transferă în stivă registrul de fanioane	OF DF IF TF SF ZF AF PF CF

Descrierea formală a semanticii:

$$(SP) \leftarrow (SP) - 2$$

$$((SS) \uparrow 0H + (SP) + 1) \leftarrow (F)_h$$

$$((SS) \uparrow 0H + (SP)) \leftarrow (F)_l$$

POP d	Transferă din stivă la destinație	OF DF IF TF SF ZF AF PF CF

Descrierea formală a semanticii, în general:

$$(d)_l \leftarrow ((SS) \uparrow 0H + (SP))$$

$$(d)_h \leftarrow ((SS) \uparrow 0H + (SP) + 1)$$

$$(SP) \leftarrow (SP) + 2$$

Operanzi	Exemple	Descrierea formală a semanticii
r16	POP CX	$(CL) \leftarrow ((SS) \uparrow 0H + (SP))$ $(CH) \leftarrow ((SS) \uparrow 0H + (SP) + 1)$ $(SP) \leftarrow (SP) + 2$
mem16	POP [F0F0H]	$((DS) \uparrow 0H + F0F0H) \leftarrow ((SS) \uparrow 0H + (SP))$ $((DS) \uparrow 0H + F0F1H) \leftarrow ((SS) \uparrow 0H + (SP) + 1)$ $(SP) \leftarrow (SP) + 2$
rs (nu CS)	POP SS	$(SS) \leftarrow ((SS) \uparrow 0H + (SP) + 1) \uparrow ((SS) \uparrow 0H + (SP))$ $(SP) \leftarrow (SP) + 2$

POPF	Transferă din stivă registrul de fanioane	OF DF IF TF SF ZF AF PF CF
		x x x x x x x x x

Descrierea formală a semanticii:

$$(F)_l \leftarrow ((SS) \uparrow 0H + (SP))$$

$$(F)_h \leftarrow ((SS) \uparrow 0H + (SP) + 1)$$

$$(SP) \leftarrow (SP) + 2$$

XCHG d,s	Transferă sursa la destinație și destinația la sursă	OF DF IF TF SF ZF AF PF CF

Descrierea formală a semanticii, în general: $(d) \leftrightarrow (s)$

Operanzi	Exemple	Descrierea formală a semanticii
r16	XCHG DX XCHG AX este NOP	$(AX) \leftrightarrow (DX)$ $(AX) \leftrightarrow (AX) \equiv \text{NOP}$
r1, r2	XCHG CH,CL	$(CH) \leftrightarrow (CL)$
r, mem	XCHG BX, [BX+DI]	$(BL) \leftrightarrow ((DS) \uparrow 0H + (BX) + (DI))$ $(BH) \leftrightarrow ((DS) \uparrow 0H + (BX) + (DI) + 1)$

XLAT	Translatează	OF DF IF TF SF ZF AF PF CF

Descrierea formală a semanticii: $(AL) \leftarrow ((DS) \uparrow 0H + (BX) + (AL))$

LAHF	Încarcă(AH)cu octetul inferior al registrului F	OF DF IF TF SF ZF AF PF CF
-------------	---	----------------------------

Descrierea formală a semanticii: $(AH) \leftarrow (F)_l$

SAHF	Încarcă octetul inferior al registrului F cu (AH)	OF DF IF TF SF ZF AF PF CF x x x x x
-------------	---	---

Descrierea formală a semanticii: $(F)_l \leftarrow (AH)$

IN d,s	Transferul unui octet sau cuvânt de la un port în acumulator	OF DF IF TF SF ZF AF PF CF
---------------	--	----------------------------

Descrierea formală a semanticii, în general: $(AL) \mid (AX) \leftarrow (s)$.

Operanzi	Exemple	Descrierea formală a semanticii
AL AX, port	IN AL, 0FH	$(AL) \leftarrow (0FH)$
AL AX, DX	IN AX, DX	$(AX) \leftarrow ((DX) + 1) \uparrow ((DX))$

OUT d,s	Transferul unui octet sau cuvânt din acumulator la un port	OF DF IF TF SF ZF AF PF CF
----------------	--	----------------------------

Descrierea formală a semanticii, în general: $(d) \leftarrow (AL) \mid (AX)$.

Operanzi	Exemple	Descrierea formală a semanticii
port, AL AX	OUT 10H, AL	$(10H) \leftarrow (AL)$
DX, AL AX	OUT DX, AX	$((DX) + 1) \uparrow ((DX)) \leftarrow (AX)$

LDS d,s	Încarcă un registru de 16 biți și reg. segment de date	OF DF IF TF SF ZF AF PF CF
----------------	--	----------------------------

Descrierea formală a semanticii, în general: $(r16) \leftarrow (mem32)_l$
 $(DS) \leftarrow (mem32)_h$

Operanzi	Exemple	Descrierea formală a semanticii
r16, mem32	LDS SI, [10H]	$(SI) \leftarrow ((DS) \uparrow 0H + 11H) \uparrow ((DS) \uparrow 0H + 10H)$ $(DS) \leftarrow ((DS) \uparrow 0H + 13H) \uparrow ((DS) \uparrow 0H + 12H)$

LES d,s	Încarcă un registru de 16 biți și registrul ES	OF DF IF TF SF ZF AF PF CF

Descrierea formală a semanticii, în general: $(r16) \leftarrow (mem32)_l$
 $(ES) \leftarrow (mem32)_h$

Operanzi	Exemple	Descrierea formală a semanticii
r16, mem32	LES DI, [DI]	$(DI) \leftarrow ((DS) \uparrow 0H + (DI) + 1) \uparrow ((DS) \uparrow 0H + (DI))$ $(ES) \leftarrow ((DS) \uparrow 0H + (DI) + 3) \uparrow ((DS) \uparrow 0H + (DI) + 2)$

LEA d,s	Încarcă un registru de 16 biți cu o adresă efectivă	OF DF IF TF SF ZF AF PF CF

Descrierea formală a semanticii, în general: $(r16) \leftarrow AE$ a locației mem

Operanzi	Exemple	Descrierea formală a semanticii
r16, mem16	LEA BX, [BX+DI+10H]	$(BX) \leftarrow (BX) + (DI) + 10H$ adresa fizică: $AF = (DS) \uparrow 0H + (BX) + (DI) + 10H$ adresa efectivă: $AE = (BX) + (DI) + 10H$

Notă: Singurele instrucțiuni care afectează fanioanele sunt **POPF** și **LAHF**.

2.2. Operații de transferuri de șiruri

Pentru aceste instrucțiuni sursa și destinația se găsesc implicit astfel:

sursa - în segmentul format cu **DS**, cu adresa efectivă **SI**;

destinația - în segmentul format cu **ES**, cu adresa efectivă **DI**.

Operațiile cu șiruri permit următoarele prefixe de repetabilitate:

REP	Repetă necondiționat primitiva de operație cu șiruri pe care o precede	OF DF IF TF SF ZF AF PF CF

Operanzi		Nr.de stări	Octeți	Exemple							
		9	1	REP MOVSB							
REPE REPZ	Repetă cât timp "egal" cât timp "zero" primitiva CMPS sau SCAS			OF DF IF TF SF ZF AF PF CF							
Operanzi		Nr.de stări	Octeți	Exemple							
		9	1	REPE CMPSB							
REPNE REPNZ	Repetă cât timp "ne-egal" cât timp "non-zero" primiti-va CMPS sau SCAS			OF DF IF TF SF ZF AF PF CF							
Operanzi		Nr.de stări	Octeți	Exemple							
		9	1	REPNE SCASW							

Primitivele operațiilor de transferuri de șiruri sunt:

MOVS	Transferă o compo- nentă a șirului sursă în șirul destinație		OF DF IF TF SF ZF AF PF CF							
Operanzi	Nr.de stări	Octeți	Exemple							
	18	1	MOVSB ; pe octeți							
	18	1	MOVSW ; pe cuvinte							

Obs.: Primitiva MOVS poate fi însoțită de prefixul de repetabilitate **REP**; acesta adaugă 9 stări.

Descrierea formală a semanticii:

$$((ES) \uparrow 0H + (DI)) \leftarrow ((DS) \uparrow 0H + (SI))$$

$$(((ES) \uparrow 0H + (DI) + 1) \leftarrow ((DS) \uparrow 0H + (SI) + 1))$$

if (DF) = 0 then

$$(DI) \leftarrow (DI) + N$$

$$(SI) \leftarrow (SI) + N$$

else

$$(DI) \leftarrow (DI) - N$$

$$(SI) \leftarrow (SI) - N$$

cu **N=1** pentru octet
și **N=2** pentru cuvânt.

LODS	Încarcă componen- tele unui șir în acumulator		OF DF IF TF SF ZF AF PF CF
Operanzi	Nr.de stări	Octeți	Exemple
	12	1	LODSB ; pe octeți
	12	1	LODSW ; pe cuvinte

Descrierea formală a semanticii:

$$(\text{AL}) \leftarrow ((\text{DS})\uparrow 0\text{H} + (\text{SI}))$$

$$| (\text{AX}) \leftarrow ((\text{DS})\uparrow 0\text{H} + (\text{SI}) + 1) \uparrow ((\text{DS})\uparrow 0\text{H} + (\text{SI}))$$

if $(\text{DF}) = 0$ then

$$(\text{SI}) \leftarrow (\text{SI}) + \text{N}$$

else

$$(\text{SI}) \leftarrow (\text{SI}) - \text{N}$$

cu $\text{N}=1$ pentru octet

și $\text{N}=2$ pentru cuvânt.

STOS	Încarcă component-le unui șir din acumulator	OF DF IF TF SF ZF AF PF CF							
Operanzi	Nr.de stări	Octeți	Exemple						
	11	1	STOSB ; pe octeți						
	11	1	STOSW ; pe cuvinte						

Obs.: Primitiva STOS poate fi însoțită de prefixul de repetabilitate **REP**; acesta adaugă 9 stări.

Descrierea formală a semanticii:

$$((\text{ES})\uparrow 0\text{H} + (\text{DI})) \leftarrow (\text{AL})$$

$$| ((\text{ES})\uparrow 0\text{H} + (\text{DI}) + 1) \uparrow ((\text{ES})\uparrow 0\text{H} + (\text{DI})) \leftarrow (\text{AX})$$

if $(\text{DF}) = 0$ then

$$(\text{DI}) \leftarrow (\text{DI}) + \text{N}$$

else

$$(\text{DI}) \leftarrow (\text{DI}) - \text{N}$$

cu $\text{N}=1$ pentru octet

și $\text{N}=2$ pentru cuvânt.

3. Directive de asamblare și cuvinte rezervate în TASMB

- **ORG** “ASSIGN LOCATION COUNTER” DIRECTIVĂ
 Sintaxa: **ORG** **adr.**
 Efect: instrucțiunea următoare va fi plasată la adresa **adr.**
- **DB** “DEFINE BYTE” DIRECTIVĂ
 Sintaxa: **[simbol] DB data,[data],[data,.....].**
 Efect: rezervă spațiu, sub forma de byte, în memorie pentru date. **Simbol**, dacă există, va avea ca valoare adresa la care a fost plasat primul byte.
- **DW** “DEFINE WORD” DIRECTIVĂ
 Sintaxa: **[simbol] DW data,[data],[data,....].**
 Efect: rezervă spațiu, sub formă de word, în memorie pentru date. **Simbol**, dacă există, va avea ca valoare adresa la care a fost plasat primul word.
- **EQU** “CREATE SYMBOL” DIRECTIVĂ
 Sintaxa: **simbol EQU expresie**
 Efect: asamblorul atribuie lui **simbol** valoarea obținută prin evaluarea **expresiei**. **Expresie** poate fi o **data**, sau o combinație de **data** prin semnele + și -. Odată folosit **EQU** pentru un simbol, acest simbol nu mai poate fi modificat în cursul programului.
- **OFFSET** “OFFSET OF EXPRESSION” OPERATOR
 Sintaxa: **OFFSET simbol.**
 Efect: obține adresa pentru **simbol**.
- **BYTE PTR** “CHANGE TYPE OF VARIABLE” OPERATOR
 Sintaxa: **BYTE PTR simbol**
 Efect: convertește **simbol** în tip byte.
- **WORD PTR** “CHANGE TYPE OF VARIABLE” OPERATOR
 Sintaxa: **WORD PTR simbol**
 Efect: convertește **simbol** în tip word.

4. Mod de lucru recomandat

Pentru exersarea instrucțiunilor de transfer se propun în anexă trei programe, astfel:

- **Programul 1** reprezintă o serie de instrucțiuni **MOV** care transferă date între registre și între registre și memorie.

- **Programul 2** realizează salvarea unui registru și a unei locații de memorie în stivă, interschimbul datelor și apoi refacerea registrului și a locației de memorie. După aceasta, programul setează, conform acumulatorului, indicatorii de condiții și îi modifică individual.

- **Programul 3** inițializează zone de memorie cu texte, pe care le va copia în alte zone de memorie prin intermediul instrucțiunilor de transferuri de șiruri.

Se recomandă editarea și asamblarea fiecărui program propus și apoi, cu ajutorul utilitarului AFD, execuția instrucțiune cu instrucțiune urmărindu-se modificarea registrelor și a memoriei.

Observație: cifrele care apar la începutul fiecărei linii nu fac parte din program, fiind niște numere ghid pentru urmărirea programului.

5. Desfășurarea lucrării

5.1. Se lansează turbo-asamblorul TASMB și se editează textul din Anexa 1 sub numele **prog1.asm**.

5.2. Se assemblează programul pe disc (cu opțiunea de asamblare **F8-com file**), obținându-se fișierul **prog1.com**.

5.3 Se obține lista simbolurilor (comanda **S**) și se notează adresele acestora.

5.4. Se părăsește turbo-asamblorul.

5.5. Se lansează debugger-ul AFD și se încarcă fișierul **prog1.com** (cu comanda **L prog1.com**).

5.6. Se execută programul pas cu pas (prin folosirea succesivă a comenzii **F2**) și se urmărește evoluția locațiilor de memorie sau registrelor implicate, conform indicațiilor din Anexa 1.

5.7. Se repetă punctele 5.1. - 5.6. pentru programele 2 și 3 (**prog2** și **prog3**) din Anexele 2 și 3.

ANEXA 1

Programul 1

```

1      org      100h
2      mov      bx,cs           ;seteaza segmentul de
3      mov      ds,bx         ;date egal cu segmentul de cod
4      mov      ax,const       ;immediat in acumulator
5      mov      dx,ax          ;din acumulator in
                                ;registrul DX
6      mov      var,89abh      ;immediat in memorie
7      mov      ax,var         ;din memorie in acumulator
8      mov      ax,[var]       ;adresare directa
9      mov      ax,140h        ;immediat in acumulator
10     mov      ax,[140h]      ;din memorie in acumulator
11     mov      byte ptr var,ah ;din acumulator in memorie -
                                ;- adresare directa
12     mov      bx,offset var   ;incarca in BX adresa lui var
13     mov      bp,bx          ;incarca BP cu BX -
                                ;- adresare tip registru
14     mov      si,4           ;immediat in registrul index SI
15     mov      al,[bx+si-2]    ;din memorie in acumulator -
                                ;- adresare relativa la baza,
                                ;indexata
16     mov      di,2           ;immediat in registrul index DI
17     mov      [bp+di],dl      ;din registrul DH in memorie -
                                ;- adresare in stiva implicita
18     mov      [bp][si],al     ;din registrul AL in memorie -
                                ;- adresare in stiva implicita
19     mov      cuv,11h        ;incarca in locatia de memorie
                                ;cu adresa "cuv" octetul 011h -
                                ;-adresare imediata
20     mov      word ptr cuv,3456h;incarca in locatiile de memorie
                                ;cu adresele "cuv" si "cuv"+1,
                                ;octetii 034h si 056h -
                                ;- adresare imediata
21     int      20h

22 var      dw      0
23 const     equ     0abcdh
24 cuv       db      0

```

Elemente de observat pentru fiecare dintre următoarele instrucțiuni (în AFD se recomandă fixarea **zonei doi** de afișare a memoriei la adresa **DS:13F**):

- 2-3 Vezi **CS**, **BX** și **DS**.
- 4 Vezi **AX** și **DX**.
- 6 Vezi memoria la adresa lui **"var"**.
- 7-10 Vezi **AX** și codul instrucțiunilor.
- 11 Vezi memoria la adresa lui **"var"**.

12-13	Vezi BX și BP .
14	Vezi SI .
15	Vezi memoria la adresa BX+SI-2 și AL .
16	Vezi DI .
17-20	Vezi DL , AL , DX și memoria.

ANEXA 2

Programul 2

```

1      org      100h
2      mov     bx,cs           ;seteaza segmentul de date
3      mov     ds,bx          ;egal cu segmentul de cod
4      mov     temp,0aabbh    ;pune in locatiile de memorie
                                ;cu adresa "temp" si "temp"+1
                                ;cuvantul 0aabbh
5      mov     dx,0ccddh      ;pune in registrul DX
                                ;cuvantul 0ccddh
6      push    dx              ;pune in stiva continutul
7      push    temp            ;lui DX si al locatiilor de
                                ;memorie cu adresa "temp"
                                ;si "temp"+1
8      xchg    dx,temp         ;interschimba continutul lui DX
                                ;cu cel al locatiilor de memorie
9      pop     temp            ;reface din stiva continutul lui
                                ;DX si al locatiilor de memorie
10     pop     dx              ;cu adresa "temp" si "temp"+1
11     mov     ah,11010101b    ;incarca acumulatorul cu 0d5h
12     sahf                    ;incarca fanioanele cu
                                ;continutul lui AH
13     clc                    ;se actioneaza asupra
14     stc                    ;fanionului CF:
15     cmc                    ;clear, set, complement
16     lahf                    ;incarca in AH fanioanele
17     int     20h
18 temp    dw     ?

```

Elemente de observat pentru fiecare dintre următoarele instrucțiuni (în AFD se recomandă fixarea **zonei doi** de afișare a memoriei la adresa **DS:124**):

4-5	Vezi DX și memoria la adresa temp .
6-7	Vezi SP și stiva.
8	Vezi DX și memoria la adresa temp .
9-10	Vezi SP , DX , stiva și memoria la adresa temp .
11-12	Vezi AL și fanioanele.
13-15	Vezi fanionul de transport CF .
16	Vezi fanioanele și AL .

ANEXA 3

Programul 3

```

1      org    100h
2      mov    bx,cs           ;seteaza segmentul de
3      mov    ds,bx           ;date egal cu segmentul de cod
4      mov    bx,6000h        ;seteaza segmentul ES la 6000h
5      mov    es,bx
6      mov    si,offset sursa  ;incarca in SI, DI adresele de
7      mov    di,offset dest   ;inceput ale sirurilor sursa,
                                ;respectiv destinatie
8      lodsb                  ;incarca in AL continutul
                                ;locatiei cu adresa DS:SI
9      stosb                  ;incarca in memorie la adresa
                                ;ES:DI continutul lui AL
10     movsb                  ;transfera byte din sursa
11     movsb                  ;in destinatie de doua ori
12     mov     di,5+offset dest ;incarca in registrul index DI
                                ;adresa elementului 5
                                ;al sirului destinatie
13     mov     si,offset sursa+5 ;incarca in registrul index SI
                                ;adresa elementului 5
                                ;al sirului sursa
14     std                      ;seteaza fanionul de directie DF
15     movsw                  ;transfera word din sursa
16     movsw                  ;in destinatie de doua ori
17     cld                    ;sterge fanionul de directie DF
18     lea     si,sursal        ;incarca in SI si DI adresa
19     lea     di,dest          ;efectiva a simbolurilor
                                ; "sursal" si "dest"
20     mov     cx,dest-sursal   ;incarca in CX lungimea
                                ;sirului "sursal"
21     repnz movsb              ;transfera "sursal" in dest
22     int     20h
sursa db    'exemplu          '
                                ;zona de definire date.
sursal db    'test string'
dest    db    10 dup(?)

```

Elemente de observat pentru fiecare dintre următoarele instrucțiuni:

4-5 Vezi **BX** și **ES**.

6-7 Vezi **SI** și **DI**.

8-9 Vezi **AL**, **DF**, **SI**, **DI**, memoria la adresele **sursa** și **dest**.

Pentru a vizualiza șirul sursă, se setează **zona 1** de afișare a memoriei în segmentul format cu **DS** și având adresa efectivă dată de adresa simbolului **sursa**. Pentru a vizualiza șirul destinație se setează **zona 2** de afișare a memoriei în segmentul format cu **ES** și având adresa efectivă dată de adresa simbolului **dest**.

10-11 Vezi **DF, SI, DI**, memoria.

12-13 Vezi **SI, DI**.

14 Vezi **DF**.

15-16 Vezi **DF, SI, DI**, memoria.

17 Vezi **DF**.

18-20 Vezi **SI, DI, CX**.

21 Vezi **DI, SI**, memoria.