ARHITECTURA SISTEMELOR DE CALCUL
– Seminar 4 –

OPERAȚII PE ȘIRURI DE OCTEȚI/CUVINTE/DUBLUCUVINTE

Instrucțiunile pe șiruri nu au operanzi.

Pregătirea execuției instrucțiunilor pe șiruri este obligatorie și constă în:

- stabilirea valorii DF (Direction Flag) utilizând instrucțiunile: **CLD** (DF = 0) sau **STD** (DF = 1);
- încărcarea în registrul ESI (Source Index) a offset-ului șirului sursă;
- încărcarea în registrul EDI (Destination Index) a offset-ului șirului destinație.

a. Instrucțiuni pentru transferul datelor

LODS (**LO**a**D** **S**tring)
STOS (**STO**re **S**tring)
MOVS (**MOV**e **S**tring)
} urmate de o literă care indică dimensiunea operației {
B (byte)
W (word)
D (doubleword)

LODSB	În AL se încarcă octetul de la adresa <DS:ESI> Dacă DF=0 atunci inc ESI, altfel dec ESI
LODSW	În AX se încarcă cuvântul de la adresa <DS:ESI> Dacă DF=0 atunci ESI=ESI+2, altfel ESI=ESI-2
LODSD	În EAX se încarcă dublucuvântul de la adresa <DS:ESI> Dacă DF=0 atunci ESI=ESI+4, altfel ESI=ESI-4
STOSB	La adresa <ES:EDI> se încarcă octetul din AL Dacă DF=0 atunci inc EDI, altfel dec EDI
STOSW	La adresa <ES:EDI> se încarcă cuvântul din AX Dacă DF=0 atunci EDI=EDI+2, altfel EDI=EDI-2
STOSD	La adresa <ES:EDI> se încarcă dublucuvântul din EAX Dacă DF=0 atunci EDI=EDI+4, altfel EDI=EDI-4
MOVSB	La adresa <ES:EDI> se încarcă octetul de la adresa <DS:ESI> Dacă DF=0 atunci inc ESI și inc EDI, altfel dec ESI și dec EDI
MOVSW	La adresa <ES:EDI> se încarcă cuvântul de la adresa <DS:ESI> Dacă DF=0 atunci ESI=ESI+2 și EDI=EDI+2, altfel ESI=ESI-2 și EDI=EDI-2
MOVSD	La adresa <ES:EDI> se încarcă dublucuvântul de la adresa <DS:ESI> Dacă DF=0 atunci ESI=ESI+4 și EDI=EDI+4, altfel ESI=ESI-4 și EDI=EDI-4

b. Instrucțiuni pentru consultarea/compararea datelor

SCAS (**SCA**n **S**tring)
CMPS (**C**o**MP**are **S**tring)
} urmate de o literă care indică dimensiunea operației {
B (byte)
W (word)
D (doubleword)

SCASB	CMP AL, <ES:EDI> Dacă DF=0 atunci inc EDI, altfel dec EDI
SCASW	CMP AX, <ES:EDI> Dacă DF=0 atunci EDI=EDI+2, altfel EDI=EDI-2

SCASD	CMP EAX, <ES:EDI> Dacă DF=0 atunci EDI=EDI+4, altfel EDI=EDI-4
CMPSB	CMP <DS:ESI>, <ES:EDI> Dacă DF=0 atunci inc ESI și inc EDI, altfel dec ESI și dec EDI
CMPSW	CMP <DS:ESI>, <ES:EDI> Dacă DF=0 atunci ESI=ESI+2 și EDI=EDI+2, altfel ESI=ESI-2 și EDI=EDI-2
CMPSD	CMP <DS:ESI>, <ES:EDI> Dacă DF=0 atunci ESI=ESI+4 și EDI=EDI+4, altfel ESI=ESI-4 și EDI=EDI-4

c. Prefixe pentru execuția repetată a unei instrucțiuni pe șiruri

Construcția:

prefix_instrucțiune *instrucțiune_pe_șiruri*

este echivalentă cu:

```
repeta:
    instrucțiune_pe_șiruri
loop repeta
```

unde **prefix_instrucțiune** este unul din următoarele prefixe:

REP (REPeat)	Repetă cât timp **ECX ≠ 0**
REPE (**REP**eat while **E**qual) **REPZ** (**REP**eat while **Z**ero)	Repetă cât timp **ECX ≠ 0** și **ZF = 1**
REPNE (**REP**eat while **N**ot **E**qual) **REPNZ** (**REP**eat while **N**ot **Z**ero)	Repetă cât timp **ECX ≠ 0** și **ZF = 0**

Observație

Prefixele **REPE/REPZ** și **REPNE/REPNZ** sunt destinate pentru a fi utilizate DOAR cu instrucțiunile **STOSx**, **MOVSx**, **SCASx** sau **CMPSx**.

Exemple

1. Copierea unui șir într-un alt șir

```
...
segment data use32 class=data
  s db 'a', 'b', 'c', 'd', 'e'
  len equ $-s           ; lungimea șirului s (în octeți)
  d times len db 0

segment code use32 class=code
    start:
      mov ecx, len      ; numărul de elemente ale șirului
      cld               ; DF = 0
      mov esi, s        ; ESI = offset-ul șirului sursă
      mov edi, d        ; EDI = offset-ul șirului destinație
      rep movsb         ; execuție repetată până când ECX = 0
...
```

2. Căutarea unei valori într-un șir

```
...
segment data use32 class=data
  s db 2, 1, 3, 5, 7
  len equ $-s           ; lungimea șirului s (în octeți)
```

```
segment code use32 class=code
    start:
      mov ecx, len      ; numărul de elemente ale șirului
      cld               ; DF = 0
      mov edi, s        ; ESI = offset-ul șirului sursă
      mov al, 5         ; AL = 5 (5 este valoarea căutată)
      repe scasb        ; execuție repetată până când ECX = 0 sau ZF=1
...
```

3. Cea mai rapidă cale de inițializare a unui bloc de memorie de mare dimensiune: **`rep stosb`**

EXERCIȚII

1. Se dă un șir de caractere S. Să se copieze elementele șirului S într-un alt șir de caractere D, folosind instrucțiuni pe șiruri.

2. Se dă un șir de caractere S format din litere mici. Să se construiască un șir de caractere D care să conțină literele din șirul inițial transformate în majuscule, folosind instrucțiuni pe șiruri.

3. Se dă un șir de octeti S. Să se construiască șirul de octeți D, care conține pe fiecare poziție numărul de biți care au valoarea 1 din octetul de pe poziția corespunzătoare din S.

Exemplu:

S: 5, 25, 55, 127

S în baza 2: 101, 11001, 10111, 1111111

D: 2, 3, 5, 7

4. Se dă un șir de cuvinte S. Să se construiască două șiruri de octeți:

– B1 care are ca elemente partea superioară a cuvintelor din s;

– B2 care are ca elemente partea inferioară a cuvintelor din s.

5. Se dau două șiruri de octeți S1 și S2 de lungimi egale. Să se determine poziția p în care elementele ambelor șiruri sunt egale.

6. Se dă un șir de octeți S. Să se ordoneze crescător elementele șirului.

7. Se dă un unui șir de octeți reprezentând un text (o succesiune de șiruri de caractere separate prin spații). Să se determine cuvintele din șir care sunt palindroame (ex. cojoc, capac etc.).