Seminar 4 – Instructiuni pe siruri. Probleme “complexe” pe siruri

Instructiuni pe siruri

* Au operanzi impliciti
* Sunt folosite astfel: fac ceva cu elementul curent si apoi ma mut la urmatorul element din sir
* Pentru a le utiliza:
  + Copiem offset-ul sirului sursa in registrul ESI
  + Copiem offset-ul sirului destinatie in registrul EDI
  + Configuram directia de parcurgere a sirului
    - “Stanga => dreapta” <=> CLD <=> DF = 0
    - “Dreapta => Stanga” <=> STD <=> DF = 1

Clasificare:

1. Instructiuni pentru transferul datelor

LODSB; AL <- [DS:ESI] si add ESI, 1 / sub ESI, 1 daca DF = 0 / DF = 1  
LODSW; AX <- [DS:ESI] si add ESI, 2 / sub ESI, 2 daca DF = 0 / DF = 1  
LODSD; EAX <- [DS:ESI] si add ESI, 4 / sub ESI, 4 daca DF = 0 / DF = 1  
  
STOSB; [ES:EDI] <- AL si add EDI, 1 / sub EDI, 1 daca DF = 0 / DF = 1  
STOSW; [ES:EDI] <- AX si add EDI, 2 / sub EDI, 2 daca DF = 0 / DF = 1  
STOSD; [ES:EDI] <- EAX si add EDI, 4 / sub EDI, 4 daca DF = 0 / DF = 1  
  
MOVSB; [ES:EDI] <- byte [DS:ESI] si add EDI, 1 si add ESI,1 / sub EDI, 1 si sub ESI, 1 daca DF = 0 / DF = 1  
MOVSW; [ES:EDI] <- word [DS:ESI] si add EDI, 2 si add ESI, 2/ sub EDI, 2 si sub ESI, 2 daca DF = 0 / DF = 1  
MOVSD  
; [ES:EDI] <- doubleword[DS:ESI] si add EDI, 4 si add ESI,4 / sub EDI, 4 si sub ESI, 4 daca DF = 0 / DF = 1

1. Instructiuni pentru compararea datelor

SCASB; CMP AL, [ES:EDI] si add EDI, 1 / sub EDI, 1 daca DF = 0 / DF = 1  
SCASW; CMP AX, [ES:EDI] si add EDI, 2 / sub EDI, 2 daca DF = 0 / DF = 1  
SCASD; CMP EAX, [ES:EDI] si add EDI, 4 / sub EDI, 4 daca DF = 0 / DF = 1  
  
CMPSB  
; CMP byte [DS:ESI], byte [ES:EDI] si add EDI, 1 si add ESI,1 / sub EDI, 1 si sub ESI, 1 daca DF = 0 / DF = 1  
CMPSW  
; CMP word[DS:ESI], word[ES:EDI] si add EDI, 2 si add ESI, 2/ sub EDI, 2 si sub ESI, 2 daca DF = 0 / DF = 1  
CMPSD  
; CMP doubleword[DS:ESI], doubleword[ES:EDI] si add EDI, 4 si add ESI,4 / sub EDI, 4 si sub ESI, 4 daca DF = 0 / DF = 1

Exercitii:

1. Se da un sir de octeti care contine litere mari. Sa se construiasca un nou sir de octeti care sa contina literele din sirul initial transformate in litere mici.

|  |  |
| --- | --- |
| Data segment | Code segment |
| A db ‘A’, ‘D’, ‘E’, ‘T’, ‘H’  len equ $ - A  B resb len | Mov ecx, len  Jecxz no\_len  Mov esi, A  Mov edi, B  cld  Iterate:  Lodsb  Add al, ‘a’ - ‘A’  stosb  Loop iterate  no\_len: |

1. Se da un sir de octeti. Sa se obtina sirul oglindit.

S1 db 1,2,3,4,5,6 => s2 db 6,5,4,3,2,1

|  |  |
| --- | --- |
| Data segment | Code segment |
| S1 db 1, 2, 3,4, 5, 6  L equ $-S1  S2 times L db 0 | Mov ecx, L  Jecxz Sfarsit  Mov esi, S1  Mov edi, S2  Add edi, L-1  repeta:  cld  Lodsb  Std  Stosb  Loop repeta  Sfarsit: |

1. Se dau doua siruri de cuvinte. Sa se concateneze sirul octetilor inferiori ai cuvintelor din primul sir cu sirul octetilor superiori ai cuvintelor din al doilea sir. Sirul rezultat trebuie sortat crescator in interpretare cu semn.

|  |  |
| --- | --- |
| Data segment | Code segment |
| s1 dw 1,2,3,4,5,6  l1 equ ($-s1) / 2  s2 dw 7,6,5,4  zl2 equ ($-s2) / 2  l equ l1+l2  sd times l db 0 | ;//////////////////////// COPIE din S1  Mov ecx, l1  Jecxz sfarsit\_1  Mov esi, s1  Mov edi, sd  Cld  repeta\_1:   Lodsw  stosb  Loop repeta\_1  sfarsit\_1:  ;//////////////////////// COPIE din S2  Mov esi, S2  Mov ecx, l2  Jecxz sfarsit\_2  repeta\_2:  Lodsw  Mov AL, AH  Stosb  Loop repeta\_2  sfarsit\_2:    ;//////////////////////// SORTARE SD  Mov eax, 1  Repeta\_3:  Cmp eax, 0  Je sfarsit\_3  Mov eax, 0  Mov esi,sd  Mov ecx, l-1  For:  Mov bl, [esi+1]  Cmp bl, [esi]  Jge urmatorulElement  Mov bh, [esi]  Mov [esi+1], bh  Mov [esi], bl  Mov eax, 1  urmatorulElement:  inc esi  Loop For  Jmp Repeta\_3  Sfarsit\_3 |

Ok =1

While (ok ==1 ){

Ok =0  
 for(I=1;I<=n-1;I++)  
 if (sd[I+1]<sd[I]) {interschimba valori si ok = 1}

}

1. Se da un sir de numere intregi reprezentate pe dublucuvant. Sa se construiasca sirul corespunzator octetilor strict negativi din reprezentarea in memorie a dublucuvintelor.

|  |  |
| --- | --- |
| Data segment | Code segment |
| S dd 1,2,3,4  L equ $ - S  Sd times L db 0 | Mov esi, S  Mov edi, Sd  Mov ecx, L  Jecxz Sfarsit  cld  Repeta:  Lodsb  Cmp al, 0  Jge Sfarsitbucla  stosb  Sfarsitbucla:  Loop Repeta  Sfarsit : |

1. Se da un sir de dublucuvinte. Sa se calculeze suma octetilor high ai cuvintelor low multiplii de 10.

|  |  |
| --- | --- |
| Data segment | Code segment |
| S dd 1,2,3,4  L equ ($ - S)/4  suma db 0 | Mov esi, S  Mov ecx, L  Jecxz Sfarsit  cld  Repeta:  LODSD  Mov AL,AH  MOV AH,0  Mov BL,10  Mov BH, AL  Div BL  cmp AH,0  JNE Sfarsitbucla  Add [suma],BL  Sfarsitbucla:  Loop Repeta  Sfarsit : |