

UNIVERSITATEA DE STAT DIN MOLDOVA

**FACULTATEA MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL INFORMATICĂ**

CEMÎRTAN CRISTIAN

Lucrarea individuală nr. 3
la disciplina *Arhitectura Calculatoarelor și Limbaje de Asamblare*

Coordonator: Sturza Greta, lector universitar

Chișinău, 2021

Cuprins

Sarcină.....	3
Cod sursă.....	3
Rezultate.....	5
Depanare.....	5
Concluzie.....	9

Sarcină

Să se introducă un șir cu lungimea de cel puțin 10 simboluri. Să se copieze din șirul **s1** în șirul **s2** toate simbolurile de pe poziție impară. Să se afișeze la consolă ambele șiruri.

Cod sursă

```
COMMENT *
    Lucrare individuala nr. 3, varianta 3
    Copyright Cemirtan Cristian 2021
    Grupa I 2101
*

.MODEL small
.STACK

len EQU 20                                ; lungimea constanta a sirului
clrf EQU 0Dh, 0Ah, '$'                    ; linia noua + terminator

.DATA
txt1 DB 'Introduceti sirul s1:', clrf
txt2 DB 0Ah, 'Sirul s1 este:', clrf ; 0Ah prefixat pentru ca 0Ah/21h lasa
la urma 0Dh pe ecran
txt3 DB 0Ah, 'Sirul s2 este:', clrf
prefix_s1 DB len + 1                      ; nr. max. de caractere - inca unu
pentru caracterul 0Dh
lungime_s1 DB ?
s1 DB len + 2 DUP (?)                     ; + 1 pentru terminatorul '$'
s2 DB len / 2 + 1 DUP ('$')               ; sa nu adaugam explicit terminatorul,
+ 1 pentru terminator

.CODE
; initializarea registrelor
    mov dx, @data
    mov ds, dx
    lea si, s1 ; s1 ca sursa
    lea di, s2 ; s2 ca destinatie

; afisare mesaj #1
    mov ah, 09h
    lea dx, txt1
    int 21h

; citire s1
    inc ah ; acum e 0Ah
    lea dx, prefix_s1
    int 21h ; lasa la urma in s1 caracterul 0Dh

; calculam pozitia sfarsitului s1
    xor bh, bh ; setam bh la 0
    mov bl, lungime_s1 ; incarcam lungimea lui s1
```

```

        mov BYTE PTR [bx + si + 1], '$'; adaugam terminator la s1, dupa
caracterul 0Dh
; BYTE PTR asigura ca destinatia este un octet, nu cuvant

; afisare mesaj #2
        dec ah          ; acum e 09h
        lea dx, txt2
        int 21h

; afisare s1
; ah este 09h
        mov dx, si      ; se poate - lea dx, s1
        int 21h

; afisare text 3
; ah este 09h
        lea dx, txt3
        int 21h

; testam daca cl este 0, seteaza flagul Z daca este adevarat
        test bl, bl
        jz iesire

buc1a:
        mov al, [si]
        mov [di], al      ; copiem caracterul din si in di
        add si, 2          ; 2 pasi inainte pentru ca copiem de pe pozitiile
impare
        inc di
        sub bl, 2          ; ibidem
        ja buc1a          ; salt daca flagurile Z si C au 0 in comun

afisare:
; ah este 09h
        lea dx, s2
        int 21h

iesire:
; iesire cu succes, cu codul de retur 0
        mov ax, 4C00h
        int 21h
END

```

Rezultate

```

D:\asm>tasm i3c
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:    i3c.ASM
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   452k

D:\asm>tlink i3c
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

D:\asm>i3c
Introduceti sirul s1:
ipipipipipipipip
Sirul s1 este:
ipipipipipipipip
Sirul s2 este:
iiiiiii
D:\asm>_

```

Figura 1. Programul executat cu succes.

Debanare

- De analizat modificările registrului **ip** și a segmentului de date:

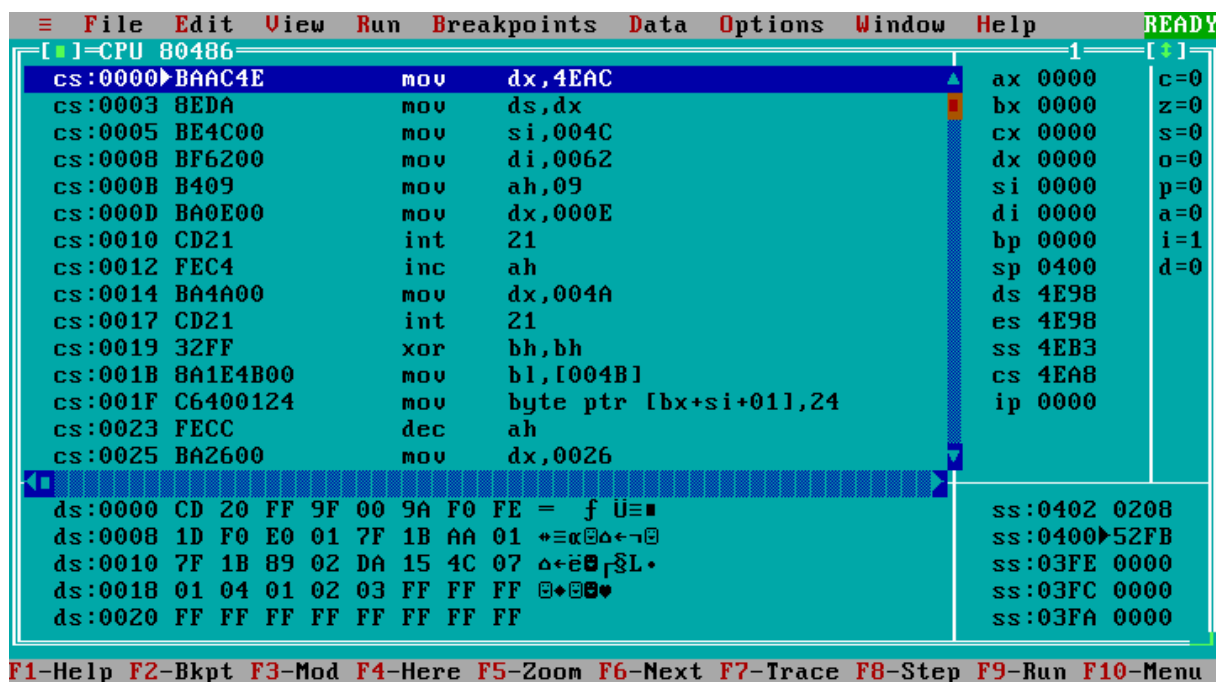


Figura 2. Depanatorul înainte de a rula programul.

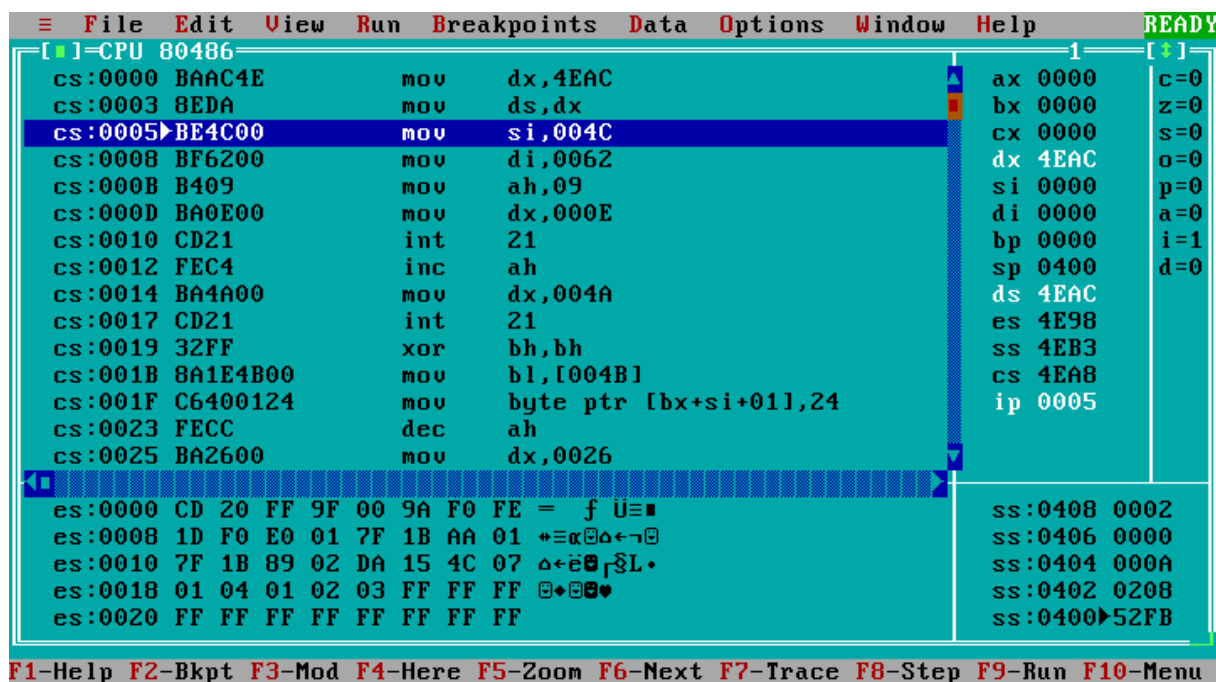


Figura 3. Registrul ds modificat explicit, după executarea instrucțiunii aflată la adresa cs:0003h. Registrul ip indică locația instrucțiunii ce va fi executată.

```
D:\asm>td i3c
Turbo Debugger Version 3.1 Copyright (c) 1988,92 Borland International
Introduceti sirul s1:
Individual 3_
```

Figura 4. Introducem de la tastatură, șirul s1. Întreruperea a avut loc la adresa cs:0017h.

```

File Edit View Run Breakpoints Data Options Window Help
[CPU 80486]
cs:000D BA0E00 mov dx,000E
cs:0010 CD21 int 21
cs:0012 FEC4 inc ah
cs:0014 BA4A00 mov dx,004A
cs:0017 CD21 int 21
cs:0019 32FF xor bh,bh
cs:001B 8A1E4B00 mov bl,[004B]
cs:001F C6400124 mov byte ptr [bx+si+01],24
cs:0023 FECC dec ah
cs:0025 BA2600 mov dx,0026
cs:0028 CD21 int 21
cs:002A 8BD6 mov dx,si
cs:002C CD21 int 21
cs:002E BA3800 mov dx,0038
cs:0031 CD21 int 21

es:0178 0A 53 69 72 75 6C 20 73 Sirul s
es:0180 32 20 65 73 74 65 3A 0D 2 este:
es:0188 0A 24 15 0C 49 6E 64 69 $$$Indi
es:0190 76 69 64 75 61 6C 20 33 vidual 3
es:0198 0D 24 00 00 00 00 00 00 R$

ss:0408 0002
ss:0406 0000
ss:0404 000A
ss:0402 0208
ss:0400 52FB
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

Figura 5. Segmentul de date modificat după prima întrerupere 0Ah. Instrucțiunea la adresa cs:001Fh, a adăugat explicit terminatorul ('\$') la adresa ds:bx+si+1. Registrul bx conține lungimea șirului s1.

```

File Edit View Run Breakpoints Data Options Window Help
[CPU 80486]
cs:0028 CD21 int 21
cs:002A 8BD6 mov dx,si
cs:002C CD21 int 21
cs:002E BA3800 mov dx,0038
cs:0031 CD21 int 21
cs:0033 84DB test bl,bl
cs:0035 7412 je 0049
cs:0037 8A04 mov al,[si]
cs:0039 8805 mov [di],al
cs:003B 83C602 add si,0002
cs:003E 47 inc di
cs:003F 80EB02 sub bl,02
cs:0042 77F3 ja 0037
cs:0044 BA6200 mov dx,0062
cs:0047 CD21 int 21

es:0198 0D 24 00 00 00 00 00 00 R$
es:01A0 00 00 24 24 24 24 24 24 $$$$$$
es:01A8 24 24 24 24 24 00 00 00 $$$$$$
es:01B0 00 00 00 00 00 00 00 00
es:01B8 00 00 00 00 00 00 00 00

ss:0408 0002
ss:0406 0000
ss:0404 000A
ss:0402 0208
ss:0400 52FB
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

Figura 6. Bucla ce conține două instrucțiuni de salt cu condiție (cs:0035h și cs:0042h). Ajută la modificarea șirului s2. Primul salt termină prematur programul când bl este 0 (s1 este gol). Al doilea salt reiterează bucla cât timp bl este mai mare decât 0.

```

File Edit View Run Breakpoints Data Options Window Help
[CPU 80486] 1 [i]
cs:0031 CD21      int     21          ax 0924      c=0
cs:0033 84DB      test    bl,bl        bx 0000      z=1
cs:0035 7412      je      0049         cx 0000      s=0
cs:0037 8A04      mov     al,[si]       dx 0062      o=0
cs:0039 8805      mov     [di],al       si 0058      p=1
cs:003B 83C602    add     si,0002       di 0068      a=0
cs:003E 47        inc     di            bp 0000      i=1
cs:003F 80EB02    sub     bl,02         sp 0400      d=0
cs:0042 77F3      ja      0037         ds 4EAC
cs:0044 BA6200    mov     dx,0062       es 4E98
cs:0047 CD21      int     21          ss 4EB3
cs:0049 B8004C    mov     ax,4C00       cs 4EA8
cs:004C CD21      int     21          ip 0049
cs:004E 49        dec     cx
cs:004F 6E        outsb

es:0198 0D 24 00 00 00 00 00 00  J$
es:01A0 00 00 49 64 76 64 61 20  Idvda
es:01A8 24 24 24 24 24 00 00 00  $$$$
es:01B0 00 00 00 00 00 00 00 00
es:01B8 00 00 00 00 00 00 00 00

ss:0408 0002
ss:0406 0000
ss:0404 000A
ss:0402 0208
ss:0400 52FB
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

```

D:\asm>td i3c
Turbo Debugger Version 3.1 Copyright (c) 1988,92 Borland International
Introduceti sirul s1:
Individual 3
Sirul s1 este:
Individual 3
Sirul s2 este:
Idvda _

```

Figurile 7 și 8. Șirul s2.

Concluzie

În concluzie, limbajul de asamblare x86, fiind un limbaj de programare de nivel jos, oferă o comoditate ce pare a fi de nivel înalt, care este invocarea întreruperilor. Întreruperile software de sistem, din categoria 21h, reprezintă o interfață de programare pentru sistemele de operare DOS, care conține o listă de funcții standarde pentru a gestiona mediul în care se află calculatorul.

De exemplu, interfața DOS ajută la invocarea serviciilor sale, de exemplu citire și afișarea unui șir de caractere, fără a cere programatorului să implementeze de la zero un subprogram ce răspunde de operații pe fluxuri standarde de intrare/ieșire.

Depanatorul „Turbo Debugger” ajută la depanarea produselor program, și de a verifica comportamentul programului în situații extreme sau cruciale, cum ar fi prelucrarea datelor de intrare, de o lungime vastă. De asemenea, afișează la ecran valorile registrelor a microprocesorului, și a segmentului de date, de cod, de stivă și de date suplimentare. Această conveniență previne programatorului să insereze explicit, în cod sursă, operațiile de afișare la ecran, a valorilor temporare, ce aduc un folos neutil în programe finale.