

UNIVERSITATEA DE STAT DIN MOLDOVA

FACULTATEA MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL INFORMATICĂ

CEMÎRTAN CRISTIAN

Lucrarea individuală nr. 4
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 10 simboluri. Să se copieze din șirul s1 în șirul s2 fiecare al treilea simbol s1. Să se afișeze la consolă ambele șiruri. Introducerea și afișarea șirurilor vor fi efectuate folosind funcțiile **3Fh** și **40h**.

Cod sursă

```
COMMENT *
    Lucrare individuala nr. 4, varianta 3
    Copyright Cemirtan Cristian 2021
    Grupa I 2101
*

.MODEL small
.STACK

len    EQU 10                ; lungimea constanta a sirului
crlf   EQU 0Dh, 0Ah         ; linia noua
crlft  EQU crlf, '$'        ; crlf cu terminator

.DATA
txt1 DB 'Introduceti sirul s1:', crlft
txt2 DB 'Sirul s1 este:', crlft
txt3 DB 'Sirul s2 este:', crlft

s1 DB len + 2 DUP (?)
s2 DB len / 3 DUP (?)

.CODE
; initializarea segmentului de date
    mov dx, @data
    mov ds, dx
    lea si, s1 ; s1 ca sursa, pozitionata la al treilea caracter
    lea di, s2 ; s2 ca destinatie

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

; citire s1
    mov ah, 3Fh
    xor bx, bx
    mov cx, len + 2
    mov dx, si
    int 21h

; salvam lungimea s1
    mov cl, al
```

```

; afisare mesaj #2
    mov ah, 09h
    lea dx, txt2
    int 21h

; afisare s1
    mov ah, 40h
    inc bl
    mov dx, si
    int 21h

; testare daca lungimea s1 este mai mare decat 2
    sub cl, 4          ; scapam de primele 2 caractere, 0Dh si 0Ah
    jbe iesire         ; salt daca flagul C sau Z este setat

; scapam de primele doua caractere din s1
    add si, 2

buc1a:
    mov al, [si]
    mov [di], al       ; copiem caracterul din si in di
    add si, 3          ; 3 pasi inainte pentru ca copiem fiecare al
treilea caracter
    inc di
    sub cl, 3          ; ibidem
    ja buc1a           ; salt daca flagurile Z si C au 0 in comun

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

; afisare s2
    mov ah, 40h
    mov cx, di
    sub cx, OFFSET s2 ; calculam lungimea s2
    lea dx, s2
    int 21h

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

```

Rezultate

```
E:\>tasm i4
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:    i4.ASM
Error messages:    None
Warning messages:  None
Passes:           1
Remaining memory:  452k

E:\>tlink i4
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

E:\>i4
Introduceti sirul s1:
12X45Y78Z0
Sirul s1 este:
12X45Y78Z0
Sirul s2 este:
XYZ
E:\>
```

Figura 1. Programul executat cu succes.

Depanare

- De analizat modificările registrelor **sp**, **ip** și a segmentului de date:

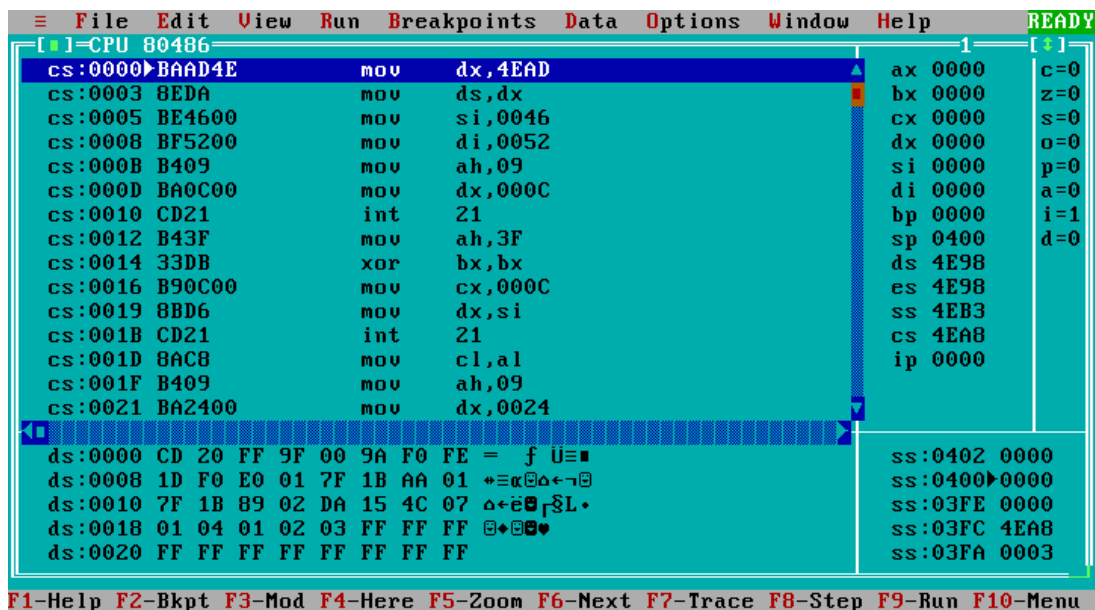


Figura 2. Depanatorul înainte de a rula programul.

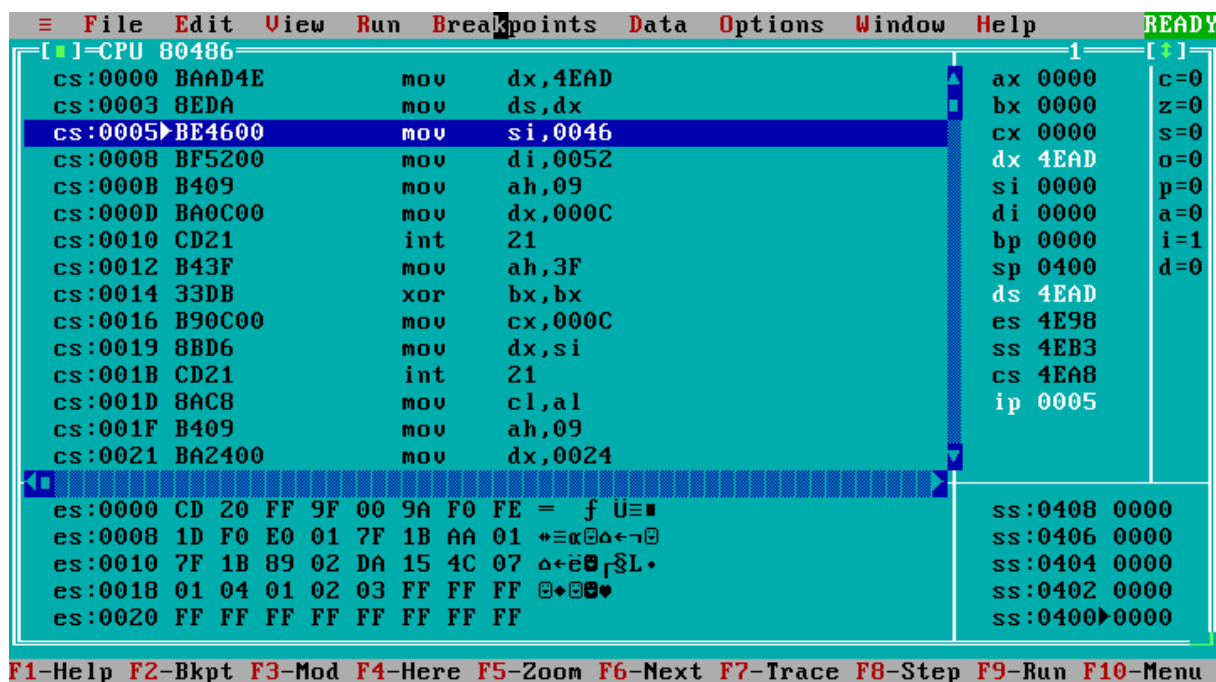


Figura 3. Registrul ds modificat explicit, după executarea instrucțiunii aflată la adresa cs:0003h. Registrul ip indică locația instrucțiunii ce va fi executată.

```
E:\>td i4
Turbo Debugger Version 3.1 Copyright (c) 1988,92 Borland International
Introduceti sirul s1:
Individ 4_
```

Figura 4. Introducem de la tastatură șirul s1. Întreruperea a avut loc la adresa cs:001Bh.

```

File Edit View Run Breakpoints Data Options Window Help READY
[ ]-CPU 80486
cs:0000 BAAD4E mov dx,4EAD ax 000B c=0
cs:0003 8EDA mov ds,dx bx 0000 z=1
cs:0005 BE4600 mov si,0046 cx 000C s=0
cs:0008 BF5200 mov di,0052 dx 0046 o=0
cs:000B B409 mov ah,09 si 0046 p=1
cs:000D BA0C00 mov dx,000C di 0052 a=0
cs:0010 CD21 int 21 bp 0000 i=1
cs:0012 B43F mov ah,3F sp 0400 d=0
cs:0014 33DB xor bx,bx ds 4EAD
cs:0016 B90C00 mov cx,000C es 4E98
cs:0019 8BD6 mov dx,si ss 4EB3
cs:001B CD21 int 21 cs 4EA8
cs:001D BAC8 mov cl,al ip 001D
cs:001F B409 mov ah,09
cs:0021 BA2400 mov dx,0024

es:0188 75 6C 20 73 32 20 65 73 ul s2 es ss:0408 0000
es:0190 74 65 3A 0D 0A 24 49 6E te:F$In ss:0406 0000
es:0198 64 69 76 69 64 20 34 0D divid 4F ss:0404 0000
es:01A0 0A 00 00 00 00 00 00 00 ss:0402 0000
es:01A8 00 00 00 00 00 00 00 00 ss:0400 0000

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 3Fh. În s1 s-au adăugat caracterele 0Dh și 0Ah, de la tasta Enter. Registrul ax conține lungimea șirului s1.

```

File Edit View Run Breakpoints Data Options Window Help READY
[ ]-CPU 80486
cs:002C CD21 int 21 ax 0034 c=1
cs:002E 80E904 sub cl,04 bx 0001 z=0
cs:0031 7624 jbe 0057 cx 00FE s=1
cs:0033 83C602 add si,0002 dx 0046 o=0
cs:0036 8A04 mov al,[sil si 0051 p=0
cs:0038 8805 mov [dil,al di 0055 a=1
cs:003A 83C603 add si,0003 bp 0000 i=1
cs:003D 47 inc di sp 0400 d=0
cs:003E 80E903 sub cl,03 ds 4EAD
cs:0041 77F3 ja 0036 es 4E98
cs:0043 B409 mov ah,09 ss 4EB3
cs:0045 BA3500 mov dx,0035 cs 4EA8
cs:0048 CD21 int 21 ip 0043
cs:004A B440 mov ah,40
cs:004C 8BCF mov cx,di

es:0188 75 6C 20 73 32 20 65 73 ul s2 es ss:0408 0000
es:0190 74 65 3A 0D 0A 24 49 6E te:F$In ss:0406 0000
es:0198 64 69 76 69 64 20 34 0D divid 4F ss:0404 0000
es:01A0 0A 00 64 69 34 00 00 00 di4 ss:0402 0000
es:01A8 00 00 00 00 00 00 00 00 ss:0400 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

```
E:\>td i4
Turbo Debugger Version 3.1 Copyright (c) 1988,92 Borland International
Introduceti sirul s1:
Individ 4
Sirul s1 este:
Individ 4
Sirul s2 este:
di4_
```

Figurile 6 și 7. Șirul s2.

- Segmentul de cod a programului, nu conține nici o operație de stivă (**push** și **pop**), de aceea nu s-a depistat nici o modificare a registrului **sp**.

Concluzie

În concluzie, funcțiile 3Fh și 40h, din întreruperea 21h, nu sunt specializate pentru a prelucra datele de la tastatură, deoarece aceste funcții sunt destinate pentru operații de intrare/ieșire cu fișierele din dispozitive de stocare.

Funcția 3Fh diferă în comportament, în unele sisteme DOS, când se citesc datele de la tastatură:

1. În DOSBox-X¹, citirea de la tastatură se termină când se depășește numărul maxim de caractere, specificat de registrul cx;
2. În MS-DOS, citirea nu se întrerupe, dar informația în exces va deveni citită implicit la următoarea invocare a acestei funcții.

În deosebire de funcția 0Ah, funcția 3Fh nu eliberează fluxul de intrare standard, de caractere suplimentare.

Funcția 40h, în deosebire de 09h, nu tratează caracterul '\$' ca terminator, dar necesită de specificat în registrul cx, numărul de caractere, ce vor fi scrise în fluxul de ieșire (în cazul nostru, cel standard pentru a afișa la consolă).

¹ DOSBox-X – Accurate emulation & retro-development. <https://dosbox-x.com/>