

Ministerul Educației, Culturii și Cercetării Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatica și Microelectronică

RAPORT

Lucrare de laborator Nr.2 la
Arhitectura Calculatoarelor
Var. 13

Tema: Elaborare programelor în limbajul de asamblare MASM în
Visual Studio

A efectuat:

Victoria Popescu

A verificat:

Colesnic V.

Chișinău 2019

Scopul lucrării:

Se prezinta problemele principale legate de conversii de date, reprezentarea datelor întregi, reprezentarea întregilor in format BCD, reprezentarea caracterelor si a şirurilor de caractere, reprezentarea valorilor reale, elemente de memorie, tipuri de date utilizate si modurile de adresare a operanzilor.

1. Cod sursa program 3.1

```
1. INCLUDE Irvine32.inc
2. .data
3. alfa    DW    3 DUP(?) ; Define Word? 3 cuvinte
4. .code
5. main proc
6. mov     ax,17 ; adresare indirecta a operandului sursa
7. mov     ax,10101b ;
8. mov     ax,11b ;
9. mov     ax,21o ;
10. mov     alfa,ax ; Adresare directa a operandului destinatie
11. mov     cx,ax ; Interschimba registrele ax si bx
12. mov     ax,bx ; Folosind registrul cx
13. mov     ax,cx ;
14. xchg     ax,bx ; Interschimba direct cele 2 registre.
15. mov     si,2
16. mov     alfa[si],ax ; Transferul cont reg ax in adresa conti si
17. mov     esi,2
18. mov     ebx,offset alfa ; Adresare imediata a operandului sursa /transf adr
efectitive alfa in ebx
19. lea     ebx,alfa ; Acelasi efect
20. mov     ecx,[ebx][esi] ; se uita la ad ment si transfera
21. mov     cx,alfa[2] ; Acelasi efect.
22. mov     cx,[alfa+2] ; Acelasi efect
23. mov     di,4
24. mov     byte ptr [ebx][edi],55h ;
25. mov     esi,2
26. mov     ebx,3
27. mov     alfa[ebx][esi],33h ; Adresare bazata indexata relativa a
28. ; destinatiei
29. mov     alfa[ebx+esi],33h ; Notatii echivalente
30. mov     [alfa+ebx+esi],33h
31. mov     [ebx][esi]+alfa,33h
32.
33. exit
34. main ENDP
35. END main
```

3.2 Codul programului:

Condiție:Să se calculeze expresia aritmetică: $e=((a+b*c-d)/f+g*h)/i$. Se cere obținerea fisierului executabil și rularea apoi pas cu pas

```

1. INCLUDE Irvine32.inc
2.
3.      ; Sa se calculeze expresia aritmetica: e=((a+b*c-d)/f+g*h)/i
4.      ; se considera a, d, f - cuvânt b, c, g, h, i -byte
5.      ; ca sa putem executa impartirea cu f convertim impartitorul la
dublucuvânt
6.      ; ne vor interesa doar caturile impartirilor, rezultatul va fi de tip
octet
7.
8.      .data
9.      a dw 5
10.     b db 6
11.     cd db 10
12.     d dw 5
13.     f dw 6
14.     g db 10
15.     h db 11
16.     i db 10
17.     interm dw ?
18.     rez db ?
19.
20.     .code
21.     main proc
22.     mov eax,0
23.     mov al, b
24.     imul cd      ; in ax avem b*c
25.     add ax, a    ; ax=b*c+a
26.     sub ax, d    ; ax=b*c+a-d
27.     cwd         ; am convertit cuvântul din ax, in dublu cuvântul ,
retinut in dx:ax
28.     idiv f      ; obtinem câtul în ax si restul în dx ax=(a+b*c-
d)/f
29.     mov interm, ax ; interm=(a+b*c-d)/f
30.     mov al, g
31.     imul h      ; ax=g*h
32.     add ax, interm ; ax=(a+b*c-d)/f+g*h
33.     idiv i      ; se obtine catul în al si restul în ah
34.     mov rez, al
35.
36. exit
37. main ENDP
38. END main

```

3.3 Codul programului:

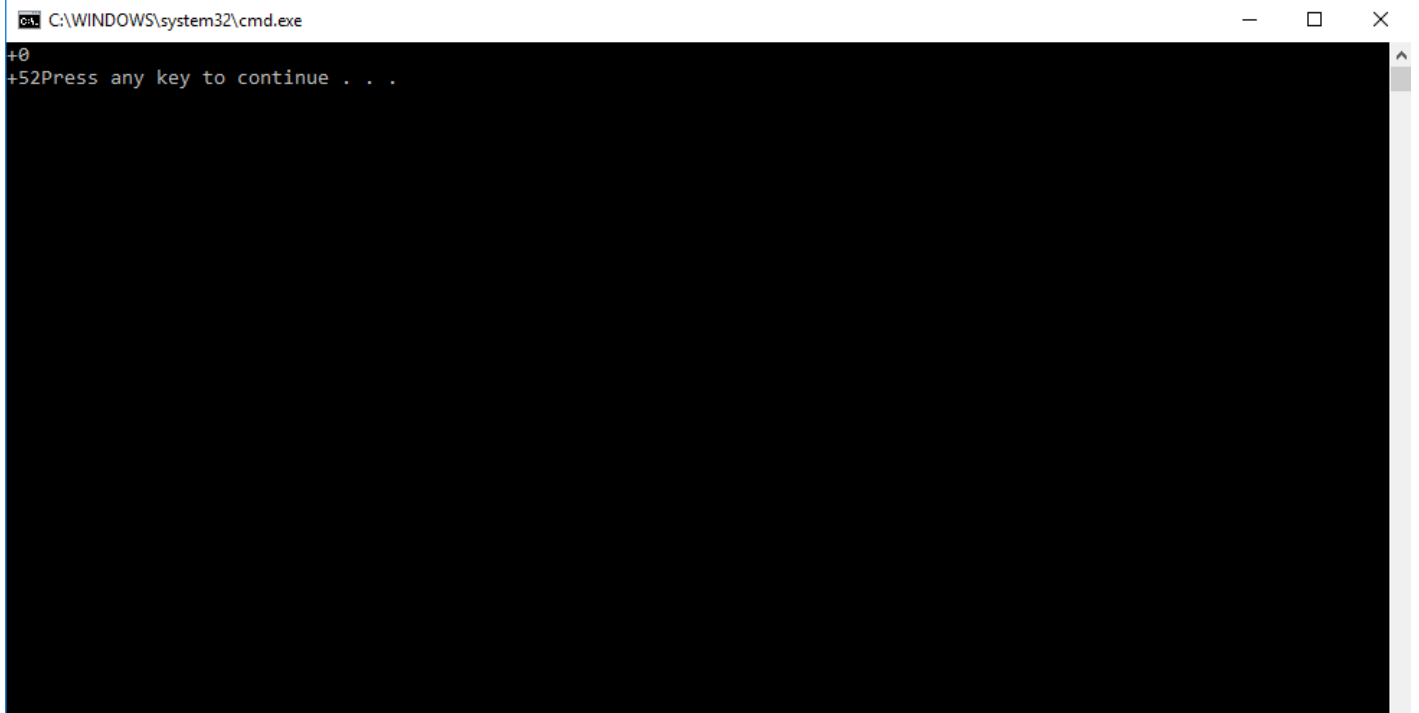
Condiție: Să se calculeze expresia aritmetică: $z = ((a+b)/i + 2*d)/e$. Se cere obținerea fișierului executabil și rularea apoi pas cu pas.

```
INCLUDE Irvine32.inc
```

```
    ; Sa se calculeze expresia aritmetica:  $z=((a+b)/i+2*d)/e$   
z=((10+6)/1+2*5)/1
```

```
    .data  
        a db 10      ; declararea si atribuirea variabilei a de tip double  
byte 10  
        b db 6       ; declararea si atribuirea variabilei b de tip double  
byte 6  
        cd db 10     ; declararea si atribuirea variabilei cd de tip  
double byte  
        d db 5  
        e db 2  
        g db 2  
        h db 11  
        i db 1  
        interm db ?   ; declararea si initializarea variabilei interm de  
tip double byte  
        rez db ?
```

```
    .code          ; directiva de a declara inceputul de cod  
main proc         ; declararea procedurii cu numele main  
    mov eax,0      ; initializarea cu 0 registrul ax  
    mov al, a      ; atribuie registrului al valoarea variabilei a  
    add al, b  
    mov bl, i      ; in ax avem a+b  
    idiv bl        ; ax=(a+b)/i  
    mov interm, al ; interm=(a+b)/i  
    mov al, d  
    mov bl, 2  
    imul bl        ; ax=2*d  
    add al, interm ; ax=(a+b)/c+2*d  
    idiv e         ; ax=((a+b)/c+2*d)/e  
    mov rez, al  
  
    mov bl, al  
    mov al, 0  
    call WriteInt  
  
    call Crlf  
    mov al, bl  
    call WriteInt  
  
exit             ; apel la procedura de iesire din program, inclusiv din irvince  
main ENDP       ; sfarsitul procedurii main  
END main        ; finalizarea programului
```



A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The command prompt shows two lines of assembly code: "+0" and "+52Press any key to continue . . .". The rest of the window is black, indicating that the program has finished execution and is waiting for a key press to continue.

Concluzie:

În urma efectuării lucrării de laborator nr. 2 s-a realizat un program ce efectuează mai multe operații aritmetice pentru calcularea unei valori finale Z. S-au utilizat două tipuri de date – db -define byte și dw - define word. Totodată s-au utilizat și instrucțiuni precum add, mov, imul, div, etc. și s-au pus în practică cunoștințele referitor la modurile de adresare a operanzilor. În concluzie s-a familiarizat cu limbajul Assembler.