Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Automatica și Tehnologii Informaționale

**RAPORT**

Lucrare de laborator Nr.1

*la Arhitectura Calculatorului*

A efectuat: st. gr. TI-143

D. Besliu

A verificat: lect. sup.

V. Colesnic

Chişinău 2016

**Tema:** **Elaborare programelor în limbajul de asamblare MASM în Mediul de Programare Visual Studio**

**SCOPUL LUCRARII**

Lucrarea urmărește familiarizarea studenților cu **regiştrii microprocesoarelor de 16, 32 şi 64 biți**, cu mediul de dezvoltare a programelor scrise pus la dispoziție de către produsul Visual Studio al firmei Microsoft. Se prezinta setările mediului Visual Studio necesare pentru a crea, rula şi depăna aplicaţii elaborate in limbaj de asamblare MASM utilizând regiştri pe 16, 32 şi 64 biţi.

**MERSUL LUCRĂRII**

Codul programului pe 16-bit

INCLUDE Irvine16.inc

.DATA ; directiva ce declară începutul segmentului de date

Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]$' ;definirea unui byte cu offset-ul Prompt

Dad DB 13,10,'Vei deveni!',13,10,'$' ; definirea unui byte cu offset-ul Dad

Nud DB 13,10,'Vei deveni filosof!',13,10,'$' ; definirea unui byte cu offset-ul Nud

.CODE ; directiva ce declară începutul segmentului de cod

main PROC ; declararea procedurii main

mov ax,@data ; Initializarea segmentului de date **ds**

mov ds,ax ; cu adresa datelor

mov dx,OFFSET Promt ; în dx- deplasamentul (offset) sirului Promt

mov ah,9 ; functia MSDOS, codul funcţiei 9 – afisarea sirului

Int 21h ; întreruperea 21h – apel la serviciul MSDOS

mov ah,1 ; functia MSDOS, codul funcţiei 1 – introducerea de la tastatură(codul tastei)

Int 21h ; codul tastei in registrul **al**

cmp al,'y' ; compararea conţinutului registrului al cu codul ASCII a literei ***y***

jz IsDad ; salt condiţionat (jz-jump if zero), dacă rezultatul comparării este zero, salt la eticheta **IsDad**

cmp al,'n' ; compararea- din **al** se scade codul ASCII a literei ***n***

jz IsNud ; da, rezultatul comparării este zero, salt la eticheta **IsNud**

IsDad: mov dx,OFFSET Dad ; în dx- offsetul sirului Dad

Jmp SHORT Disp ; Salt necondiţionat la eticheta ”Disp”

IsNud: mov dx,OFFSET Nud ; în dx- offsetul șirului ”Nud”

Disp: mov ah,9 ; functia MSDOS – afisarea sirului

Int 21h ; apel la serviciul MSDOS

mov ah,1 ; asteptarea unui clic

Int 21h

Exit ; apel la procedura de iesirea din program, din fisierul **Irvine16.inc**

main ENDP ; sfârşitul procedurii **main**

END main ; finalizarea programului

Listing 16-bit

Microsoft (R) Macro Assembler Version 14.00.23026.0 02/21/16 20:56:12

main.asm Page 1 - 1

;exemplu de program pe 16 biti

INCLUDE Irvine16.inc

C ; Irvine16.inc - Include file for programs using

C ; the Irvine16.lib (Real-address mode library).

C

C ; Last update: 7/29/05

C

C .NOLIST

C .LIST

C

0000 .DATA

0000 44 6F 72 65 73 74 Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]$'

69 20 73 61 20 64

65 76 69 69 20 70

72 6F 67 72 61 6D

61 74 6F 72 3F 28

64 61 2F 6E 75 29

2D 5B 79 2F 6E 5D

24

002B 0D 0A 56 65 69 20 Dad DB 13,10,'Vei deveni!',13,10,'$'

64 65 76 65 6E 69

21 0D 0A 24

003B 0D 0A 56 65 69 20 Nud DB 13,10,'Vei deveni filosof!',13,10,'$'

64 65 76 65 6E 69

20 66 69 6C 6F 73

6F 66 21 0D 0A 24

0000 .CODE

0000 main PROC

0000 B8 ---- R mov ax,@data

0003 8E D8 mov ds,ax

0005 BA 0000 R mov dx,OFFSET Promt

0008 B4 09 mov ah,9

000A CD 21 Int 21h

000C B4 01 mov ah,1

000E CD 21 Int 21h

0010 3C 79 cmp al,'y'

0012 74 04 jz IsDad

0014 3C 6E cmp al,'n'

0016 74 05 jz IsNud

0018 BA 002B R IsDad: mov dx,OFFSET Dad

001B EB 03 Jmp SHORT Disp

001D BA 003B R IsNud: mov dx,OFFSET Nud

0020 B4 09 Disp: mov ah,9

0022 CD 21 Int 21h

0024 B4 01 mov ah,1

0026 CD 21 Int 21h

Exit

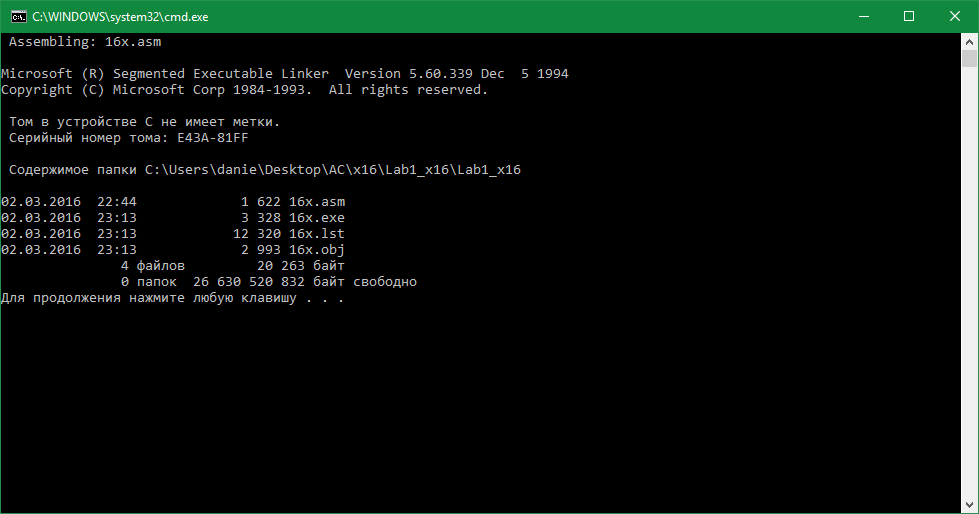
002C main ENDP

END main

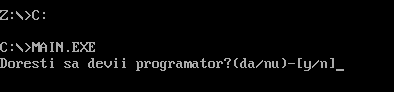
Microsoft (R) Macro Assembler Version 14.00.23026.0 02/21/16 20:56:12

main.asm Symbols 2 – 1

Build programul



Rulearea programului



Codul programului 32-bit

INCLUDE Irvine32.inc

.data

Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]',0

Dad DB 13,10,'Vei deveni!',13,10,0

Nud DB 13,10,'Vei deveni filosof!',13,10,0

.code

main PROC

mov edx,OFFSET Promt

call WriteString

call ReadChar

cmp al,'y'

jz IsDad

cmp al,'n'

jz IsNud

IsDad: mov edx,OFFSET Dad

call WriteString

jmp ex

IsNud: mov edx,OFFSET Nud

call WriteString

ex:

; exit

main ENDP

END main

Listing 32-bit

Microsoft (R) Macro Assembler Version 14.00.23026.0 02/18/16 18:53:15

main.asm Page 1 - 1

INCLUDE Irvine16.inc

C ; Irvine16.inc - Include file for programs using

C ; the Irvine16.lib (Real-address mode library).

C

C ; Last update: 7/29/05

C

C .NOLIST

C .LIST

C

0000 .data

0000 44 6F 72 65 73 74 Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]',0

69 20 73 61 20 64

65 76 69 69 20 70

72 6F 67 72 61 6D

61 74 6F 72 3F 28

64 61 2F 6E 75 29

2D 5B 79 2F 6E 5D

00

002B 0D 0A 56 65 69 20 Dad DB 13,10,'Vei deveni!',13,10,0

64 65 76 65 6E 69

21 0D 0A 00

003B 0D 0A 56 65 69 20 Nud DB 13,10,'Vei deveni filosof!',13,10,0

64 65 76 65 6E 69

20 66 69 6C 6F 73

6F 66 21 0D 0A 00

0000 .code

0000 main PROC

0000 66| BA 00000000 R mov edx,OFFSET Promt

0006 E8 0000 E call WriteString

0009 E8 0000 E call ReadChar

000C 3C 79 cmp al,'y'

000E 74 04 jz IsDad

0010 3C 6E cmp al,'n'

0012 74 0B jz IsNud

0014 66| BA 0000002B R IsDad: mov edx,OFFSET Dad

001A E8 0000 E call WriteString

001D EB 09 jmp ex

001F 66| BA 0000003B R IsNud: mov edx,OFFSET Nud

0025 E8 0000 E call WriteString

0028 ex:

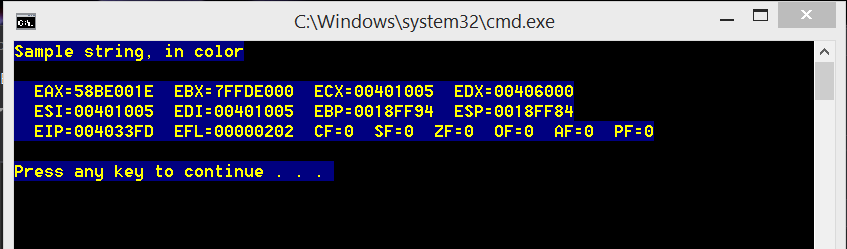
exit

002C main ENDP

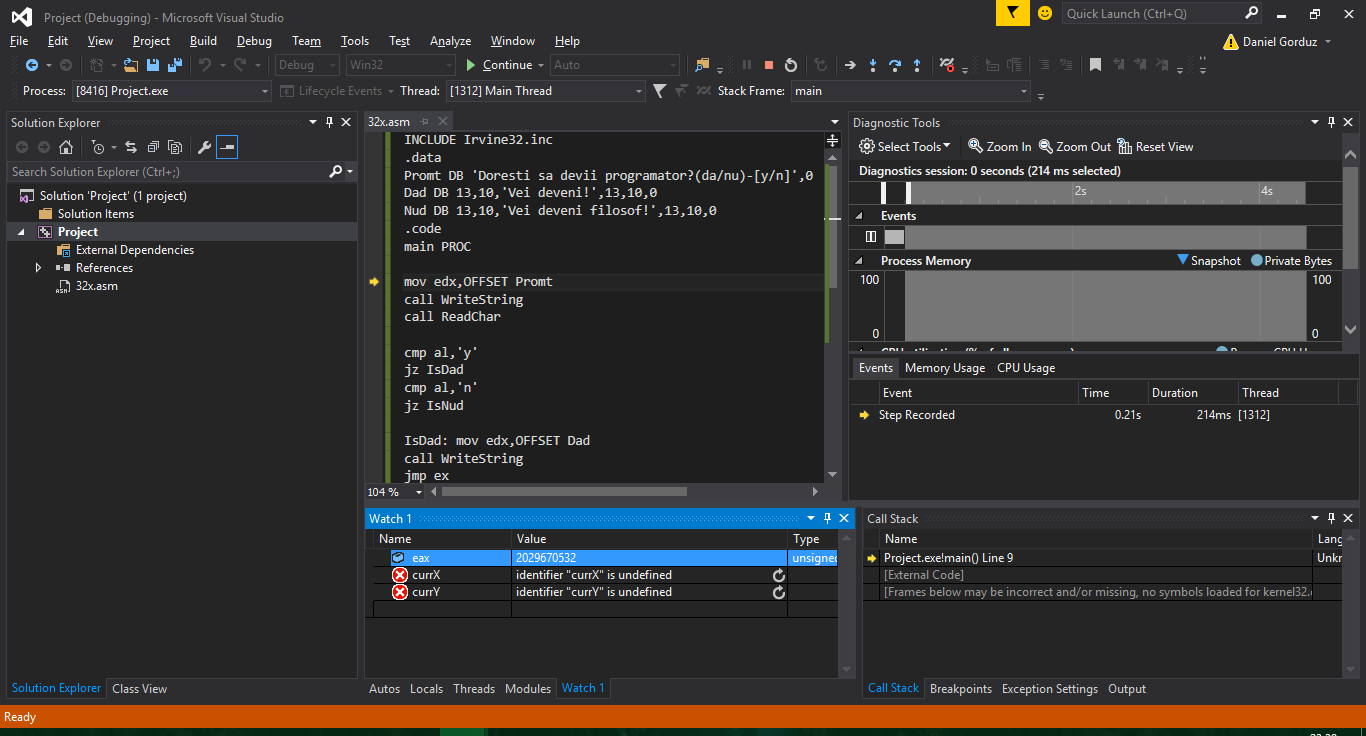
END main

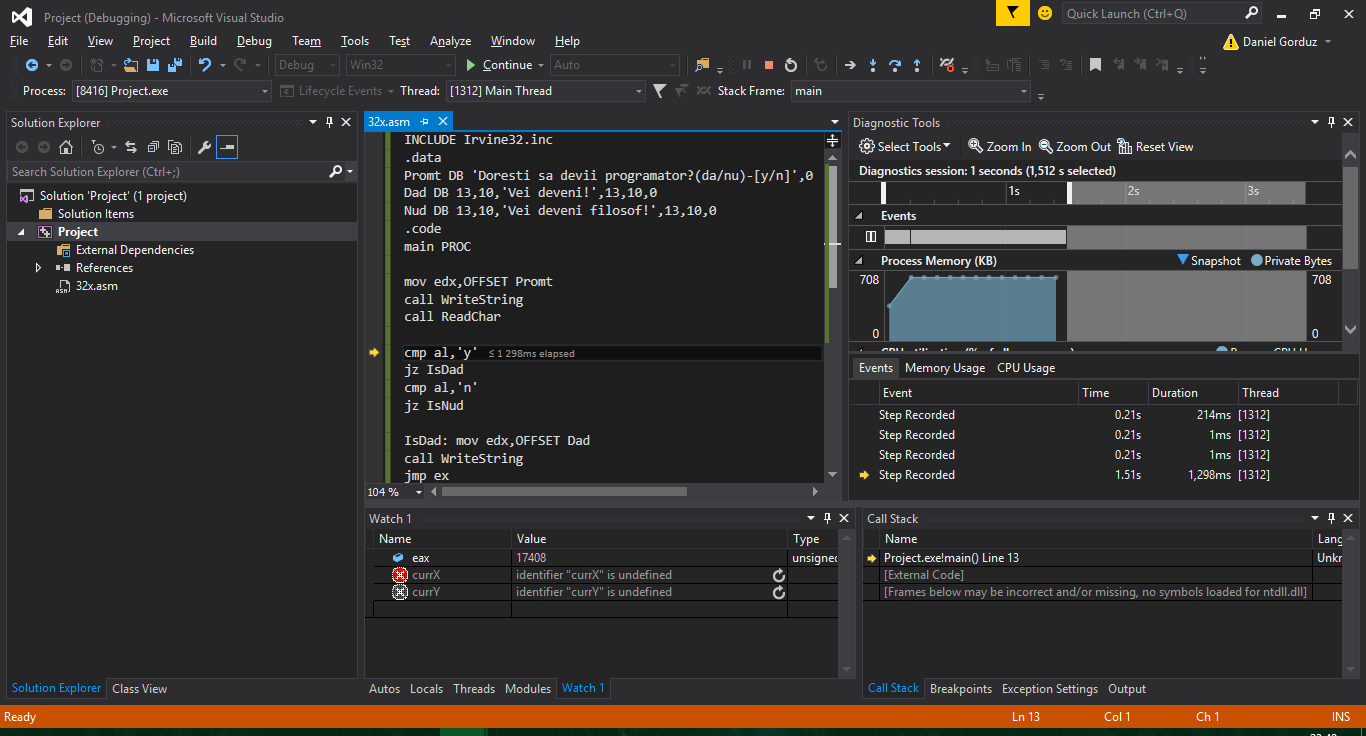
Microsoft (R) Macro Assembler Version 14.00.23026.0 02/18/16 18:53:15

main.asm Symbols 2 – 1



Debugging





Codul 64-bit

; AddTwoSum\_64.asm - Chapter 3 example.

ExitProcess proto

.data

sum qword 0

.code

main proc

mov rax,5

add rax,6

mov sum,rax

mov ecx,0

call ExitProcess

main endp

end

Listing 64-bit

Microsoft (R) Macro Assembler (x64) Version 14.00.23506.0 02/25/16 12:55:46

..\ch03\64 bit\AddTwoSum\_64.asm Page 1 - 1

; AddTwoSum\_64.asm - Chapter 3 example.

ExitProcess proto

00000000 .data

00000000 sum qword 0

0000000000000000

00000000 .code

00000000 main proc

00000000 48/ C7 C0 mov rax,5

00000005

00000007 48/ 83 C0 06 add rax,6

0000000B 48/ 89 05 mov sum,rax

00000000 R

00000012 B9 00000000 mov ecx,0

00000017 E8 00000000 E call ExitProcess

0000001C main endp

end

Microsoft (R) Macro Assembler (x64) Version 14.00.23506.0 02/25/16 12:55:46

..\ch03\64 bit\AddTwoSum\_64.asm Symbols 2 - 1

Procedures, parameters, and locals:

N a m e Type Value Attr

ExitProcess . . . . . . . . . . P 00000000 Length= 00000000 External

main . . . . . . . . . . . . . . P 00000000 \_TEXT Length= 0000001C Public

Symbols:

N a m e Type Value Attr

sum . . . . . . . . . . . . . . QWord 00000000 \_DATA

0 Warnings

0 Errors

**Concluzie**

In urma executiei acestei lucrari de laborator am fost nevoit sa setez mediul Visual Studio pentru a avea posibilitate de a programa în assembler. Am studiat regiștrii pe 16-bit și pe 32-bit. În baza programului scris putem face o comparație între modul de lucru cu un tip sau altul.

Citind conținutul fișierelor listing, putem vizualiza cum arată codul scris și cum acesta este interpretat de către compilator. Un moment interesant a fost faptul ca pentru a putea lucra cu sistemul pe 64-bit este necesar de a utiliza DOSBox, deoarece programul nu poate direct interactiona cu programele pe 16 bit