Ministerul Educaţiei, Tineretului şi Sportului al Republicii Moldova Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică Departamentul Informatică și Ingineria Sistemelor

RAPORT

Lucrare de laborator nr.1

la Arhitecturi de Calculatoare

Tema:,,Elaborarea programelor în limbajul de asamblare MASM în Visual Studio ’’

A efectuat: st.gr.TI-214

Buza Cătălin

A verificat: asist.univ.

Secrieru Andrian

Chişinău 2023

**SCOPUL LUCRĂRII**

Lucrarea urmărește familiarizarea studenților cu regiştrii microprocesoarelor pe 16, 32 şi 64 de biți, cu mediul de dezvoltare a programelor Visual Studio. Se prezintă setările mediului Visual Studio necesare pentru a elabora, rula şi depăna aplicaţiile elaborate în limbajul de asamblare MASM utilizând regiştri pe 16, 32 şi 64 de biţi.

# MODUL DE LUCRU

1.Se vor asambla si rula exemplele de programe prezentate obţinându‑se fișiere *.EXE,* ele se află în Project, cu numele Project.exe ;

2.Se vor obține și se vor analiza fișierele listing ale programelor Project.lst;

3.Se va rula sub *Debug* pas cu pas (clic F10) exemplele de programe actualizând Registers şi Watch1(pentru variabile);

**Codul sursa .asm 16 biți**

|  |
| --- |
| ;exemplu de program pe 16 biti  INCLUDE Irvine16.inc  .DATA  Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]$'  Dad DB 13,10,'Vei deveni!',13,10,'$'  Nud DB 13,10,'Vei deveni filosof!',13,10,'$'  .CODE ;directiva ce declara inceputul segmentului de cod  main PROC ;se indica procedura cu numele main  mov ax,@data ;Initializarea segmentului de date ds  mov ds,ax ;cu adresa datelor  mov dx,OFFSET Promt ;in dx- deplasamentul (offset) sirului Promt  mov ah,9 ;functia MSDOS, codul functiei 9 - afisarea sirului  Int 21h ;intreruperea 21h - apel la serviciul MSDOS  mov ah,1 ;functia MSDOS, codul functiei 1 - introducerea de la tastatura(codul tastei)  Int 21h ;codul tastei in registrul al  cmp al,'y' ;compararea continutului registrului al cu codul ASCII a literei y  jz IsDad ;salt conditionat (jz-jump if zero), daca rezultatul compararii este zero, salt la eticheta IsDad  cmp al,'n' ;compararea- din al se scade codul ASCII a literei n  jz IsNud ;da, rezultatul compararii este zero, salt la eticheta IsNud  IsDad: mov dx,OFFSET Dad ;in dx- offsetul sirului Dad  Jmp SHORT Disp ;Salt neconditionat la eticheta "Disp"  IsNud: mov dx,OFFSET Nud ;in dx- offsetul sirului "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 ;sfarsitul procedurii main  END main ;finalizarea programului/ punctul de intrare in program |

**Fișier -listing 16 biți**

|  |
| --- |
| Microsoft (R) Macro Assembler Version 12.00.40629.0 01/31/22 23:53:14  x16.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 ;directiva ce declara inceputul segmentului de cod  0000 main PROC ;se indica procedura cu numele main  0000 B8 ---- R mov ax,@data ;Initializarea segmentului de date ds  0003 8E D8 mov ds,ax ;cu adresa datelor  0005 BA 0000 R mov dx,OFFSET Promt ;in dx- deplasamentul (offset) sirului Promt  0008 B4 09 mov ah,9 ;functia MSDOS, codul functiei 9 - afisarea sirului  000A CD 21 Int 21h ;intreruperea 21h - apel la serviciul MSDOS  000C B4 01 mov ah,1 ;functia MSDOS, codul functiei 1 - introducerea de la tastatura(codul tastei)  000E CD 21 Int 21h ;codul tastei in registrul al  0010 3C 79 cmp al,'y' ;compararea continutului registrului al cu codul ASCII a literei y  0012 74 04 jz IsDad ;salt conditionat (jz-jump if zero), daca rezultatul compararii este zero, salt la eticheta IsDad  0014 3C 6E cmp al,'n' ;compararea- din al se scade codul ASCII a literei n  0016 74 05 jz IsNud ;da, rezultatul compararii este zero, salt la eticheta IsNud  0018 BA 002B R IsDad: mov dx,OFFSET Dad ;in dx- offsetul sirului Dad  001B EB 03 Jmp SHORT Disp ;Salt neconditionat la eticheta "Disp"  001D BA 003B R IsNud: mov dx,OFFSET Nud ;in dx- offsetul sirului "Nud"  0020 B4 09 Disp: mov ah,9 ;functia MSDOS - afisarea sirului  0022 CD 21 Int 21h ;apel la serviciul MSDOS  0024 B4 01 mov ah,1 ;asteptarea unui clic  0026 CD 21 Int 21h  Exit ;apel la procedura de iesirea din program, din fisierul Irvine16.inc  002C main ENDP ;sfarsitul procedurii main  END main ;finalizarea programului/ punctul de intrare in program  Microsoft (R) Macro Assembler Version 12.00.40629.0 01/31/22 23:53:14  x16.asm Symbols 2 - 1  Structures and Unions:  N a m e Size  Offset Type  CursorPosStruc . . . . . . . . . 0002  Ycoord . . . . . . . . . . . . 0000 Byte  Xcoord . . . . . . . . . . . . 0001 Byte  ExtGetDskFreSpcStruc . . . . . . 002C  StructSize . . . . . . . . . . 0000 Word  Level . . . . . . . . . . . . 0002 Word  SectorsPerCluster . . . . . . 0004 DWord  BytesPerSector . . . . . . . . 0008 DWord  AvailableClusters . . . . . . 000C DWord  TotalClusters . . . . . . . . 0010 DWord  AvailablePhysSectors . . . . . 0014 DWord  TotalPhysSectors . . . . . . . 0018 DWord  AvailableAllocationUnits . . . 001C DWord  TotalAllocationUnits . . . . . 0020 DWord  Rsvd . . . . . . . . . . . . . 0024 DWord  FPU\_ENVIRON . . . . . . . . . . 001C  controlWord . . . . . . . . . 0000 Word  statusWord . . . . . . . . . . 0004 Word  tagWord . . . . . . . . . . . 0008 Word  instrPtrLow . . . . . . . . . 000C Word  opcodeAndInstrPtrHi . . . . . 0010 DWord  operandPtrLow . . . . . . . . 0014 Word  operandPtrHi . . . . . . . . . 0018 DWord  TimeRecord . . . . . . . . . . . 0004  hours . . . . . . . . . . . . 0000 Byte  minutes . . . . . . . . . . . 0001 Byte  seconds . . . . . . . . . . . 0002 Byte  hhss . . . . . . . . . . . . . 0003 Byte  VideoInfoStruc . . . . . . . . . 0041  supportedInfoPtr . . . . . . . 0000 DWord  videoMode . . . . . . . . . . 0004 Byte  numCharColumns . . . . . . . . 0005 Word  videoBufferLen . . . . . . . . 0007 Word  videoBufferStartPtr . . . . . 0009 Word  cursors . . . . . . . . . . . 000B Word  cursorStartLine . . . . . . . 001B Byte  cursorEndLine . . . . . . . . 001C Byte  activeDisplayPage . . . . . . 001D Byte  adapterBasePortAddr . . . . . 001E Word  currentRegister3B8or3D8 . . . 0020 Byte  currentRegister3B9or3D9 . . . 0021 Byte  numCharRows . . . . . . . . . 0022 Byte  characterScanHeight . . . . . 0023 Word  activeDisplayCode . . . . . . 0025 Byte  inactiveDisplayCode . . . . . 0026 Byte  numberOfColors . . . . . . . . 0027 Word  numberOfVideoPages . . . . . . 0029 Byte  numberOfScanLines . . . . . . 002A Word  primaryCharBlock . . . . . . . 002C Byte  secondaryCharBlock . . . . . . 002D Byte  miscStateInfo . . . . . . . . 002E Byte  videoMemAvail . . . . . . . . 0032 Byte  savePointerStateInfo . . . . . 0033 Byte  Segments and Groups:  N a m e Size Length Align Combine Class  DGROUP . . . . . . . . . . . . . GROUP  \_DATA . . . . . . . . . . . . . 16 Bit 0053 Word Public 'DATA'  STACK . . . . . . . . . . . . . 16 Bit 0200 Para Stack 'STACK'  \_TEXT . . . . . . . . . . . . . 16 Bit 002C Word Public 'CODE'  Procedures, parameters, and locals:  N a m e Type Value Attr  Clrscr . . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Crlf . . . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Delay . . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  DumpMem . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  DumpRegs . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  GetCommandtail . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  GetMaxXY . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  GetMseconds . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Gotoxy . . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  IsDigit . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Random32 . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  RandomRange . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Randomize . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  ReadChar . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  ReadFloat . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  ReadHex . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  ReadInt . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  ReadString . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  SetTextColor . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  ShowFPUStack . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Str\_compare . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Str\_copy . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Str\_length . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Str\_trim . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  Str\_ucase . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WaitMsg . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteBinB . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteBin . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteChar . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteDec . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteFloat . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteHexB . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteHex . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteInt . . . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  WriteString . . . . . . . . . . P Near 0000 Length= 0000 External STDCALL  main . . . . . . . . . . . . . . P Near 0000 \_TEXT Length= 002C Public STDCALL  IsDad . . . . . . . . . . . . L Near 0018 \_TEXT  IsNud . . . . . . . . . . . . L Near 001D \_TEXT  Disp . . . . . . . . . . . . . L Near 0020 \_TEXT  Symbols:  N a m e Type Value Attr  @CodeSize . . . . . . . . . . . Number 0000h  @DataSize . . . . . . . . . . . Number 0000h  @Interface . . . . . . . . . . . Number 0003h  @Model . . . . . . . . . . . . . Number 0002h  @cmdline . . . . . . . . . . . . Text OFFSET \_\_98765765cmdline  @code . . . . . . . . . . . . . Text \_TEXT  @data . . . . . . . . . . . . . Text DGROUP  @fardata? . . . . . . . . . . . Text FAR\_BSS  @fardata . . . . . . . . . . . . Text FAR\_DATA  @stack . . . . . . . . . . . . . Text DGROUP  ACTION\_CREATED\_OPENED . . . . . Number 0002h  ACTION\_OPENED . . . . . . . . . Number 0001h  ACTION\_REPLACED\_OPENED . . . . . Number 0003h  Dad . . . . . . . . . . . . . . Byte 002B \_DATA  FILE\_CREATE . . . . . . . . . . Number 0010h  FILE\_OPEN . . . . . . . . . . . Number 0001h  FILE\_TRUNCATE . . . . . . . . . Number 0002h  LIGHT\_BLUE . . . . . . . . . . . Number 0009h  LIGHT\_CYAN . . . . . . . . . . . Number 000Bh  LIGHT\_GRAY . . . . . . . . . . . Number 0007h  LIGHT\_GREEN . . . . . . . . . . Number 000Ah  LIGHT\_MAGENTA . . . . . . . . . Number 000Dh  LIGHT\_RED . . . . . . . . . . . Number 000Ch  Nud . . . . . . . . . . . . . . Byte 003B \_DATA  OPEN\_ACCESS\_READONLY . . . . . . Number 0000h  OPEN\_ACCESS\_READWRITE . . . . . Number 0002h  OPEN\_ACCESS\_RO\_NOMODLASTACCESS . Number 0004h  OPEN\_ACCESS\_WRITEONLY . . . . . Number 0001h  OPEN\_FLAGS\_ALIAS\_HINT . . . . . Number 0400h  OPEN\_FLAGS\_COMMIT . . . . . . . Number 4000h  OPEN\_FLAGS\_NOCRITERR . . . . . . Number 2000h  OPEN\_FLAGS\_NOINHERIT . . . . . . Number 0080h  OPEN\_FLAGS\_NO\_BUFFERING . . . . Number 0100h  OPEN\_FLAGS\_NO\_COMPRESS . . . . . Number 0200h  OPEN\_SHARE\_COMPATIBLE . . . . . Number 0000h  OPEN\_SHARE\_DENYNONE . . . . . . Number 0040h  OPEN\_SHARE\_DENYREADWRITE . . . . Number 0010h  OPEN\_SHARE\_DENYREAD . . . . . . Number 0030h  OPEN\_SHARE\_DENYWRITE . . . . . . Number 0020h  Promt . . . . . . . . . . . . . Byte 0000 \_DATA  \_A\_ARCH . . . . . . . . . . . . Number 0020h  \_A\_HIDDEN . . . . . . . . . . . Number 0002h  \_A\_NORMAL . . . . . . . . . . . Number 0000h  \_A\_RDONLY . . . . . . . . . . . Number 0001h  \_A\_SYSTEM . . . . . . . . . . . Number 0004h  \_A\_VOLID . . . . . . . . . . . . Number 0008h  black . . . . . . . . . . . . . Number 0000h  blue . . . . . . . . . . . . . . Number 0001h  brown . . . . . . . . . . . . . Number 0006h  cyan . . . . . . . . . . . . . . Number 0003h  exit . . . . . . . . . . . . . . Text .EXIT  gray . . . . . . . . . . . . . . Number 0008h  green . . . . . . . . . . . . . Number 0002h  lightBlue . . . . . . . . . . . Number 0009h  lightCyan . . . . . . . . . . . Number 000Bh  lightGray . . . . . . . . . . . Number 0007h  lightGreen . . . . . . . . . . . Number 000Ah  lightMagenta . . . . . . . . . . Number 000Dh  lightRed . . . . . . . . . . . . Number 000Ch  magenta . . . . . . . . . . . . Number 0005h  red . . . . . . . . . . . . . . Number 0004h  white . . . . . . . . . . . . . Number 000Fh  yellow . . . . . . . . . . . . . Number 000Eh  0 Warnings  0 Errors |

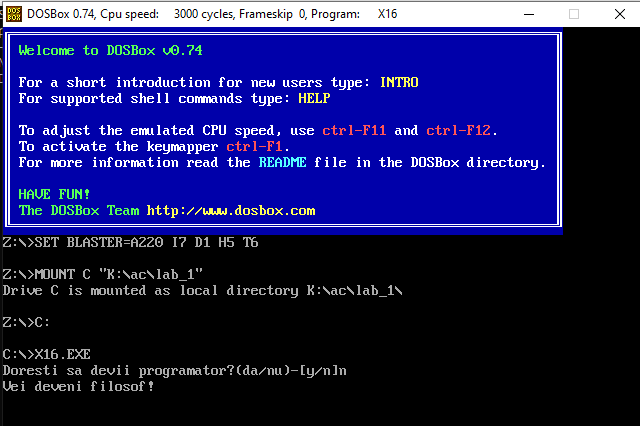
**Rularea programului 16 biți**

1. Apasam build

2. Apasam run



Rezultat:



**Codul sursa .asm 32 biți**

|  |
| --- |
| 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 ;directiva ce declara inceputul segmentului de cod  main PROC ;se indica procedura cu numele main  mov edx, OFFSET Promt  call WriteString  call ReadChar  cmp al,'y' ;compararea continutului registrului al cu codul ASCII a literei y  jz IsDad ;salt conditionat (jz-jump if zero), daca rezultatul compararii este zero, salt la eticheta IsDad  cmp al,'n' ;compararea- din al se scade codul ASCII a literei n  jz IsNud ;da, rezultatul compararii este zero, salt la eticheta IsNud  IsDad: mov edx,OFFSET Dad ;in edx offsetul sirului Dad  call WriteString  jmp ex  IsNud: mov edx,OFFSET Nud ;in edx offsetul sirului "Nud"  call WriteString  ex:  exit ;apel la procedura de iesirea din program, din fisierul Irvine32.inc  main ENDP ;sfarsitul procedurii main  END main ;finalizarea programului/ punctul de intrare in program |

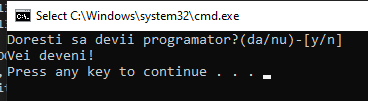
**Fișier -listing 32 biți**

|  |
| --- |
| Microsoft (R) Macro Assembler Version 12.00.40629.0 02/01/22 00:16:57  K:\ac\lab\_1\x32.asm Page 1 - 1  INCLUDE Irvine32.inc  C ; Include file for Irvine32.lib (Irvine32.inc)  C  C ;OPTION CASEMAP:NONE ; optional: make identifiers case-sensitive  C  C INCLUDE SmallWin.inc ; MS-Windows prototypes, structures, and constants  C .NOLIST  C .LIST  C  C INCLUDE VirtualKeys.inc  C ; VirtualKeys.inc  C .NOLIST  C .LIST  C  C  C .NOLIST  C .LIST  C  00000000 .data  00000000 44 6F 72 65 73 Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]', 0  74 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  0000002B 0D 0A 56 65 69 Dad DB 13, 10, 'Vei deveni!', 13, 10, 0  20 64 65 76 65  6E 69 21 0D 0A  00  0000003B 0D 0A 56 65 69 Nud DB 13, 10, 'Vei deveni filosof!', 13, 10, 0  20 64 65 76 65  6E 69 20 66 69  6C 6F 73 6F 66  21 0D 0A 00  00000000 .code ;directiva ce declara inceputul segmentului de cod  00000000 main PROC ;se indica procedura cu numele main  00000000 BA 00000000 R mov edx, OFFSET Promt  00000005 E8 00000000 E call WriteString  0000000A E8 00000000 E call ReadChar  0000000F 3C 79 cmp al,'y' ;compararea continutului registrului al cu codul ASCII a literei y  00000011 74 04 jz IsDad ;salt conditionat (jz-jump if zero), daca rezultatul compararii este zero, salt la eticheta IsDad  00000013 3C 6E cmp al,'n' ;compararea- din al se scade codul ASCII a literei n  00000015 74 0C jz IsNud ;da, rezultatul compararii este zero, salt la eticheta IsNud  00000017 BA 0000002B R IsDad: mov edx,OFFSET Dad ;in edx offsetul sirului Dad  0000001C E8 00000000 E call WriteString  00000021 EB 0A jmp ex  00000023 BA 0000003B R IsNud: mov edx,OFFSET Nud ;in edx offsetul sirului "Nud"  00000028 E8 00000000 E call WriteString  0000002D ex:  exit ;apel la procedura de iesirea din program, din fisierul Irvine32.inc  0000002D 6A 00 \* push +000000000h  0000002F E8 00000000 E \* call ExitProcess  00000034 main ENDP ;sfarsitul procedurii main  END main ;finalizarea programului/ punctul de intrare in program  Microsoft (R) Macro Assembler Version 12.00.40629.0 02/01/22 00:16:57  K:\ac\lab\_1\x32.asm Symbols 2 - 1  Structures and Unions:  N a m e Size  Offset Type  CONSOLE\_CURSOR\_INFO . . . . . . 00000008  dwSize . . . . . . . . . . . . 00000000 DWord  bVisible . . . . . . . . . . . 00000004 DWord  CONSOLE\_SCREEN\_BUFFER\_INFO . . . 00000016  dwSize . . . . . . . . . . . . 00000000 DWord  dwCursorPosition . . . . . . . 00000004 DWord  wAttributes . . . . . . . . . 00000008 Word  srWindow . . . . . . . . . . . 0000000A QWord  dwMaximumWindowSize . . . . . 00000012 DWord  COORD . . . . . . . . . . . . . 00000004  X . . . . . . . . . . . . . . 00000000 Word  Y . . . . . . . . . . . . . . 00000002 Word  FILETIME . . . . . . . . . . . . 00000008  loDateTime . . . . . . . . . . 00000000 DWord  hiDateTime . . . . . . . . . . 00000004 DWord  FOCUS\_EVENT\_RECORD . . . . . . . 00000004  bSetFocus . . . . . . . . . . 00000000 DWord  FPU\_ENVIRON . . . . . . . . . . 0000001C  controlWord . . . . . . . . . 00000000 Word  statusWord . . . . . . . . . . 00000004 Word  tagWord . . . . . . . . . . . 00000008 Word  instrPointerOffset . . . . . . 0000000C DWord  instrPointerSelector . . . . . 00000010 DWord  operandPointerOffset . . . . . 00000014 DWord  operandPointerSelector . . . . 00000018 Word  INPUT\_RECORD . . . . . . . . . . 00000014  EventType . . . . . . . . . . 00000000 Word  Event . . . . . . . . . . . . 00000004 XmmWord  bKeyDown . . . . . . . . . . . 00000000 DWord  wRepeatCount . . . . . . . . . 00000004 Word  wVirtualKeyCode . . . . . . . 00000006 Word  wVirtualScanCode . . . . . . . 00000008 Word  uChar . . . . . . . . . . . . 0000000A Word  UnicodeChar . . . . . . . . . 00000000 Word  AsciiChar . . . . . . . . . . 00000000 Byte  dwControlKeyState . . . . . . 0000000C DWord  dwMousePosition . . . . . . . 00000000 DWord  dwButtonState . . . . . . . . 00000004 DWord  dwMouseControlKeyState . . . . 00000008 DWord  dwEventFlags . . . . . . . . . 0000000C DWord  dwSize . . . . . . . . . . . . 00000000 DWord  dwCommandId . . . . . . . . . 00000000 DWord  bSetFocus . . . . . . . . . . 00000000 DWord  KEY\_EVENT\_RECORD . . . . . . . . 00000010  bKeyDown . . . . . . . . . . . 00000000 DWord  wRepeatCount . . . . . . . . . 00000004 Word  wVirtualKeyCode . . . . . . . 00000006 Word  wVirtualScanCode . . . . . . . 00000008 Word  uChar . . . . . . . . . . . . 0000000A Word  UnicodeChar . . . . . . . . . 00000000 Word  AsciiChar . . . . . . . . . . 00000000 Byte  dwControlKeyState . . . . . . 0000000C DWord  MENU\_EVENT\_RECORD . . . . . . . 00000004  dwCommandId . . . . . . . . . 00000000 DWord  MOUSE\_EVENT\_RECORD . . . . . . . 00000010  dwMousePosition . . . . . . . 00000000 DWord  dwButtonState . . . . . . . . 00000004 DWord  dwMouseControlKeyState . . . . 00000008 DWord  dwEventFlags . . . . . . . . . 0000000C DWord  SMALL\_RECT . . . . . . . . . . . 00000008  Left . . . . . . . . . . . . . 00000000 Word  Top . . . . . . . . . . . . . 00000002 Word  Right . . . . . . . . . . . . 00000004 Word  Bottom . . . . . . . . . . . . 00000006 Word  SYSTEMTIME . . . . . . . . . . . 00000010  wYear . . . . . . . . . . . . 00000000 Word  wMonth . . . . . . . . . . . . 00000002 Word  wDayOfWeek . . . . . . . . . . 00000004 Word  wDay . . . . . . . . . . . . . 00000006 Word  wHour . . . . . . . . . . . . 00000008 Word  wMinute . . . . . . . . . . . 0000000A Word  wSecond . . . . . . . . . . . 0000000C Word  wMilliseconds . . . . . . . . 0000000E Word  WINDOW\_BUFFER\_SIZE\_RECORD . . . 00000004  dwSize . . . . . . . . . . . . 00000000 DWord  Segments and Groups:  N a m e Size Length Align Combine Class  FLAT . . . . . . . . . . . . . . GROUP  STACK . . . . . . . . . . . . . 32 Bit 00001000 Para Stack 'STACK'  \_DATA . . . . . . . . . . . . . 32 Bit 00000053 Para Public 'DATA'  \_TEXT . . . . . . . . . . . . . 32 Bit 00000034 Para Public 'CODE'  Procedures, parameters, and locals:  N a m e Type Value Attr  CloseFile . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  CloseHandle . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Clrscr . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  CreateFileA . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  CreateOutputFile . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Crlf . . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Delay . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  DumpMem . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  DumpRegs . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ExitProcess . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  FileTimeToDosDateTime . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  FileTimeToSystemTime . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  FlushConsoleInputBuffer . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  FormatMessageA . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetCommandLineA . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetCommandTail . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetConsoleCP . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetConsoleCursorInfo . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetConsoleMode . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetConsoleScreenBufferInfo . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetDateTime . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetFileTime . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetKeyState . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetLastError . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetLocalTime . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetMaxXY . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetMseconds . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetNumberOfConsoleInputEvents . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetProcessHeap . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetStdHandle . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetSystemTime . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetTextColor . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  GetTickCount . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Gotoxy . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  HeapAlloc . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  HeapCreate . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  HeapDestroy . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  HeapFree . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  HeapSize . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  IsDigit . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  LocalFree . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  MessageBoxA . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  MsgBoxAsk . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  MsgBox . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  OpenInputFile . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ParseDecimal32 . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ParseInteger32 . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  PeekConsoleInputA . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Random32 . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  RandomRange . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Randomize . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadChar . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadConsoleA . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadConsoleInputA . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadDec . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadFile . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadFloat . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadFromFile . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadHex . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadInt . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadKeyFlush . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadKey . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ReadString . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetConsoleCursorInfo . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetConsoleCursorPosition . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetConsoleMode . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetConsoleScreenBufferSize . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetConsoleTextAttribute . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetConsoleTitleA . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetConsoleWindowInfo . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetFilePointer . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetLocalTime . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SetTextColor . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  ShowFPUStack . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Sleep . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  StrLength . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Str\_compare . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Str\_copy . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Str\_length . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Str\_trim . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  Str\_ucase . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  SystemTimeToFileTime . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WaitMsg . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteBinB . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteBin . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteChar . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteConsoleA . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteConsoleOutputAttribute . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteConsoleOutputCharacterA . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteDec . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteFile . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteFloat . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteHexB . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteHex . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteInt . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteStackFrameName . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteStackFrame . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteString . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteToFile . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  WriteWindowsMsg . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL  main . . . . . . . . . . . . . . P Near 00000000 \_TEXT Length= 00000034 Public STDCALL  IsDad . . . . . . . . . . . . L Near 00000017 \_TEXT  IsNud . . . . . . . . . . . . L Near 00000023 \_TEXT  ex . . . . . . . . . . . . . . L Near 0000002D \_TEXT  printf . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External C  scanf . . . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External C  wsprintfA . . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External C  Symbols:  N a m e Type Value Attr  @CodeSize . . . . . . . . . . . Number 00000000h  @DataSize . . . . . . . . . . . Number 00000000h  @Interface . . . . . . . . . . . Number 00000003h  @Model . . . . . . . . . . . . . Number 00000007h  @code . . . . . . . . . . . . . Text \_TEXT  @data . . . . . . . . . . . . . Text FLAT  @fardata? . . . . . . . . . . . Text FLAT  @fardata . . . . . . . . . . . . Text FLAT  @stack . . . . . . . . . . . . . Text FLAT  ALT\_MASK . . . . . . . . . . . . Number 00000003h  CAPSLOCK\_ON . . . . . . . . . . Number 00000080h  CREATE\_ALWAYS . . . . . . . . . Number 00000002h  CREATE\_NEW . . . . . . . . . . . Number 00000001h  CTRL\_MASK . . . . . . . . . . . Number 0000000Ch  CreateFile . . . . . . . . . . . Text CreateFileA  DO\_NOT\_SHARE . . . . . . . . . . Number 00000000h  Dad . . . . . . . . . . . . . . Byte 0000002B \_DATA  ENABLE\_ECHO\_INPUT . . . . . . . Number 00000004h  ENABLE\_LINE\_INPUT . . . . . . . Number 00000002h  ENABLE\_MOUSE\_INPUT . . . . . . . Number 00000010h  ENABLE\_PROCESSED\_INPUT . . . . . Number 00000001h  ENABLE\_PROCESSED\_OUTPUT . . . . Number 00000001h  ENABLE\_WINDOW\_INPUT . . . . . . Number 00000008h  ENABLE\_WRAP\_AT\_EOL\_OUTPUT . . . Number 00000002h  ENHANCED\_KEY . . . . . . . . . . Number 00000100h  FALSE . . . . . . . . . . . . . Number 00000000h  FILE\_APPEND\_DATA . . . . . . . . Number 00000004h  FILE\_ATTRIBUTE\_ARCHIVE . . . . . Number 00000020h  FILE\_ATTRIBUTE\_COMPRESSED . . . Number 00000800h  FILE\_ATTRIBUTE\_DEVICE . . . . . Number 00000040h  FILE\_ATTRIBUTE\_DIRECTORY . . . . Number 00000010h  FILE\_ATTRIBUTE\_ENCRYPTED . . . . Number 00004000h  FILE\_ATTRIBUTE\_HIDDEN . . . . . Number 00000002h  FILE\_ATTRIBUTE\_NORMAL . . . . . Number 00000080h  FILE\_ATTRIBUTE\_NOT\_CONTENT\_INDEXED . Number 00002000h  FILE\_ATTRIBUTE\_OFFLINE . . . . . Number 00001000h  FILE\_ATTRIBUTE\_READONLY . . . . Number 00000001h  FILE\_ATTRIBUTE\_REPARSE\_POINT . . Number 00000400h  FILE\_ATTRIBUTE\_SPARSE\_FILE . . . Number 00000200h  FILE\_ATTRIBUTE\_SYSTEM . . . . . Number 00000004h  FILE\_ATTRIBUTE\_TEMPORARY . . . . Number 00000100h  FILE\_BEGIN . . . . . . . . . . . Number 00000000h  FILE\_CURRENT . . . . . . . . . . Number 00000001h  FILE\_DELETE\_CHILD . . . . . . . Number 00000040h  FILE\_END . . . . . . . . . . . . Number 00000002h  FILE\_READ\_DATA . . . . . . . . . Number 00000001h  FILE\_SHARE\_DELETE . . . . . . . Number 00000004h  FILE\_SHARE\_READ . . . . . . . . Number 00000001h  FILE\_SHARE\_WRITE . . . . . . . . Number 00000002h  FILE\_WRITE\_DATA . . . . . . . . Number 00000002h  FOCUS\_EVENT . . . . . . . . . . Number 00000010h  FORMAT\_MESSAGE\_ALLOCATE\_BUFFER . Number 00000100h  FORMAT\_MESSAGE\_FROM\_SYSTEM . . . Number 00001000h  FormatMessage . . . . . . . . . Text FormatMessageA  GENERIC\_ALL . . . . . . . . . . Number 10000000h  GENERIC\_EXECUTE . . . . . . . . Number 20000000h  GENERIC\_READ . . . . . . . . . . Number -80000000h  GENERIC\_WRITE . . . . . . . . . Number 40000000h  GetCommandLine . . . . . . . . . Text GetCommandLineA  HANDLE . . . . . . . . . . . . . Text DWORD  HEAP\_GENERATE\_EXCEPTIONS . . . . Number 00000004h  HEAP\_GROWABLE . . . . . . . . . Number 00000002h  HEAP\_NO\_SERIALIZE . . . . . . . Number 00000001h  HEAP\_REALLOC\_IN\_PLACE\_ONLY . . . Number 00000010h  HEAP\_ZERO\_MEMORY . . . . . . . . Number 00000008h  IDABORT . . . . . . . . . . . . Number 00000003h  IDCANCEL . . . . . . . . . . . . Number 00000002h  IDCLOSE . . . . . . . . . . . . Number 00000008h  IDCONTINUE . . . . . . . . . . . Number 0000000Bh  IDHELP . . . . . . . . . . . . . Number 00000009h  IDIGNORE . . . . . . . . . . . . Number 00000005h  IDNO . . . . . . . . . . . . . . Number 00000007h  IDOK . . . . . . . . . . . . . . Number 00000001h  IDRETRY . . . . . . . . . . . . Number 00000004h  IDTIMEOUT . . . . . . . . . . . Number 00007D00h  IDTRYAGAIN . . . . . . . . . . . Number 0000000Ah  IDYES . . . . . . . . . . . . . Number 00000006h  INVALID\_HANDLE\_VALUE . . . . . . Number -00000001h  KBDOWN\_FLAG . . . . . . . . . . Number 00000001h  KEY\_EVENT . . . . . . . . . . . Number 00000001h  KEY\_MASKS . . . . . . . . . . . Number 0000001Fh  LEFT\_ALT\_PRESSED . . . . . . . . Number 00000002h  LEFT\_CTRL\_PRESSED . . . . . . . Number 00000008h  MB\_ABORTRETRYIGNORE . . . . . . Number 00000002h  MB\_APPLMODAL . . . . . . . . . . Number 00000000h  MB\_CANCELTRYCONTINUE . . . . . . Number 00000006h  MB\_DEFBUTTON1 . . . . . . . . . Number 00000000h  MB\_DEFBUTTON2 . . . . . . . . . Number 00000100h  MB\_DEFBUTTON3 . . . . . . . . . Number 00000200h  MB\_DEFBUTTON4 . . . . . . . . . Number 00000300h  MB\_HELP . . . . . . . . . . . . Number 00004000h  MB\_ICONASTERISK . . . . . . . . Number 00000040h  MB\_ICONERROR . . . . . . . . . . Number 00000010h  MB\_ICONEXCLAMATION . . . . . . . Number 00000030h  MB\_ICONHAND . . . . . . . . . . Number 00000010h  MB\_ICONINFORMATION . . . . . . . Number 00000040h  MB\_ICONQUESTION . . . . . . . . Number 00000020h  MB\_ICONSTOP . . . . . . . . . . Number 00000010h  MB\_ICONWARNING . . . . . . . . . Number 00000030h  MB\_OKCANCEL . . . . . . . . . . Number 00000001h  MB\_OK . . . . . . . . . . . . . Number 00000000h  MB\_RETRYCANCEL . . . . . . . . . Number 00000005h  MB\_SYSTEMMODAL . . . . . . . . . Number 00001000h  MB\_TASKMODAL . . . . . . . . . . Number 00002000h  MB\_USERICON . . . . . . . . . . Number 00000080h  MB\_YESNOCANCEL . . . . . . . . . Number 00000003h  MB\_YESNO . . . . . . . . . . . . Number 00000004h  MENU\_EVENT . . . . . . . . . . . Number 00000008h  MOUSE\_EVENT . . . . . . . . . . Number 00000002h  MessageBox . . . . . . . . . . . Text MessageBoxA  NULL . . . . . . . . . . . . . . Number 00000000h  NUMLOCK\_ON . . . . . . . . . . . Number 00000020h  Nud . . . . . . . . . . . . . . Byte 0000003B \_DATA  OPEN\_ALWAYS . . . . . . . . . . Number 00000004h  OPEN\_EXISTING . . . . . . . . . Number 00000003h  PeekConsoleInput . . . . . . . . Text PeekConsoleInputA  Promt . . . . . . . . . . . . . Byte 00000000 \_DATA  RIGHT\_ALT\_PRESSED . . . . . . . Number 00000001h  RIGHT\_CTRL\_PRESSED . . . . . . . Number 00000004h  ReadConsoleInput . . . . . . . . Text ReadConsoleInputA  ReadConsole . . . . . . . . . . Text ReadConsoleA  SCROLLLOCK\_ON . . . . . . . . . Number 00000040h  SHIFT\_MASK . . . . . . . . . . . Number 00000010h  SHIFT\_PRESSED . . . . . . . . . Number 00000010h  STD\_ERROR\_HANDLE . . . . . . . . Number -0000000Ch  STD\_INPUT\_HANDLE . . . . . . . . Number -0000000Ah  STD\_OUTPUT\_HANDLE . . . . . . . Number -0000000Bh  SetConsoleTitle . . . . . . . . Text SetConsoleTitleA  TAB . . . . . . . . . . . . . . Number 00000009h  TRUE . . . . . . . . . . . . . . Number 00000001h  TRUNCATE\_EXISTING . . . . . . . Number 00000005h  VK\_11 . . . . . . . . . . . . . Number 000000BDh  VK\_12 . . . . . . . . . . . . . Number 000000BBh  VK\_ADD . . . . . . . . . . . . . Number 0000006Bh  VK\_BACK . . . . . . . . . . . . Number 00000008h  VK\_CANCEL . . . . . . . . . . . Number 00000003h  VK\_CAPITAL . . . . . . . . . . . Number 00000014h  VK\_CLEAR . . . . . . . . . . . . Number 0000000Ch  VK\_CONTROL . . . . . . . . . . . Number 00000011h  VK\_DECIMAL . . . . . . . . . . . Number 0000006Eh  VK\_DELETE . . . . . . . . . . . Number 0000002Eh  VK\_DIVIDE . . . . . . . . . . . Number 0000006Fh  VK\_DOWN . . . . . . . . . . . . Number 00000028h  VK\_END . . . . . . . . . . . . . Number 00000023h  VK\_ESCAPE . . . . . . . . . . . Number 0000001Bh  VK\_EXECUTE . . . . . . . . . . . Number 0000002Bh  VK\_F10 . . . . . . . . . . . . . Number 00000079h  VK\_F11 . . . . . . . . . . . . . Number 0000007Ah  VK\_F12 . . . . . . . . . . . . . Number 0000007Bh  VK\_F13 . . . . . . . . . . . . . Number 0000007Ch  VK\_F14 . . . . . . . . . . . . . Number 0000007Dh  VK\_F15 . . . . . . . . . . . . . Number 0000007Eh  VK\_F16 . . . . . . . . . . . . . Number 0000007Fh  VK\_F17 . . . . . . . . . . . . . Number 00000080h  VK\_F18 . . . . . . . . . . . . . Number 00000081h  VK\_F19 . . . . . . . . . . . . . Number 00000082h  VK\_F1 . . . . . . . . . . . . . Number 00000070h  VK\_F20 . . . . . . . . . . . . . Number 00000083h  VK\_F21 . . . . . . . . . . . . . Number 00000084h  VK\_F22 . . . . . . . . . . . . . Number 00000085h  VK\_F23 . . . . . . . . . . . . . Number 00000086h  VK\_F24 . . . . . . . . . . . . . Number 00000087h  VK\_F2 . . . . . . . . . . . . . Number 00000071h  VK\_F3 . . . . . . . . . . . . . Number 00000072h  VK\_F4 . . . . . . . . . . . . . Number 00000073h  VK\_F5 . . . . . . . . . . . . . Number 00000074h  VK\_F6 . . . . . . . . . . . . . Number 00000075h  VK\_F7 . . . . . . . . . . . . . Number 00000076h  VK\_F8 . . . . . . . . . . . . . Number 00000077h  VK\_F9 . . . . . . . . . . . . . Number 00000078h  VK\_HELP . . . . . . . . . . . . Number 0000002Fh  VK\_HOME . . . . . . . . . . . . Number 00000024h  VK\_INSERT . . . . . . . . . . . Number 0000002Dh  VK\_LBUTTON . . . . . . . . . . . Number 00000001h  VK\_LCONTROL . . . . . . . . . . Number 000000A2h  VK\_LEFT . . . . . . . . . . . . Number 00000025h  VK\_LMENU . . . . . . . . . . . . Number 000000A4h  VK\_LSHIFT . . . . . . . . . . . Number 000000A0h  VK\_MENU . . . . . . . . . . . . Number 00000012h  VK\_MULTIPLY . . . . . . . . . . Number 0000006Ah  VK\_NEXT . . . . . . . . . . . . Number 00000022h  VK\_NUMLOCK . . . . . . . . . . . Number 00000090h  VK\_NUMPAD0 . . . . . . . . . . . Number 00000060h  VK\_NUMPAD1 . . . . . . . . . . . Number 00000061h  VK\_NUMPAD2 . . . . . . . . . . . Number 00000062h  VK\_NUMPAD3 . . . . . . . . . . . Number 00000063h  VK\_NUMPAD4 . . . . . . . . . . . Number 00000064h  VK\_NUMPAD5 . . . . . . . . . . . Number 00000065h  VK\_NUMPAD6 . . . . . . . . . . . Number 00000066h  VK\_NUMPAD7 . . . . . . . . . . . Number 00000067h  VK\_NUMPAD8 . . . . . . . . . . . Number 00000068h  VK\_NUMPAD9 . . . . . . . . . . . Number 00000069h  VK\_PAUSE . . . . . . . . . . . . Number 00000013h  VK\_PRINT . . . . . . . . . . . . Number 0000002Ah  VK\_PRIOR . . . . . . . . . . . . Number 00000021h  VK\_RBUTTON . . . . . . . . . . . Number 00000002h  VK\_RCONTROL . . . . . . . . . . Number 000000A3h  VK\_RETURN . . . . . . . . . . . Number 0000000Dh  VK\_RIGHT . . . . . . . . . . . . Number 00000027h  VK\_RMENU . . . . . . . . . . . . Number 000000A5h  VK\_RSHIFT . . . . . . . . . . . Number 000000A1h  VK\_SCROLL . . . . . . . . . . . Number 00000091h  VK\_SEPARATER . . . . . . . . . . Number 0000006Ch  VK\_SHIFT . . . . . . . . . . . . Number 00000010h  VK\_SNAPSHOT . . . . . . . . . . Number 0000002Ch  VK\_SPACE . . . . . . . . . . . . Number 00000020h  VK\_SUBTRACT . . . . . . . . . . Number 0000006Dh  VK\_TAB . . . . . . . . . . . . . Number 00000009h  VK\_UP . . . . . . . . . . . . . Number 00000026h  WINDOW\_BUFFER\_SIZE\_EVENT . . . . Number 00000004h  WriteConsoleOutputCharacter . . Text WriteConsoleOutputCharacterA  WriteConsole . . . . . . . . . . Text WriteConsoleA  black . . . . . . . . . . . . . Number 00000000h  blue . . . . . . . . . . . . . . Number 00000001h  brown . . . . . . . . . . . . . Number 00000006h  cyan . . . . . . . . . . . . . . Number 00000003h  exit . . . . . . . . . . . . . . Text INVOKE ExitProcess,0  gray . . . . . . . . . . . . . . Number 00000008h  green . . . . . . . . . . . . . Number 00000002h  lightBlue . . . . . . . . . . . Number 00000009h  lightCyan . . . . . . . . . . . Number 0000000Bh  lightGray . . . . . . . . . . . Number 00000007h  lightGreen . . . . . . . . . . . Number 0000000Ah  lightMagenta . . . . . . . . . . Number 0000000Dh  lightRed . . . . . . . . . . . . Number 0000000Ch  magenta . . . . . . . . . . . . Number 00000005h  red . . . . . . . . . . . . . . Number 00000004h  white . . . . . . . . . . . . . Number 0000000Fh  wsprintf . . . . . . . . . . . . Text wsprintfA  yellow . . . . . . . . . . . . . Number 0000000Eh  0 Warnings  0 Errors |

**Rularea programului 32 biți**

1. Apasam butonul **Start Without Debugging**





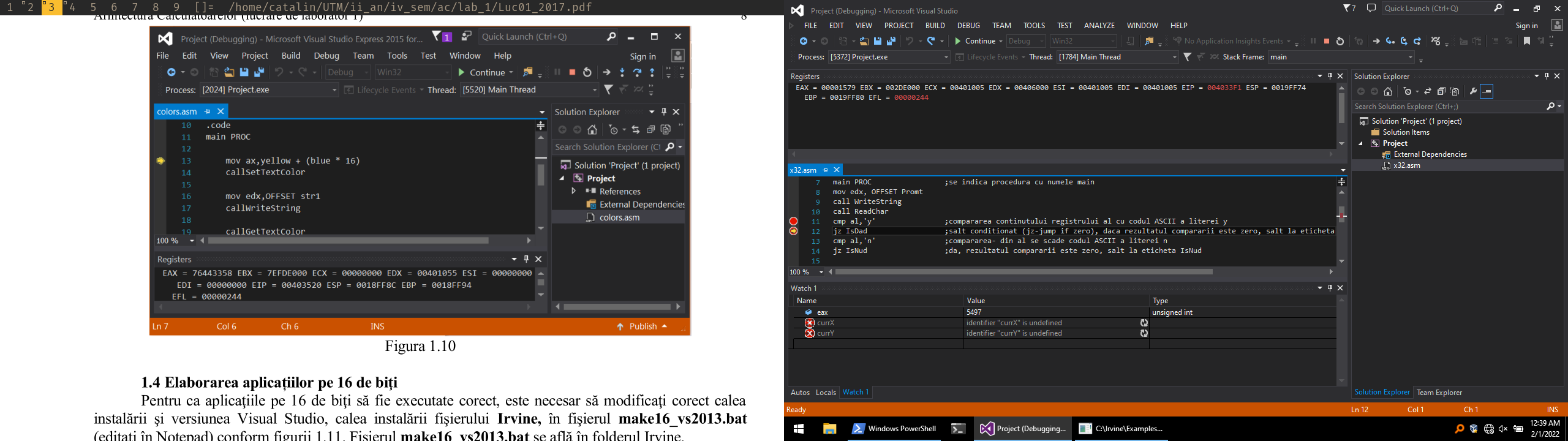
**Pașii rularii în debug**

1. Cream niște breaking points;

2. Apasam tasta F10 pentru a rula programul în debug;

3. Apasam tasta F5 pentru a continua la următorii pași;

4. În fereastra Registri și Watch1 putem vedea valorile registrilor și cum acestea se modifica.

****

**Concluzii:**

Efectuarea lucrării a dus la familiarizarea cu regiştrii microprocesoarelor pe 16 și 32 de biți, cu mediul de dezvoltare a programelor Visual Studio. A fost setat mediul Visual Studio ca sa poată rula programe MASM pe 16 și 32 biți.

Am încercat să evit mediul de dezvoltare Visual Studio, însă am eșuat și se pare că pentru următoarele laboratoare voi fi nevoit să scriu cod în Visual Studio pe Windows într-o mașina virtuală.

Am observat că deși programele fac același lucru, versia pe 32 de biți crează un executabil de 3 ori mai mare decât versia de 16 biți. Ceea ce înseamnă că în unele cazuri când resursele disponibilie sunt limitate este de dorit să scrim cod pentru 16 biți deși codul în sine este mai mare.