

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Организация ЭВМ и систем»
Тема: Изучение режимов адресации и формирования исполнительного
адреса.

Студент гр. 0383

Куликов А. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучить режимы адресации и формирования исполнительного адреса.

Задание.

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу `lr2_comp.asm` на Ассемблере, которая в автоматическом режиме выполняться не должна, так как не имеет самостоятельного функционального назначения, а только тестирует режимы адресации. Поэтому ее выполнение должно производиться под управлением отладчика в пошаговом режиме.

В программу введен ряд ошибок, которые необходимо объяснить в отчете по работе, а соответствующие команды закомментировать для прохождения трансляции. Необходимо составить протокол выполнения программы в пошаговом режиме отладчика по типу таблицы 1 предыдущей лабораторной работы и подписать его у преподавателя. На защите студенты должны уметь объяснить результат выполнения каждой команды с учетом используемого вида адресации. Результаты, полученные с помощью отладчика, не являются объяснением, а только должны подтверждать ваши объяснения.

Выполнение работы.

Вар. 9

`vec1 31,32,33,34,38,37,36,35`

`vec2 50,60,-50,-60,70,80,-70,-80`

`matr -4,-3,7,8,-2,-1,5,6,-8,-7,3,4,-6,-5,1,2`

1. Программа была протранслирована со следующими ошибками:

1. `mov mem3, [bx] lab2.asm(42): error A2052: Improper operand type.`

Попытка переместить данные из одной ячейки памяти в другую. Перемещать данные можно только между регистрами или регистрами и ячейками памяти.

2. `mov cx,vec2[di] ab2.asm(49): warning A4031: Operand types must match.` Попытка поместить данные размером в 1 байт в регистр размером 2 байта.

3. `mov cx,matr[bx][di] lab2.asm(53): warning A4031: Operand types must match.` Попытка поместить данные размером в 1 байт в регистр размером 2 байта.
4. `mov ax,matr[bx*4][di] lab2.asm(54): error A2055: Illegal register value.` Недопустимое значение регистра.
5. `mov ax,matr[bp+bx] lab2.asm(73): error A2046: Multiple base registers.` Попытка использовать несколько базовых регистров для адресации.
6. `mov ax,matr[bp+di+si] lab2.asm(74): error A2047: Multiple index registers.` Попытка использовать несколько индексных регистров для адресации.

2. Строки с ошибками были закомментированы, новый файл `lab2_fix.asm` протранслирован без ошибок, создан диагностический файл `lab2_fix.lst` и объектный файл `lab2_fix.obj`. `lab2_fix.obj` был слинкован без ошибок, получен исполняемый файл `lab2_fix.exe`. Он был запущен в отладчике `afdbg`.

(CS) = 1A0A

(DS) = 19F5

(ES) = 19F5

(SS) = 1A05

(CX) = 00B0

(BP) = 0000

(DX) = 0000

Табл. 1

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения.	После выполнения.
0000	PUSH DS	1E	(AX) = 0000 (DS) = 19F5 (IP) = 0000 (SP) = 0018 STACK+0 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0001 (SP) = 0016 STACK+0 19F5

0001	SUB AX, AX	2BC0	(AX) = 0000 (DS) = 19F5 (IP) = 0001	(AX) = 0000 (DS) = 19F5 (IP) = 0003
0003	PUSH AX	50	(AX) = 0000 (DS) = 19F5 (IP) = 0003 (SP) = 0016 STACK+0 19F5	(AX) = 0000 (DS) = 19F5 (IP) = 0004 (SP) = 0014 STACK+0 0000 STACK+2 19F5
0004	MOV AX, 1A07	B0871A	(AX) = 0000 (DS) = 19F5 (IP) = 0004	(AX) = 1A07 (DS) = 19F5 (IP) = 0007
0007	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 0007	(AX) = 1A07 (DS) = 1A07 (IP) = 0009
0009	MOV AX, 01F4	B8F401	(AX) = 1A07 (DS) = 1A07 (IP) = 0009	(AX) = 01F4 (DS) = 1A07 (IP) = 000C
000C	MOV CX, AX	8BC8	(AX) = 01F4 (DS) = 1A07 (CX) = 00B0 (IP) = 000C	(AX) = 01F4 (DS) = 1A07 (CX) = 01F4 (IP) = 000E
000E	MOV BL, 24	B324	(AX) = 01F4 (DS) = 1A07 (BX) = 0000 (IP) = 000E	(AX) = 01F4 (DS) = 1A07 (BX) = 0024 (IP) = 0010
0010	MOV BH, CE	B7CE	(AX) = 01F4 (DS) = 1A07 (BX) = 0024 (IP) = 0010	(AX) = 01F4 (DS) = 1A07 (BX) = CE24 (IP) = 0012
0012	MOV [0002], FFCE	C7060200CE FF	(AX) = 1A07 (DS) = 1A07 (IP) = 0012 DATA SEG +2	(AX) = 1A07 (DS) = 1A07 (IP) = 0018 DATA SEG +2

			0000	CEFF
0018	MOV BX, 0006	BB0600	(AX) = 1A07 (DS) = 1A07 (BX) = CE24 (IP) = 0018	(AX) = 1A07 (DS) = 1A07 (BX) = 0006 (IP) = 001B
001B	MOV [0000], AX	A30000	(AX) = 01F4 (DS) = 1A07 (IP) = 001B DATA SEG +0 0000	(AX) = 01F4 (DS) = 1A07 (IP) = 001E DATA SEG +0 F401
001E	MOV AL, [BX]	8A07	(AX) = 01F4 (DS) = 1A07 (IP) = 001E	(AX) = 011F (DS) = 1A07 (IP) = 0020
0020	MOV AX, [BX+03]	8A4703	(AX) = 011F (DS) = 1A07 (IP) = 0020	(AX) = 0122 (DS) = 1A07 (IP) = 0023
0023	MOV CX, [BX+03]	8B4F03	(AX) = 0122 (DS) = 1A07 (CX) = 01F4 (IP) = 0023	(AX) = 0122 (DS) = 1A07 (CX) = 2622 (IP) = 0026
0026	MOV DI, 0002	BF0200	(AX) = 0122 (DS) = 1A07 (DI) = 0000 (IP) = 0026	(AX) = 0122 (DS) = 1A07 (DI) = 0002 (IP) = 0029
0029	MOV AL, [000E+DI]	8A850E00	(AX) = 0122 (DS) = 1A07 (DI) = 0002 (IP) = 0029	(AX) = 01CE (DS) = 1A07 (DI) = 0002 (IP) = 002D
002D	MOV BX, 0003	BB0300	(AX) = 01CE (DS) = 1A07 (BX) = 0006 (IP) = 002D	(AX) = 01CE (DS) = 1A07 (BX) = 0003 (IP) = 0030

0030	MOV AL, [0016+BX+DI]	8A811600	(AX) = 01CE (DS) = 1A07 (BX) = 0003 (DI) = 0002 (IP) = 0030	(AX) = 01FF (DS) = 1A07 (BX) = 0003 (DI) = 0002 (IP) = 0034
0034	MOV AX, 1A07	B8071A	(AX) = 01FF (DS) = 1A07 (IP) = 0034	(AX) = 1A07 (DS) = 1A07 (IP) = 0037
0037	MOV ES, AX	8EC0	(AX) = 1A07 (DS) = 1A07 (ES) = 19F5 (IP) = 0037	(AX) = 1A07 (DS) = 1A07 (ES) = 1A07 (IP) = 0039
0039	MOV AX, ES:[BX]	268B07	(AX) = 1A07 (DS) = 1A07 (ES) = 1A07 (IP) = 0039	(AX) = 00FF (DS) = 1A07 (ES) = 1A07 (IP) = 003C
003C	MOV AX, 0000	B80000	(AX) = 00FF (DS) = 1A07 (IP) = 003C	(AX) = 0000 (DS) = 1A07 (IP) = 003F
003F	MOV ES, AX	8EC0	(AX) = 0000 (DS) = 1A07 (ES) = 1A07 (IP) = 003F	(AX) = 0000 (DS) = 1A07 (ES) = 0000 (IP) = 0041
0041	PUSH DS	1E	(AX) = 0000 (DS) = 1A07 (IP) = 0041 (SP) = 0014 STACK +0 0000	(AX) = 0000 (DS) = 1A07 (IP) = 0042 (SP) = 0012 STACK +0 1A07
0042	POP ES	07	(AX) = 0000 (DS) = 1A07 (IP) = 0042 (SP) = 0012 (ES) = 0000 STACK +0 1A07	(AX) = 0000 (DS) = 1A07 (IP) = 0043 (SP) = 0014 (ES) = 1A07 STACK +0 0000

0043	MOV CX, ES:[BX-01]	268B4FFF	(AX) = 0000 (DS) = 1A07 (IP) = 0043 (ES) = 1A07 (BX) = 0003 (CX) = 2622	(AX) = 0000 (DS) = 1A07 (IP) = 0047 (ES) = 1A07 (BX) = 0003 (CX) = FFCE
0047	XCHG AX, CX	91	(AX) = 0000 (DS) = 1A07 (IP) = 0047 (CX) = FFCE	(AX) = FFCE (DS) = 1A07 (IP) = 0048 (CX) = 0000
0048	MOV DI, 0002	BF0200	(AX) = FFCE (DS) = 1A07 (IP) = 0048 (DI) = 0002	(AX) = FFCE (DS) = 1A07 (IP) = 004B (DI) = 0002
004B	MOV ES:[BX+DI], AX	268901	(AX) = FFCE (DS) = 1A07 (IP) = 004B (DI) = 0002 (BX) = 0003 (ES) = 1A07 DATA SEG +5 00 1F	(AX) = FFCE (DS) = 1A07 (IP) = 004E (DI) = 0002 (BX) = 0003 (ES) = 1A07 DATA SEG +5 CE FF
004E	MOV BP, SP	8BEC	(AX) = FFCE (DS) = 1A07 (IP) = 004E (BP) = 0000 (SP) = 0014	(AX) = FFCE (DS) = 1A07 (IP) = 0050 (BP) = 0014 (SP) = 0014
0050	PUSH [0000]	FF360000	(AX) = 0000 (DS) = 1A07 (IP) = 0050 (SP) = 0014 STACK +0 0000	(AX) = 0000 (DS) = 1A07 (IP) = 0054 (SP) = 0012 STACK +0 01F4
0054	PUSH [0002]	FF360200	(AX) = 0000 (DS) = 1A07	(AX) = 0000 (DS) = 1A07

			(IP) = 0054 (SP) = 0012 STACK +0 01F4	(IP) = 0058 (SP) = 0010 STACK +0 FFCE
0058	MOV BP, SP	8BEC	(AX) = FFCE (DS) = 1A07 (IP) = 0058 (BP) = 0014 (SP) = 0010	(AX) = FFCE (DS) = 1A07 (IP) = 005A (BP) = 0010 (SP) = 0010
005A	MOV DX, [BP+02]	8B5602	(AX) = FFCE (DS) = 1A07 (IP) = 005A (BP) = 0010 (DX) = 0000 STACK +0 FFCE STACK +2 01F4	(AX) = FFCE (DS) = 1A07 (IP) = 005D (BP) = 0010 (DX) = 01F4 STACK +0 FFCE STACK +2 01F4
005D	RET FAR 0002	CA0200	(AX) = FFCE (DS) = 1A07 (IP) = 005D (SP) = 0010 STACK +0 FFCE STACK +2 01F4	(AX) = FFCE (DS) = 1A07 (IP) = FFCE (SP) = 0016 STACK +0 19F5

Из последней строки таблицы видно, что программа, которая должна была завершиться, не завершилась, так как на стэк были записаны ненужные данные. Чтобы программа завершалась корректно, необходимо либо закомментировать две строки, которые добавляют значения на стэк, либо дописать две строки, которые будут извлекать данные со стэка. Данная ошибка не выявляется компилятором, из-за этого не была исправлена в п. 2 данной работы.

Выводы.

В данной лабораторной работе были изучены режимы адресации и формирования исполнительного адреса.

ПРИЛОЖЕНИЕ А
Тексты исходных файлов программ

lab2.asm

; Программа изучения режимов адресации процессора IntelX86

EOL EQU '\$'

ind EQU 2

n1 EQU 500

n2 EQU -50

; Стек программы

AStack SEGMENT STACK

 DW 12 DUP(?)

AStack ENDS

; Данные программы

DATA SEGMENT

; Директивы описания данных

mem1 DW 0

mem2 DW 0

mem3 DW 0

vec1 DB 31,32,33,34,38,37,36,35

vec2 DB 50,60,-50,-60,70,80,-70,-80

matr DB -4,-3,7,8,-2,-1,5,6,-8,-7,3,4,-6,-5,1,2

DATA ENDS

; Код программы

CODE SEGMENT

 ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура

Main PROC FAR

 push DS

 sub AX,AX

```

    push AX
    mov AX,DATA
    mov DS,AX
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ СМЕЩЕНИЙ
; Регистровая адресация
    mov ax,n1
    mov cx,ax
    mov bl,EOL
    mov bh,n2
; Прямая адресация
    mov mem2,n2
    mov bx,OFFSET vec1
    mov mem1,ax
; Косвенная адресация
    mov al,[bx]
    mov mem3,[bx]
; Базированная адресация
    mov al,[bx]+3
    mov cx,3[bx]
; Индексная адресация
    mov di,ind
    mov al,vec2[di]
    mov cx,vec2[di]
; Адресация с базированием и индексированием
    mov bx,3
    mov al,matr[bx][di]
    mov cx,matr[bx][di]
    mov ax,matr[bx*4][di]
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмента

```

```

; ----- вариант 1
    mov ax, SEG vec2
    mov es, ax
    mov ax, es:[bx]
    mov ax, 0
; ----- вариант 2
    mov es, ax
    push ds
    pop es
    mov cx, es:[bx-1]
    xchg cx,ax
; ----- вариант 3
    mov di,ind
    mov es:[bx+di],ax
; ----- вариант 4
    mov bp,sp
    mov ax,matr[bp+bx]
    mov ax,matr[bp+di+si]
; Использование сегмента стека
    push mem1
    push mem2
    mov bp,sp
    mov dx,[bp]+2
    ret 2

```

Main ENDP

CODE ENDS

END Main

lab2_fix.asm

; Программа изучения режимов адресации процессора IntelX86

EOL EQU '\$'

```

ind EQU 2
n1 EQU 500
n2 EQU -50
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
; Данные программы
DATA SEGMENT
; Директивы описания данных
mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 31,32,33,34,38,37,36,35
vec2 DB 50,60,-50,-60,70,80,-70,-80
matr DB -4,-3,7,8,-2,-1,5,6,-8,-7,3,4,-6,-5,1,2
DATA ENDS
; Код программы
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
; Головная процедура
Main PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ СМЕЩЕНИЙ
; Регистровая адресация
    mov ax,n1

```

```

    mov cx,ax
    mov bl,EOL
    mov bh,n2
; Прямая адресация
    mov mem2,n2
    mov bx,OFFSET vec1
    mov mem1,ax
; Косвенная адресация
    mov al,[bx]
    ;mov mem3,[bx]
; Базированная адресация
    mov al,[bx]+3
    mov cx,3[bx]
; Индексная адресация
    mov di,ind
    mov al,vec2[di]
    ;mov cx,vec2[di]
; Адресация с базированием и индексированием
    mov bx,3
    mov al,matr[bx][di]
    ;mov cx,matr[bx][di]
    ;mov ax,matr[bx*4][di]
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмента
; ----- вариант 1
    mov ax, SEG vec2
    mov es, ax
    mov ax, es:[bx]
    mov ax, 0
; ----- вариант 2

```

```

    mov es, ax
    push ds
    pop es
    mov cx, es:[bx-1]
    xchg cx,ax
; ----- вариант 3
    mov di,ind
    mov es:[bx+di],ax
; ----- вариант 4
    mov bp,sp
    ;mov ax,matr[bp+bx]
    ;mov ax,matr[bp+di+si]
; Использование сегмента стека
    push mem1
    push mem2
    mov bp,sp
    mov dx,[bp]+2
    ret 2
Main ENDP
CODE ENDS
    END Main

```

ПРИЛОЖЕНИЕ Б
Тексты файлов диагностических сообщений программ

lab2.lst

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
11:45:48

9/29/21

PAGE 1-1

```
                                ; ПРОГРАММА ИЗУЧЕНИЯ РЕЖИМОВ
АДРЕСАЦИИ ПРОЦЕССО
                                PA INTELX86
= 0024                          EOL EQU '$'
= 0002                          IND EQU 2
= 01F4                          N1 EQU 500
=-0032                          N2 EQU -50
                                ; СТЕК ПРОГРАММЫ
0000                            ASTACK SEGMENT STACK
0000 000C[                      DW 12 DUP(?)
    ????
]

0018                            ASTACK ENDS
                                ; ДАННЫЕ ПРОГРАММЫ
0000                            DATA SEGMENT
                                ; ДИРЕКТИВЫ ОПИСАНИЯ ДАННЫХ
0000 0000                      MEM1 DW 0
0002 0000                      MEM2 DW 0
0004 0000                      MEM3 DW 0
0006 1F 20 21 22 26 25        VEC1 DB 31,32,33,34,38,37,36,35
    24 23
000E 32 3C CE C4 46 50        VEC2 DB 50,60,-50,-60,70,80,-70,-80
    BA B0
0016 FC FD 07 08 FE FF        MATR DB -4,-3,7,8,-2,-1,5,6,-8,-7,3,4,-6,-5,1,2
    05 06 F8 F9 03 04
    FA FB 01 02
0026                            DATA ENDS
                                ; КОД ПРОГРАММЫ
0000                            CODE SEGMENT
                                ASSUME CS:CODE, DS:DATA, SS:ASTACK
                                ; ГОЛОВНАЯ ПРОЦЕДУРА
0000                            MAIN PROC FAR
0000 1E                        PUSH DS
```

```

0001 2B C0                SUB AX,AX
0003 50                  PUSH AX
0004 B8 ---- R          MOV AX,DATA
0007 8E D8              MOV DS,AX

```

; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ
СМЕЩЕНИЙ

```

; РЕГИСТРОВАЯ АДРЕСАЦИЯ
0009 B8 01F4            MOV AX,N1
000C 8B C8              MOV CX,AX
000E B3 24              MOV BL,EOL
0010 B7 CE              MOV BH,N2

```

```

; ПРЯМАЯ АДРЕСАЦИЯ
0012 C7 06 0002 R FFCE  MOV MEM2,N2
0018 BB 0006 R          MOV BX,OFFSET VEC1
001B A3 0000 R          MOV MEM1,AX

```

```

; КОСВЕННАЯ АДРЕСАЦИЯ
001E 8A 07              MOV AL,[BX]
                      MOV MEM3,[BX]

```

LAB2.ASM(42): ERROR A2052: IMPROPER OPERAND TYPE
; БАЗИРОВАННАЯ АДРЕСАЦИЯ

```

0020 8A 47 03            MOV AL,[BX]+3
0023 8B 4F 03            MOV CX,3[BX]

```

```

; ИНДЕКСНАЯ АДРЕСАЦИЯ
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
11:45:48

```

9/29/21

PAGE 1-2

```

0026 BF 0002            MOV DI,IND
0029 8A 85 000E R        MOV AL,VEC2[DI]
002D 8B 8D 000E R        MOV CX,VEC2[DI]

```

LAB2.ASM(49): WARNING A4031: OPERAND TYPES MUST MATCH
; АДРЕСАЦИЯ С БАЗИРОВАНИЕМ И

ИНДЕКСИРОВАНИЕМ

```

0031 BB 0003            MOV BX,3
0034 8A 81 0016 R        MOV AL,MATR[BX][DI]
0038 8B 89 0016 R        MOV CX,MATR[BX][DI]

```

LAB2.ASM(53): WARNING A4031: OPERAND TYPES MUST MATCH
003C 8B 85 0022 R MOV AX,MATR[BX*4][DI]

LAB2.ASM(54): ERROR A2055: ILLEGAL REGISTER VALUE
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ
СЕГМЕНТОВ

```

; ПЕРЕОПРЕДЕЛЕНИЕ СЕГМЕНТА
; ----- ВАРИАНТ 1

```



```

0040 B8 ---- R          MOV AX, SEG VEC2
0043 8E C0              MOV ES, AX
0045 26: 8B 07          MOV AX, ES:[BX]
0048 B8 0000            MOV AX, 0
                        ; ----- ВАРИАНТ 2
004B 8E C0              MOV ES, AX
004D 1E                PUSH DS
004E 07                POP ES
004F 26: 8B 4F FF      MOV CX, ES:[BX-1]
0053 91                XCHG CX,AX
                        ; ----- ВАРИАНТ 3
0054 BF 0002            MOV DI,IND
0057 26: 89 01          MOV ES:[BX+DI],AX
                        ; ----- ВАРИАНТ 4
005A 8B EC              MOV BP,SP
005C 3E: 8B 86 0016 R    MOV AX,MATR[BP+BX]
LAB2.ASM(73): ERROR A2046: MULTIPLE BASE REGISTERS
0061 3E: 8B 83 0016 R    MOV AX,MATR[BP+DI+SI]
LAB2.ASM(74): ERROR A2047: MULTIPLE INDEX REGISTERS
                        ; ИСПОЛЬЗОВАНИЕ СЕГМЕНТА СТЕКА
0066 FF 36 0000 R      PUSH MEM1
006A FF 36 0002 R      PUSH MEM2
006E 8B EC              MOV BP,SP
0070 8B 56 02           MOV DX,[BP]+2
0073 CA 0002            RET 2
0076                    MAIN ENDP
LAB2.ASM(81): ERROR A2006: PHASE ERROR BETWEEN PASSES
0076                    CODE ENDS
                        END MAIN

```

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
11:45:48

9/29/21

SYMBOLS-1

SEGMENTS AND GROUPS:

N A M E	LENGTH	ALIGN	COMBINE CLASS
ASTACK	0018	PARA	STACK
CODE	0076	PARA	NONE
DATA	0026	PARA	NONE

SYMBOLS:

N A M E	TYPE	VALUE	ATTR
EOL	NUMBER	0024	
IND	NUMBER	0002	
MAIN	F PROC	0000	CODE LENGTH = 0076
MATR	L BYTE	0016	DATA
MEM1	L WORD	0000	DATA
MEM2	L WORD	0002	DATA
MEM3	L WORD	0004	DATA
N1	NUMBER	01F4	
N2	NUMBER	-0032	
VEC1	L BYTE	0006	DATA
VEC2	L BYTE	000E	DATA
@CPU	TEXT	0101H	
@FILENAME	TEXT	LAB2	
@VERSION	TEXT	510	

83 SOURCE LINES
83 TOTAL LINES
19 SYMBOLS

47828 + 461479 BYTES SYMBOL SPACE FREE

2 WARNING ERRORS
5 SEVERE ERRORS

lab2_fix.lst

```

; ПРОГРАММА ИЗУЧЕНИЯ РЕЖИМОВ
АДРЕСАЦИИ ПРОЦЕССО
РА INTELX86
= 0024          EOL EQU '$'
= 0002          IND EQU 2
= 01F4          N1 EQU 500
=-0032          N2 EQU -50
; СТЕК ПРОГРАММЫ
0000            ASTACK SEGMENT STACK
0000 000C[      DW 12 DUP(?)
      ???
      ]

0018            ASTACK ENDS
; ДАННЫЕ ПРОГРАММЫ
0000            DATA SEGMENT
; ДИРЕКТИВЫ ОПИСАНИЯ ДАННЫХ
0000 0000      MEM1 DW 0
0002 0000      MEM2 DW 0
0004 0000      MEM3 DW 0
0006 1F 20 21 22 26 25  VEC1 DB 31,32,33,34,38,37,36,35
      24 23
000E 32 3C CE C4 46 50  VEC2 DB 50,60,-50,-60,70,80,-70,-80
      BA B0
0016 FC FD 07 08 FE FF  MATR DB -4,-3,7,8,-2,-1,5,6,-8,-7,3,4,-6,-5,1,2
      05 06 F8 F9 03 04
      FA FB 01 02
0026            DATA ENDS
; КОД ПРОГРАММЫ
0000            CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:ASTACK
; ГОЛОВНАЯ ПРОЦЕДУРА
0000            MAIN PROC FAR
0000 1E          PUSH DS
0001 2B C0        SUB AX,AX
0003 50          PUSH AX
0004 B8 ---- R    MOV AX,DATA
0007 8E D8        MOV DS,AX
```

		; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ
СМЕЩЕНИЙ		
		; РЕГИСТРОВАЯ АДРЕСАЦИЯ
0009 B8 01F4	MOV AX,N1	
000C 8B C8	MOV CX,AX	
000E B3 24	MOV BL,EOL	
0010 B7 CE	MOV BH,N2	
		; ПРЯМАЯ АДРЕСАЦИЯ
0012 C7 06 0002 R FFCE	MOV MEM2,N2	
0018 BB 0006 R	MOV BX,OFFSET VEC1	
001B A3 0000 R	MOV MEM1,AX	
		; КОСВЕННАЯ АДРЕСАЦИЯ
001E 8A 07	MOV AL,[BX]	
	;MOV MEM3,[BX]	
		; БАЗИРОВАННАЯ АДРЕСАЦИЯ
0020 8A 47 03	MOV AL,[BX]+3	
0023 8B 4F 03	MOV CX,3[BX]	
		; ИНДЕКСНАЯ АДРЕСАЦИЯ

```
0026 BF 0002          MOV DI,IND
0029 8A 85 000E R      MOV AL,VEC2[DI]
                      ;MOV CX,VEC2[DI]
                      ; АДРЕСАЦИЯ С БАЗИРОВАНИЕМ И
ИНДЕКСИРОВАНИЕМ
002D BB 0003          MOV BX,3
0030 8A 81 0016 R      MOV AL,MATR[BX][DI]
                      ;MOV CX,MATR[BX][DI]
                      ;MOV AX,MATR[BX*4][DI]
                      ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ
СЕГМЕНТОВ
                      ; ПЕРЕОПРЕДЕЛЕНИЕ СЕГМЕНТА
                      ; ----- ВАРИАНТ 1
0034 B8 ---- R        MOV AX, SEG VEC2
0037 8E C0             MOV ES, AX
0039 26: 8B 07         MOV AX, ES:[BX]
003C B8 0000           MOV AX, 0
                      ; ----- ВАРИАНТ 2
003F 8E C0             MOV ES, AX
0041 1E                PUSH DS
0042 07                POP ES
0043 26: 8B 4F FF      MOV CX, ES:[BX-1]
0047 91                XCHG CX,AX
                      ; ----- ВАРИАНТ 3
0048 BF 0002          MOV DI,IND
004B 26: 89 01         MOV ES:[BX+DI],AX
                      ; ----- ВАРИАНТ 4
004E 8B EC            MOV BP,SP
                      ;MOV AX,MATR[BP+BX]
                      ;MOV AX,MATR[BP+DI+SI]
                      ; ИСПОЛЬЗОВАНИЕ СЕГМЕНТА СТЕКА
0050 FF 36 0000 R      PUSH MEM1
0054 FF 36 0002 R      PUSH MEM2
0058 8B EC            MOV BP,SP
005A 8B 56 02         MOV DX,[BP]+2
005D CA 0002          RET 2
0060                  MAIN ENDP
0060                  CODE ENDS
                      END MAIN
```

SYMBOLS-1

SEGMENTS AND GROUPS:

N A M E	LENGTH	ALIGN	COMBINE CLASS
ASTACK	0018	PARA	STACK
CODE	0060	PARA	NONE
DATA	0026	PARA	NONE

SYMBOLS:

N A M E	TYPE	VALUE	ATTR
EOL	NUMBER	0024	
IND	NUMBER	0002	
MAIN	F PROC	0000	CODE LENGTH = 0060
MATR	L BYTE	0016	DATA
MEM1	L WORD	0000	DATA
MEM2	L WORD	0002	DATA
MEM3	L WORD	0004	DATA
N1	NUMBER	01F4	
N2	NUMBER	-0032	
VEC1	L BYTE	0006	DATA
VEC2	L BYTE	000E	DATA
@CPU	TEXT	0101H	
@FILENAME	TEXT	LAB2_FIX	
@VERSION	TEXT	510	

83 SOURCE LINES
83 TOTAL LINES
19 SYMBOLS

47800 + 461507 BYTES SYMBOL SPACE FREE

0 WARNING ERRORS
0 SEVERE ERRORS