

**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 режима адресации и формирования исполнительного**  
**адреса**

Студентка гр. 0383

\_\_\_\_\_

Пустовалова Е.М.

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.**

Вариант 1.

`vec1 1,2,3,4,8,7,6,5`

`vec2 -10,-20,10,20,-30,-40,30,40`

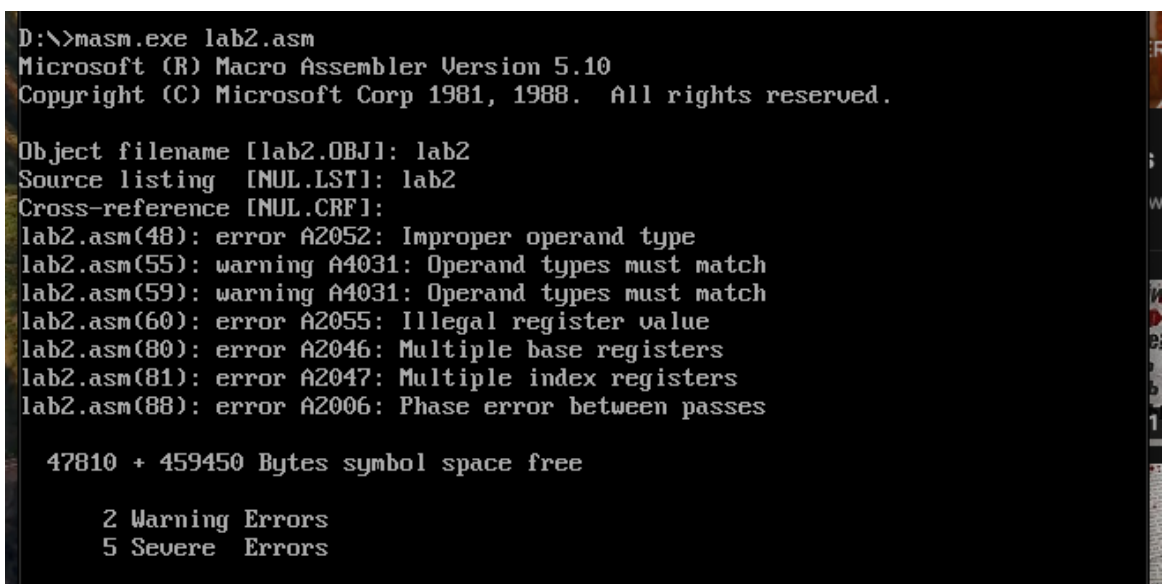
`matr 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-8,-7,-6,-5`

1. При попытке трансляции программы были выведены ошибки:

1. `mov mem3, [bx] lab2.asm(48): error A2052: Improper operand type.`

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

2. `mov cx,vec2[di]` `ab2.asm(55)`: 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(59)`: 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(60)`: error A2055: Illegal register value. Недопустимое значение регистра.
5. `mov ax,matr[bp+bx]` `lab2.asm(80)`: error A2046: Multiple base registers. Для адресации нельзя использовать несколько базовых регистров.
6. `mov ax,matr[bp+di+si]` `lab2.asm(81)`: error A2047: Multiple index registers. Нельзя использовать несколько индексных регистров одновременно.



```

D:\>masm.exe lab2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lab2.OBJ]: lab2
Source listing [NUL.LST]: lab2
Cross-reference [NUL.CRF]:
lab2.asm(48): error A2052: Improper operand type
lab2.asm(55): warning A4031: Operand types must match
lab2.asm(59): warning A4031: Operand types must match
lab2.asm(60): error A2055: Illegal register value
lab2.asm(80): error A2046: Multiple base registers
lab2.asm(81): error A2047: Multiple index registers
lab2.asm(88): error A2006: Phase error between passes

47810 + 459450 Bytes symbol space free

2 Warning Errors
5 Severe Errors
  
```

Скриншот 1. Выведенные ошибки.

2. Строки с ошибками были закомментированы, новый файл `lab2f.asm` протранслирован без ошибок, были созданы диагностический файл `lab2f.lst` и объектный файл `lab2f.obj`. `lab2_fix.obj` был слинкован без ошибок, получен исполняемый файл `lab2f.exe`, затем запущенный в отладчике `afopro`.

(CS) = 1A0A (DS) = 19F5 (ES) = 19F5 (SS) = 1A05 (CX) = 00B0 (BP) = 0000 (DX) = 0000

Таблица 1 – Результат запуска lab2f.exe в отладчике.

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	B8071A	(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	C7060200CEFF	(AX) = 1A07 (DS) = 1A07 (IP) = 0012 DATA SEG +2 000	(AX) = 1A07 (DS) = 1A07 (IP) = 0018 DATA SEG +2 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	(AX) = 0000

			(DS) = 1A07 (IP) = 0042 (SP) = 0012 (ES) = 0000 STACK +0 1A07	(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 001F	(AX) = FFCE (DS) = 1A07 (IP) = 004E (DI) = 0002 (BX) = 0003 (ES) = 1A07 DATA SEG +5 CEFF
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	(AX) = 0000 (DS) = 1A07 (IP) = 0054

			(SP) = 0014 STACK +0 0000	(SP) = 0012 STACK +0 01F4
0054	PUSH [0002]	FF360200	(AX) = 0000 (DS) = 1A07 (IP) = 0054 (SP) = 0012 STACK +0 01F4	(AX) = 0000 (DS) = 1A07 (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

Исходный код программы находится в приложении А.

### **Выводы.**

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



# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ И ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab2.asm

```
; Îðîãðàììà èçó-áíëý ðàæëìâ ääðàñàöèè îðîäññîðà IntelX86
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50

; Ñòâé îðîãðàììû
AStack SEGMENT STACK
        DW 12 DUP(?)
AStack ENDS

; Äàìíûâ îðîãðàììû
DATA SEGMENT

; Äèðäèðèâû îìèñáíëý äàìíûð
mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 1,2,3,4,8,7,6,5
vec2 DB -10,-20,10,20,-30,-40,30,40
matr DB 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-8,-7,-6,-5
DATA ENDS

; Êìä îðîãðàììû
CODE SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:AStack

; Äìèìáíàý îðîäääóðà
Main PROC FAR
        push DS
        sub AX,AX
        push AX
        mov AX,DATA
        mov DS,AX

; Îðîãðàðèðäàðæëìâ ääðàñàöèè íà óðîâíâ Ñìàóáíèé
; Ðäæèñòîðíàý ääðàñàöèý
```

```

        mov ax,n1
        mov cx,ax
        mov bl,EOL
        mov bh,n2
; Îðÿlàÿ àäðãñàöèÿ
        mov mem2,n2
        mov bx,OFFSET vec1
        mov mem1,ax
; Êîñâáííàÿ àäðãñàöèÿ
        mov al,[bx]
        mov mem3,[bx]
; Áàçèðíâáííàÿ àäðãñàöèÿ
        mov al,[bx]+3
        mov cx,3[bx]
; Èíäâêñíàÿ àäðãñàöèÿ
        mov di,ind
        mov al,vec2[di]
        mov cx,vec2[di]
; Ääðãñàöèÿ ñ áàçèðíâáííàÿ è èíäâêñèðíâáííàÿ
        mov bx,3
        mov al,matr[bx][di]
        mov cx,matr[bx][di]
        mov ax,matr[bx*4][di]

; ÎÐÎÄÊÀ ÐÆËÎÎÄ ÄÄÐÃÑÀÖÈË Ñ Ó×ÀÒÎÎ ÑÃÃÎÎÎÎÎ
; Îäðãíîðäâââââíà ñãñíàíà
; ----- àäðèàíò 1
        mov ax, SEG vec2
        mov es, ax
        mov ax, es:[bx]
        mov ax, 0
; ----- àäðèàíò 2
        mov es, ax
        push ds
        pop es
        mov cx, es:[bx-1]
        xchg cx,ax
; ----- àäðèàíò 3
        mov di,ind
        mov es:[bx+di],ax
; ----- àäðèàíò 4
        mov bp,sp

```

```

        mov ax,matr[bp+bx]
        mov ax,matr[bp+di+si]
; Èñîîëüçîâàíèâ ñâãîáíòà ñòâèà
        push mem1
        push mem2
        mov bp,sp
        mov dx,[bp]+2
        ret 2
Main ENDP
CODE ENDS
        END Main

```

## Название файла: lab2.lst

Microsoft (R) Macro Assembler Version 5.10  
23:23:28

10/6/21

Page 1-1

```

; Îðîãðàììà è÷ó÷àíèý ðâæè?
?îâ ààðâñàòèè îðîâññîðà I
ntelX86

= 0024                EOL EQU '$'
= 0002                ind EQU 2
= 01F4                n1 EQU 500
=-0032                n2 EQU -50

; Ñòâè îðîãðàììû
0000                  AStack SEGMENT STACK
0000 000C[            DW 12 DUP(?)
        ????
        ]

0018                  AStack ENDS

; Äàííûâ îðîãðàììû
0000                  DATA SEGMENT

; Äèðâèèèîâû îèèñàíèý äàíí?
?ð

0000 0000             mem1 DW 0
0002 0000             mem2 DW 0
0004 0000             mem3 DW 0
0006 01 02 03 04 08 07   vec1 DB 1,2,3,4,8,7,6,5

```

```

06 05
000E F6 EC 0A 14 E2 D8      vec2 DB -10,-20,10,20,-30,-40,30,40
1E 28
0016 01 02 03 04 FC FD      matr DB 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-8,-7,-
6,-5
FE FF 05 06 07 08
F8 F9 FA FB
0026                      DATA ENDS

; Êîä îõîäöàîîû
0000                      CODE SEGMENT
                          ASSUME CS:CODE, DS:DATA, SS:AStack

; Ãîëîâîàÿ îõîöääóðà
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

; ÌÐÎÂÊÊÀ ÐÂÊËÎÎÂ ÀÂÐÂÑÀ?
?ÈÈ ÍÀ ÓÐÎÎÎÂ ÑÎÛÛÎÈÈ
; Ðääëñòîîääÿ ääðññàöëÿ
0009 B8 01F4                      mov ax,n1
000C 8B C8                      mov cx,ax
000E B3 24                      mov bl,EOL
0010 B7 CE                      mov bh,n2

; Îöÿîàÿ ääðññàöëÿ
0012 C7 06 0002 R FFCE          mov mem2,n2

```

Microsoft (R) Macro Assembler Version 5.10  
23:23:28

10/6/21

Page 1-2

```

0018 BB 0006 R                mov bx,OFFSET vec1
001B A3 0000 R                mov mem1,ax
; Êîñääîîàÿ ääðññàöëÿ
001E 8A 07                      mov al,[bx]
                          mov mem3,[bx]

```

lab2.asm(48): error A2052: Improper operand type

```

; Áàçèðíââííàÿ àäðãñàöèÿ
0020 8A 47 03 mov al,[bx]+3
0023 8B 4F 03 mov cx,3[bx]
; Èíââêñíàÿ àäðãñàöèÿ
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(55): warning A4031: Operand types must match
; Àäðãñàöèÿ ñ áàçèðíââííèâ?
? è èíââêñèðíââííèâì
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(59): warning A4031: Operand types must match
003C 8B 85 0022 R mov ax,matr[bx*4][di]
lab2.asm(60): error A2055: Illegal register value

; ÌÐÌÂÐÊÀ ÐÂÊÈÌÂ ÀÄÐÃÑÀ?
?ÈÈ Ñ Ó×ÀÒÌÌ ÑÃÃÌÁÍÒÌÂ
; Ìäðãíìðãããèáíèà ñããíáíò
à
; ----- àäðèàíò 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
; ----- àäðèàíò 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
; ----- àäðèàíò 3
0054 BF 0002 mov di,ind
0057 26: 89 01 mov es:[bx+di],ax
; ----- àäðèàíò 4
005A 8B EC mov bp,sp
005C 3E: 8B 86 0016 R mov ax,matr[bp+bx]
lab2.asm(80): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R mov ax,matr[bp+di+si]
lab2.asm(81): error A2047: Multiple index registers
; Èñííëüçíââííèà ñããíáíòà ?

```

```

                                ?òâêà
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(88): error A2006: Phase error between passes
0076                                CODE ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10  
23:23:28

10/6/21

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

```

```

90 Source  Lines
90 Total   Lines
19 Symbols

```

47810 + 459450 Bytes symbol space free

```

2 Warning Errors
5 Severe Errors

```

## Название файла: lab2f.asm

```

; Ĭōīāōàīīā èçó÷âīēÿ ōâæēīā ääōāñàöèè ĭōīōāññīōà IntelX86
EOL EQU '$'

ind EQU 2
n1 EQU 500
n2 EQU -50

; Ñòâê ĭōīāōàīīû
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

; Äàííûâ ĭōīāōàīīû
DATA SEGMENT

; Äèōâêòèâû ĭîèñàīēÿ äàííûō
mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 1,2,3,4,8,7,6,5
vec2 DB -10,-20,10,20,-30,-40,30,40
matr DB 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-8,-7,-6,-5
DATA ENDS

; Êīā ĭōīāōàīīû

```

```

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

; Āīēīāīāÿ īōīōāāóōā
Main PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX

; ĪĎĪĀĀĎĒĀ ĎĀĒĒĪĪĀ ĀĀĎĀĒĀÖĒĒ ĪĀ ÓĎĪĀĪĀ ÑĪĀŮĀĪĒĒ
; Ďāāēñōđīāāÿ āāđāñāöèÿ
    mov ax,n1
    mov cx,ax
    mov bl,EOL
    mov bh,n2
; Īōÿīāÿ āāđāñāöèÿ
    mov mem2,n2
    mov bx,OFFSET vec1
    mov mem1,ax
; Êīñāāīīāÿ āāđāñāöèÿ
    mov al,[bx]
    ;mov mem3,[bx]
; Āāçèđīāāīīāÿ āāđāñāöèÿ
    mov al,[bx]+3
    mov cx,3[bx]
; Ēīāāēñīāÿ āāđāñāöèÿ
    mov di,ind
    mov al,vec2[di]
    ;mov cx,vec2[di]
; Āāđāñāöèÿ ñ āāçèđīāāīēāī è èīāāēñēđīāāīēāī
    mov bx,3
    mov al,matr[bx][di]
    ;mov cx,matr[bx][di]
    ;mov ax,matr[bx*4][di]

; ĪĎĪĀĀĎĒĀ ĎĀĒĒĪĪĀ ĀĀĎĀĒĀÖĒĒ Ñ Ó×ĀŌĪĪ ÑĀĀĪĀĪŌĪĀ
; Īāōāīīōāāāēāīēā ñāāīāīōā
; ----- āāōēāīō 1
    mov ax, SEG vec2
    mov es, ax

```



```

        mov ax, es:[bx]
        mov ax, 0
; ----- âàðèàíò 2
        mov es, ax
        push ds
        pop es
        mov cx, es:[bx-1]
        xchg cx, ax
; ----- âàðèàíò 3
        mov di, ind
        mov es:[bx+di], ax
; ----- âàðèàíò 4
        mov bp, sp
        ;mov ax, matr[bp+bx]
        ;mov ax, matr[bp+di+si]
; Èñîîëüçîâàíèà ñàâîáîà ñòàèà
        push mem1
        push mem2
        mov bp, sp
        mov dx, [bp]+2
        ret 2
Main ENDP
CODE ENDS
        END Main

```

## Название файла: lab2f.lst

Microsoft (R) Macro Assembler Version 5.10

10/6/21

23:32:54

Page 1-1

```

; Îðîäðàëëà èçó÷àíèý ðàæè?
?îâ àäðàñàòèè îðîâññîðà I
ntelX86

= 0024                EOL EQU '$'
= 0002                ind EQU 2
= 01F4                n1 EQU 500
=-0032                n2 EQU -50

; Ñòàè îðîäðàëëû
0000                AStack SEGMENT STACK
0000 000C[                DW 12 DUP(?)
        ???

```

]

```

0018                                AStack ENDS

                                ; Äàííûå ìðíäðàììû

0000                                DATA SEGMENT

                                ; Äèðåèèêû ìèññàìèý äàíí?
                                ?õ

0000 0000                        mem1 DW 0
0002 0000                        mem2 DW 0
0004 0000                        mem3 DW 0
0006 01 02 03 04 08 07          vec1 DB 1,2,3,4,8,7,6,5
                                06 05
000E F6 EC 0A 14 E2 D8          vec2 DB -10,-20,10,20,-30,-40,30,40
                                1E 28
0016 01 02 03 04 FC FD          matr DB 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-8,-7,-
6,-5
                                FE FF 05 06 07 08
                                F8 F9 FA FB

0026                                DATA ENDS

                                ; Êíä ìðíäðàììû

0000                                CODE SEGMENT

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

                                ; Äíêíâíàý ìðíäðåóðà

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

                                ; ÌÐÎÄÐÈÀ ÐÄÈÈÎÄ ÄÄÐÄÑÀ?
                                ?ÈÈ ÍÀ ÓÐÎÎÍÀ ÑÎÄÛÎÍÈÈ
                                ; Ðääèñòðíââ ääðåññèè

0009 B8 01F4                        mov ax,n1
000C 8B C8                        mov cx,ax
000E B3 24                        mov bl,EOL
0010 B7 CE                        mov bh,n2

                                ; Ìðýìàý ääðåññèè

```

0012 C7 06 0002 R FFCE mov mem2,n2

Microsoft (R) Macro Assembler Version 5.10

10/6/21

23:32:54

Page 1-2

```
0018 BB 0006 R          mov bx,OFFSET vec1
001B A3 0000 R          mov mem1,ax
; Êîñââîîàÿ àäðâñàöèÿ
001E 8A 07              mov al,[bx]
;mov mem3,[bx]
; Áàçèðîîââîîàÿ àäðâñàöèÿ
0020 8A 47 03          mov al,[bx]+3
0023 8B 4F 03          mov cx,3[bx]
; Èíââêñîàÿ àäðâñàöèÿ
0026 BF 0002          mov di,ind
0029 8A 85 000E R      mov al,vec2[di]
;mov cx,vec2[di]
; Àäðâñàöèÿ ñ ààçèðîîââîîèâ?
? è èíââêñèðîîââîîèâ
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]
; Ìðîîâðèà ðâðèîîâ àäðâñà?
?ÈÈ Ñ Ó×ÀÒîî Ñââîîîîâ
; Ìâðâîîðâââèâîèâ ñââîîî
à
; ----- àäðèàî 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
; ----- àäðèàî 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
; ----- àäðèàî 3
```

```

0048 BF 0002                                mov di,ind
004B 26: 89 01                            mov es:[bx+di],ax
; ----- âàðèàíò 4
004E 8B EC                                mov bp,sp
;mov ax,matr[bp+bx]
;mov ax,matr[bp+di+si]
; Èñîïëüçîââîéâ ñâãîáíòà ?
?òâêà
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

```

Microsoft (R) Macro Assembler Version 5.10  
23:32:54

10/6/21

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	lab2f	
@VERSION . . . . .	TEXT	510	

90 Source Lines  
90 Total Lines  
19 Symbols

47810 + 459450 Bytes symbol space free

0 Warning Errors  
0 Severe Errors