

**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**

Студентка гр. 0383

Преподаватель

Петровская Е.С.

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучение режимов 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.

Выриант 6:

`vec1: 18,17,16,15,11,12,13,14`

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

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

### **Выполнение работы.**

В ходе работы был создан исходный файл программы `lab2.asm`. При трансляции программы были получены ошибки:

lab2.asm(54): error A2052: Improper operand type

- Инструкция MOV пересылает один байт или слово только между регистрами или регистром и памятью и не может передавать данные между двумя адресами памяти

lab2.asm(61): warning A4031: Operand types must match

- Размеры операндов не совпадают, CX - регистр размера 2 байт, vec2[di] - ячейка памяти размера 1 байт

lab2.asm(65): warning A4031: Operand types must match

- Размеры операндов не совпадают, CX - регистр размера 2 байт, matr[bx][di] - ячейка памяти размера 1 байт

lab2.asm(66): error A2055: Illegal register value

- Недопустимое значение регистра, попытка расширения 16-разрядного регистра

lab2.asm(86): error A2046: Multiple base registers

- Попытка использовать несколько базовых регистров одновременно

lab2.asm(87): error A2047: Multiple index registers

- Попытка использовать несколько индексных регистров одновременно

lab2.asm(94): error A2006: Phase error between passes

- Ошибка между проходами в процессе трансляции

Таблица 1 – Результаты прогона исправленной программы lab2 под управлением отладчика

Начальное содержимое сегментных регистров		(CS) = 1A0A	(DS) = 19F5	(ES) = 19F5	(SS) = 1A05
Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти		
			до выполнения	после выполнения	
0000	PUSH DS	1E	(SP) = 0018	(SP) = 0016	

			(IP) = 0000 Stack +0 0000 Stack +2 0000	(IP) = 0001 Stack +0 19F5 Stack +2 0000
0001	SUB AX, AX	2BC0	(IP) = 0001	(IP) = 0003
0003	PUSH AX	50	(SP) = 0016 (IP) = 0003 Stack +0 19F5 Stack +2 0000 Stack +4 0000	(SP) = 0014 (IP) = 0004 Stack +0 0000 Stack +2 19F5 Stack +4 0000
0004	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0004	(AX) = 1A07 (IP) = 0007
0007	MOV DS, AX	8ED8	(DS) = 19F5 (IP) = 0007	(DS) = 1A07 (IP) = 0009
0009	MOV AX, 01F4	B8F401	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	MOV CX, AX	8BC8	(CX) = 00B0 (IP) = 000C	(CX) = 01F4 (IP) = 000E
000E	MOV BL, 24	B324	(BX) = 0000 (IP) = 000E	(BX) = 0024 (IP) = 0010
0010	MOV BH, CE	B7CE	(BX) = 0024 (IP) = 0010	(BX) = CE24 (IP) = 0012
0012	MOV [0002], FFCE	C7060200CEFF	(IP) = 0012	(IP) = 0018
0018	MOV BX, 0006	BB0600	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	MOV [0000], AX	A30000	(AX) = 01F4 (IP) = 001B	(AX) = 01F4 (IP) = 001E
001E	MOV AL, [BX]	8A07	(AX) = 01F4 (BX) = 0006	(AX) = 0112 (BX) = 0006

			(IP) = 001E	(IP) = 0020
0020	MOV AL, [BX+03]	8A4703	(AX) = 0112 (BX) = 0006 (IP) = 0020	(AX) = 010F (BX) = 0006 (IP) = 0023
0023	MOV CX, [BX+03]	8B4F03	(CX) = 01F4 (BX) = 0006 (IP) = 0023	(CX) = 0B0F (BX) = 0006 (IP) = 0026
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E+DI]	8A850E00	(AX) = 010F (IP) = 0029 Stack +0 0000 Stack +2 19F5 Stack +4 0000	(AX) = 01E2 (IP) = 002D Stack +0 0000 Stack +2 19F5 Stack +4 0000
002D	MOV BX, 0003	BB0300	(BX) = 0006 (IP) = 002D	(BX) = 0003 (IP) = 0030
0030	MOV AL, [0016+BX+DI]	8A811600	(AX) = 01E2 (IP) = 0030	(AX) = 01FF (IP) = 0034
0034	MOV AX, 1A07	B8071A	(AX) = 01FF (IP) = 0034	(AX) = 1A07 (IP) = 0037
0037	MOV ES, AX	8EC0	(ES) = 19F5 (AX) = 1A07 (IP) = 0037	(ES) = 1A07 (AX) = 1A07 (IP) = 0039
0039	MOV AX, ES:[BX]	268B07	(AX) = 1A07 (IP) = 0039	(AX) = 00FF (IP) = 003C
003C	MOV AX, 0000	B80000	(AX) = 00FF (IP) = 003C	(AX) = 0000 (IP) = 003F
003F	MOV ES, AX	8EC0	(ES) = 1A07 (AX) = 0000 (IP) = 003F	(ES) = 0000 (AX) = 0000 (IP) = 0041

0041	PUSH DS	1E	(DS) = 1A07 (SP) = 0014 (IP) = 0041 Stack +0 0000 Stack +2 19F5 Stack +4 0000	(DS) = 1A07 (SP) = 0012 (IP) = 0042 Stack +0 1A07 Stack +2 0000 Stack +4 19F5
0042	POP ES	07	(ES) = 0000 (SP) = 0012 (IP) = 0042 Stack +0 1A07 Stack +2 0000 Stack +4 19F5	(ES) = 1A07 (SP) = 0014 (IP) = 0043 Stack +0 0000 Stack +2 19F5 Stack +4 0000
0043	MOV CX, ES:[BX-01]	268B4FFF	(CX) = 0B0F (IP) = 0043	(CX) = FFCE (IP) = 0047
0047	XCHG AX, CX	91	(AX) = 0000 (CX) = FFCE (IP) = 0047	(AX) = FFCE (CX) = 0000 (IP) = 0048
0048	MOV DI, 0002	BF0200	(DI) = 0002 (IP) = 0048	(DI) = 0002 (IP) = 004B
004B	MOV ES:[BX+DI], AX	268901	(ES) = 1A07 (AX) = FFCE (IP) = 004B	(ES) = 1A07 (AX) = FFCE (IP) = 004E
004E	MOV BP, SP	8BEC	(BP) = 0000 (SP) = 0014 (IP) = 004E	(BP) = 0014 (SP) = 0014 (IP) = 0050
0050	PUSH [0000]	FF360000	(SP) = 0014 (IP) = 0050 Stack +0 0000 Stack +2 19F5 Stack +4 0000	(SP) = 0012 (IP) = 0054 Stack +0 01F4 Stack +2 0000 Stack +4 19F5

0054	PUSH [0002]	FF360200	(SP) = 0012 (IP) = 0054 Stack +0 01F4 Stack +2 0000 Stack +4 19F5 Stack +6 0000	(SP) = 0010 (IP) = 0058 Stack +0 FFCE Stack +2 01F4 Stack +0 0000 Stack +6 19F5
0058	MOV BP, SP	8BEC	(BP) = 0014 (SP) = 0010 (IP) = 0058	(BP) = 0010 (SP) = 0010 (IP) = 005A
005A	MOV DX, [BP+02]	8B5602	(DX) = 0000 (IP) = 005A	(DX) = 01F4 (IP) = 005D
005D	RET far 0002	CA0200	(CS) = 1A0A (SP) = 0010 (IP) = 005D Stack +0 FFCE Stack +2 01f4 Stack +4 0000 Stack +6 19F5	(CS) = 01F4 (SP) = 0016 (IP) = FFCE Stack +0 19F5 Stack +2 0000 Stack +4 0000 Stack +6 0000

Тексты исходных файлов в приложении А

### **Выводы.**

Были изучены режимы адресации и формирования исполнительного адреса.

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab2\_fix.asm

```

;-----|-----|-----|
;      | vec1 | 18,17,16,15,11,12,13,14 |
;      | vec2 | 30,40,-30,-40,10,20,-10,-20 |
;      | matr | -4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5 |
;-----|-----|-----|
;
;
; 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 18,17,16,15,11,12,13,14
vec2 DB 30,40,-30,-40,10,20,-10,-20
matr DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5
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] -> lab2.asm(54): error A2052: Improper
operand type ->
;Инструкция MOV пересылает один байт или слово только
между регистрами или регистром и памятью
;и НЕ может передавать данные между двумя адресами
памяти
; Базированная адресация
mov al,[bx]+3
mov cx,3[bx]
; Индексная адресация
mov di,ind
mov al,vec2[di]
;mov cx,vec2[di] -> lab2.asm(61): warning A4031: Operand
types must match ->
;Размеры операндов не совпадают, CX - регистр размера 2
байт, vec2[di] - ячейка памяти размера 1 байт
; Адресация с базированием и индексированием
mov bx,3
mov al,matr[bx][di]
;mov cx,matr[bx][di] lab2.asm(65): warning A4031: Operand
types must match ->
;Размеры операндов не совпадают, CX - регистр размера 2
байт, matr[bx][di] - ячейка памяти размера 1 байт
;mov ax,matr[bx*4][di] lab2.asm(66): error A2055: Illegal
register value ->
;Недопустимое значение регистра, попытка расширения 16-
разрядного регистра

; 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] lab2.asm(86): error A2046: Multiple
base registers ->
;Попытка использовать несколько базовых регистров
одновременно
;mov ax,matr[bp+di+si] lab2.asm(87): error A2047: Multiple
index registers ->
;Попытка использовать несколько индексных регистров
одновременно

```

```

; Использование сегмента стека
    push mem1
    push mem2
    mov bp,sp
    mov dx,[bp]+2
    ret 2
Main      ENDP
CODE      ENDS
END Main

```

Название файла: lab2\_o.lst

```

;-----|-----|-----|
;      | vec1   | 18,17,16,15,11,12,13,14
;      | vec2   | 30,40,-30,-40,10,20,-10,-20
;      | matr  | -4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5
;-----|-----|-----|
;
;
; 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 18,17,16,15,11,12,13,14
vec2 DB 30,40,-30,-40,10,20,-10,-20
matr DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5
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]
        push    mem1
        push    mem2
        mov     bp,sp
        mov     dx,[bp]+2
        ret     2
Main     ENDP
CODE     ENDS
END Main

```

Название файла: FIX.lst

$\hat{a}$ 

Ⓔ ⓐ ⓑ ⓔ ⓖ ⓗ ⓘ ⓙ ⓚ ⓛ ⓜ ⓝ ⓞ ⓟ ⓠ ⓡ ⓢ ⓣ ⓤ ⓶ ⓷ ⓸ ⓹ ⓻ ⓼ ⓽ ⓾ ⓿

**A**

☐ âââââââââââ€

```
;      â€œ( vec1      â€œ( 18,17,16,15,11,12,13,14
```

 $\hat{a}(\square)$ 

```

; 6  â( vec2  â( 30,40,-30,-40,10,20,-10,
      â(

```

 $\hat{a}(\square)$ 

```

;      @ mat @ (-4,-3,1,2,-2,-1,3,4,5,6,
7,8,-8,-7,-6,-5 @

```

 $\hat{\alpha}(\cdot)$ [illegible] $\hat{a}$ [illegible][illegible][illegible]*i**i*

; ĐÑĐŸĐ³ÑĐ ĆĐĈĐ ° ĐŽĐ·Ń ŃĐĐĐĐĐĐĐĐĐĐĐĐĐĐĐĐ

$$\tilde{N} \otimes \mu \otimes \check{Z}$$
 $\mathbb{E}D\ddot{Y}D^2$ 

Đ ° Đ Ž Ń Đ Ń μ Đ □ Ń ° Đ Ž Đ Ž □ □

Đ; ÑĐÑÑĐ Ñ Ñ ĐŸÑĐ ° I

ntelX86

$$= 0024$$

EOL EOU '\$'

= 0002

ind EOU 2

$$= 01F4$$

n1	EOU	500
----	-----	-----

$$= -0.032$$

n2 EQU -50

```

; D_iN(D_muD^ D_iN(DYD^3N(D^DCEDEN)

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

0018 AStack ENDS
; DDD^oeDeN )D_mu D_iN(DYD^3N(D^DCEDEN)

0000 DATA SEGMENT
; DDDN D_mu^N (DZD^2N) DYD_iDZN(D^DeDZN
DZD^DeDeN)

N)

0000 0000 mem1 DW 0
0002 0000 mem2 DW 0
0004 0000 mem3 DW 0
0006 12 11 10 0F 0B 0C vec1 DB 18,17,16,15,11,12,13,14
      0D 0E
000E 1E 28 E2 D8 0A 14 vec2 DB 30,40,-30,-40,10,20,-
10,-20
      F6 EC
0016 FC FD 01 02 FE FF matr DB -4,-3,1,2,-2,-
1,3,4,5,6,7,8,-8,
      -7,-6,-5
      03 04 05 06 07 08
      F8 F9 FA FB
0026 DATA ENDS

; DDDZ iN (DYD^3N(D^DCEDEN)

0000 CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:AStack

; DDD YD^DeD ^N D_iN(DYN(D_muDZN)N(D^
0000 Main PROC FAR

```



```

;DŽ DDD CEŸD DμŃ( DzDμŃ(DμDŽD
°D²D°Ń(Ń( DŽD°DœDœŃ)Dμ DCEμDŸDŽŃ)
DŽD²Ń)DCEŃ D°
DŽŃ(DμŃ)D°DCEDŽ DzD°DCEŃŃDŽ(
; DDD°D·ŽŃ (DŸD²D°DœDœD°Ń
D°DŽŃŃDμD°Ń°DŽŃ
0020 8A 47 03 mov al,[bx]+3
0023 8B 4F 03 mov cx,3[bx]
; DDD°DŽD °Ń )DœD°Ń D°DŽŃ(DμŃ)D°ŃDŽŃ
0026 BF 0002 mov di,ind
0029 8A 85 000E R mov al,vec2[di]
;mov cx,vec2[di] -> lab2.asm(61):
warn
ing A4031: Operand types must match ->
;D D°D·DCEμŃ(Ń)
DŸDzDμŃ(D°DœDŽD
ŸD² DœDμ Ń)DŸD²DzD°DŽD°ŃŃ, CX -
Ń(DμD³DŽŃ)Ń(Ń
( Ń(D°D·DCEμŃ(D° 2 D±D°D¹Ń(, vec2[di] -
ŃŃD>Dμ
4757° 707°7CE8837 (7 °7·7CE7μ8(7° 1
D±D°D¹Ń(
; DŃŃDμŃD°ŽŃŃ
D±D°D·DŽŃ(DŸD²D°DœDŽDμ
DCE DŽ DŽDœDŽDμD°Ń)DŽŃ(DŸD²D°DœDŽDμDCE
002D BB 0003 mov bx,3
0030 8A 81 0016 R mov al,matr[bx][di]
;mov cx,matr[bx][di] lab2.asm(65):
war
ning A4031: Operand types must match ->
;D D°D·DCEμŃ(Ń)
DŸDzDμŃ(D°DœDŽD
ŸD² DœDμ Ń)DŸD²DzD°DŽD°ŃŃ, CX -
Ń(DμD³DŽŃ)Ń(Ń
( Ń(D°D·DCEμŃ(D° 2 D±D°D¹Ń(, matr[bx*4][di]
- Ń

```

□Ñ»Đ<sub>μ</sub>Đ<sup>1</sup>Đ<sup>0</sup>Đ° Đ¿Đ°ĐĈĚÑÑĐŽÑ ( Đ(Đ°Đ·ĈĚĐ <sub>μ</sub>Ñ(Đ° 1

Đ±Đ°Đ

48(

;mov ax,matr[bx\*4][di] lab2.asm(66):

e

rror A2055: Illegal register value ->

;ĐĐĐ<sub>μ</sub>ŽĐŸĐ¿Ñ )Ñ)Ñ(ĐžĐĈĚĐŸĐ<sub>μ</sub> Đ·ĐœĐ



```

                                °Ñ»ÐµÐœÐŹÐµ                                Ñ(ÐµÐ³ÐŹÑ)Ñ(Ñ(Ð°,
Ð¿ÐŸÐ¿Ñ)Ñ(Ð°Ð° Ñ(
                                Ð°Ñ)Ñ(ÐŹÑ(ÐµÐœÐŹÑ
                                16-
Ñ(Ð°Ð·Ñ(ÑÑÐœÐŸÐ³ÐŸ Ð( µ
                                Ð³ÐŹÑ)Ñ(Ñ(Ð°

;                                ÐÐ ÐÐÐÐ ÐÐ Ð ÐÐÐÐÐ ÐÐÐÐÐ
ÐÐÐ ÐÐ ÐÐ;Ð
                                ÐŖÐÐ ; Ð£Ð$ÐÐÐÐ; ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ
;                                ÐÑµÐ(ÐµŸÐ¿Ñ (ÐµÐŹÐµÐ»ÐµÐœÐŹÐµ
Ñ)ÐµÐ³ÐœÐµÐœÑ
                                (°
; ----- Ð²Ð°Ñ(ÐŹÐ°ÐœÑ( 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
; ----- Ð²Ð°Ñ(ÐŹÐ°ÐœÑ( 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
; ----- Ð²Ð°Ñ(ÐŹÐ°ÐœÑ( 3
0048 BF 0002                    mov di, ind
004B 26: 89 01                mov es:[bx+di], ax
; ----- Ð²Ð°Ñ(ÐŹÐ°ÐœÑ( 4
004E 8B EC                    mov bp, sp
                                ;mov ax, matr[bp+bx] lab2.asm(86):

```

erro

r A2046: Multiple base registers ->

;ĐĐŃĐžŃ Đ(ž ° ) Ń) ĐžĐŸĐ»ŃĐ·Đ  
 ŸĐ²Đ°Ń(ŃĐ ĐœĐµŃ)Đ°ĐŸĐ»ŃĐ°ĐŸ Đ±Đ°Đ·ĐŸĐ²ŃŃ)  
 ŃĐ

e

rror A2047: Multiple index registers ->  
 ;ĐĐŃĐžŃ Đ(ž ° ) Ń) ĐžĐŸĐ»ŃĐ·Đ  
 ŸĐ²Đ°Ń(ŃĐ ĐœĐµŃ)Đ°ĐŸĐ»ŃĐ°ĐŸ  
 ĐžĐœĐžĐµĐ°Ń)ĐœŃŃ)  
 Ń(ĐµĐ³ĐžŃŃ(Ń(ĐŸĐ²  
 ĐŸĐžĐœĐŸĐ²Ń(ĐµĐœĐµĐœĐœĐŸ  
 ; ĐŃĐžŃĐŸĐ )ĐĐ·ŸĐ²Đ °ĐœĐžĐµ  
 Ń)ĐµĐ³ĐœĐµĐœŃ(Đ°

ŃŃ(ĐµĐ°Đ°

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
11/24/21 23:54:4
```

Symbol

s-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	

@VERSION . . . . . TEXT 510

102 Source Lines

102 Total Lines

19 Symbols

47816 + 455347 Bytes symbol space free

0 Warning Errors

0 Severe Errors

1-3

```

0037 83 C1 02          ADD CX, 2
003A EB 01 90          JMP f12save
003D                      f12save:
003D 89 0E 0008 R      MOV i1, CX
0041 A3 000A R      MOV i2, AX      ;save f
                                1 & f2 results into i1 & i2

                                ;f3 and resulting funct
ion
0044 8B 0E 0008 R      MOV CX, i1
0048 3B 0E 000A R      CMP CX, i2      ;check if i1 <= i2,
if
                                true -> jump
004C 7E 07          JLE other
004E 8B 0E 000A R      MOV CX, i2 ;i2 is min in CX
0052 A1 0008 R      MOV AX, i1 ;i1 is max in AX
0055          other:
                                ;MOV CX, i1
                                ;MOV AX, i2

0055 8B 1E 0006 R      MOV BX, k
0059 83 FB 00          CMP BX, 0
005C 74 05          JZ fin      ;JumpZero check if ZF =
0, if k
                                /= 0 continue here, otherwise jump
005E 8B C8          MOV CX, AX      ;get max into CX

0060 EB 01 90          jmp fin      ; CX has min by default,
if k =
                                = 0 then res = min
                                ;CX NOW HAS RESULT

```

```
0063  CB                fin: RET                ;return to DOS
0064                Main ENDP
0064                CODE ENDS
                END Main
```

```
#Microsoft      (R)      Macro      Assembler      Version      5.10
11/25/21 08:41:0
```

Symbol

s-1

# Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK . . . . .	0018	PARA	STACK
CODE . . . . .	0064	PARA	NONE
DATA . . . . .	000C	PARA	NONE

# Symbols:

N a m e	Type	Value	Attr
A . . . . .	L WORD	0000	DATA
B . . . . .	L WORD	0002	DATA
F12SAVE . . . . .	L NEAR	003D	CODE
F1SEC . . . . .	L NEAR	002C	CODE
FIN . . . . .	L NEAR	0063	CODE
I . . . . .	L WORD	0004	DATA
I1 . . . . .	L WORD	0008	DATA
I2 . . . . .	L WORD	000A	DATA
K . . . . .	L WORD	0006	DATA
MAIN . . . . .	F PROC	0000	CODE Length =
OTHER . . . . .	L NEAR	0055	CODE

0064

```
@CPU . . . . . TEXT 0101h
@FILENAME . . . . . TEXT lab3
@VERSION . . . . . TEXT 510
```

```
104 Source Lines
```

```
104 Total Lines
```

```
19 Symbols
```

```
47966 + 459291 Bytes symbol space free
```

```
0 Warning Errors
```

```
0 Severe Errors
```

Название файла: ORIG.lst



$\hat{a}$ [illegible]

**ââââââââââ**

[illegible]

```

;      â( vec1      â( 18,17,16,15,11,12,13,14

```

 $\hat{\alpha}(\mathbf{z})$ 

```

;      6      â□( vec2      â□( 30,40,-30,-40,10,20,-10,
-20      â□(

```

 $\hat{\alpha}(\mathbf{r})$ 

```

;      @ ( matr  @ (  -4,-3,1,2,-2,-1,3,4,5,6,
7,8,-8,-7,-6,-5 @ (

```

 $\hat{\alpha}(\mathbf{z})$ 

;ââââââââ€¢ââââââââ□□□□□ □□□□□□□□□□□□□□□□□

<sup>^</sup>à

Ⓒ⓪ⓂⓅⓈⓇⓉⓋⓌⓍⓎⓏⓐⓑⓔⓕⓖⓗⓙⓜⓝⓞⓟⓠⓡⓢ⓴⓵⓶⓷⓸⓹⓺⓻⓼⓽⓾⓿

**ââââââââââââ**

☐ ââââââââââ€

*i**i*

; ĐÑĐỖĐ³ÑĐ €Đ€Đ ° ĐŽĐ·ÑÑĐĐuĐœĐŽÑ□

 $\tilde{N} \otimes \mu \otimes \mathbb{Z}$  $\mathbb{E}D\ddot{Y}D^2$ 

Đ ° Đ Ž Ń Đ Ń μ Đ □ Ń ° Đ Ž Đ Ž □ □

Đ; ÑĐÑÑĐ Ñ<sub>μ</sub>ĐŸÑĐ □ □ □ □ ° I

ntelX86

$$= 0024$$

EOL EQU '\$'

$$= 0002$$

ind EOU 2

$$= 01F4$$

n1	EQU	500
----	-----	-----

=-0032

n2 EQU -50

; D<sub>i</sub>Ñ(D<sub>μ</sub>D<sup>o</sup> D<sub>z</sub>Ñ(D<sup>Y</sup>D<sup>3</sup>Ñ(D<sup>o</sup>DCEĐCEÑ)

0000 AStack SEGMENT STACK

0000 000C[ DW 12 DUP(?)

????

]

0018 AStack ENDS

;DĐD<sup>o</sup>œDœÑ )D<sub>μ</sub> D<sub>z</sub>Ñ(D<sup>Y</sup>D<sup>3</sup>Ñ(D<sup>o</sup>DCEĐCEÑ)

0000 DATA SEGMENT

;DĐDÑ D<sub>μ</sub><sup>o</sup>Ñ (DŽD<sup>2</sup>Ñ) D<sup>Y</sup>D<sub>z</sub>DŽÑ)D<sup>o</sup>DœDŽÑ□

DŽD<sup>o</sup>DœDœÑ)

Ñ)

0000 0000 mem1 DW 0

0002 0000 mem2 DW 0

0004 0000 mem3 DW 0

0006 12 11 10 0F 0B 0C vec1 DB 18,17,16,15,11,12,13,14

0D 0E

000E 1E 28 E2 D8 0A 14 vec2 DB 30,40,-30,-40,10,20,-

10,-20

F6 EC

0016 FC FD 01 02 FE FF matr DB -4,-3,1,2,-2,-

1,3,4,5,6,7,8,-8,

-7,-6,-5

03 04 05 06 07 08

F8 F9 FA FB

0026 DATA ENDS

; DĐDŽ ĺÑ (D<sup>Y</sup>D<sup>3</sup>Ñ(D<sup>o</sup>DCEĐCEÑ)

0000 CODE SEGMENT

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

; DĐD D<sup>Y</sup>D<sup>2</sup>DœD<sup>o</sup>Ñ□ D<sub>z</sub>Ñ(D<sup>Y</sup>Ñ(D<sub>μ</sub>DŽÑ)Ñ(D<sup>o</sup>

1-2

```
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
```

```
;      DD DDDD DD DD DD DDDD
```

```
DD DD DD DD
```

```
DŠDDD DD DD fD DDD
```

```
D;DDD DDD
```

```
; D DµD³DžÑÑÑÑ(ĐŸĐ²Đ°Ñ Đ°DžÑ(ĐµÑ)Đ°Ñ«DžÑ
```

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

```
; DÑÑDDE° Ž Ñ°D ÑµDÑ°DžÑ
```

```
0012  C7 06 0002 R FFCE  mov   mem2,n2
0018  BB 0006 R         mov   bx,OFFSET vec1
001B  A3 0000 R         mov   mem1,ax
```

```
; DĐÑD Dµ )œĐœĐ °Ñ Đ°DžÑ(ĐµÑ)Đ°Ñ«DžÑ
```

```
001E  8A 07           mov   al,[bx]
                        mov   mem3,[bx]
```

lab2.asm(54): error A2052: Improper operand type

```
;      DD°D·žÑ (ĐŸĐ²Đ°ĐœĐœĐ°Ñ
```

```
Đ°DžÑÑµDÑ°DžÑ
```

```
0020  8A 47 03         mov   al,[bx]+3
0023  8B 4F 03         mov   cx,3[bx]
```

```
; DĐDžD °Ñ )œœĐ°Ñ Đ°DžÑ(ĐµÑ)Đ°Ñ«DžÑ
```

```
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(61): warning A4031: Operand types must match
;          ÐÐÑÐ ÐµÑÐ°ŽÑÑ  □  □□
Ð±Ð°Ð·ÐžÑ(ÐŸÐ²Ð°ÐœÐžÐµ
          Ðœ Ðž ÐžÐœÐžÐµÐ°Ñ)ÐžÑ(ÐŸÐ²Ð°ÐœÐžÐµÐœÐœ
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(65): warning A4031: Operand types must match
003C 8B 85 0022 R          mov  ax,matr[bx*4][di]
lab2.asm(66): error A2055: Illegal register value
;          ÐÐ ÐÐÐÐ □□ ÐÐÐ □  Ð ÐÐÐÐÐ □□□□ □
ÐÐÐ □ ÐÐµÐ □
          ÐŠÐÐÐ □ ; Ð£Ð$ÐÐÐÐ □ Ð ÐÐÐÐÐÐ(ÐÐ □□□□□□ □
;          ÐÑµÐ(ÐµŸÐµÑ (ÐµÐžÐµÐ»ÐµÐœÐžÐµ
Ñ)ÐµÐ³ÐœÐµÐœÑ
(°
;  ----- Ð²Ð°Ñ(ÐžÐ°ÐœÑ( 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
;  ----- Ð²Ð°Ñ(ÐžÐ°ÐœÑ( 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
;  ----- Ð²Ð°Ñ(ÐžÐ°ÐœÑ( 3
0054 BF 0002              mov  di,ind
0057 26: 89 01          mov  es:[bx+di],ax
;  ----- Ð²Ð°Ñ(ÐžÐ°ÐœÑ( 4

```

```

005A 8B EC                mov  bp,sp
005C 3E: 8B 86 0016 R      mov  ax,matr[bp+bx]
lab2.asm(86): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R      mov  ax,matr[bp+di+si]
lab2.asm(87): error A2047: Multiple index registers
;                                DÑD,ÑYD )DID·YD²D °DœDŽDµ
ÑDµD³DCEµDœÑ(D°
ÑÑ(DµDºD°
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(94): error A2006: Phase error between passes
0076                     CODE      ENDS

                                END Main

```

```
#Microsoft      (R)      Macro      Assembler      Version      5.10
11/24/21 22:45:1
```

Symbol

s-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

97 Source Lines

97 Total Lines

19 Symbols

47810 + 457400 Bytes symbol space free

2 Warning Errors

5 Severe Errors