

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

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Изучение режимов адресации и формирования**  
**исполнительного адреса**  
**Вариант 5**

Студент гр. 1303

Кузнецов Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучения режимов адресации процессора IntelX86 при помощи программы на языке Ассемблер.

### **Задание.**

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу 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.

### **Порядок выполнения работы.**

1. Получить у преподавателя вариант набора значений исходных данных (массивов) `vec1`, `vec2` и `matr` из файла `lr2.dat`, приведенного в каталоге Задания и занести свои данные вместо значений, указанных в приведенной ниже программе.

2. Протранслировать программу с созданием файла диагностических сообщений; объяснить обнаруженные ошибки и закомментировать соответствующие операторы в тексте программы.

3. Снова протранслировать программу и скомпоновать загрузочный модуль.

4. Выполнить программу в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

5. Результаты прогона программы под управлением отладчика должны быть подписаны преподавателем и представлены в отчете.

### **Вариант №5:**

vec1 DB 11,12,13,14,18,17,16,15

vec2 DB 10,20,-10,-20,30,40,-30,-40

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

### **Ход работы.**

1) Занести свои данные вместо значений, указанных в приведенной ниже программе

2) Протранслировать программу с созданием файла диагностических

```
C:\>MASM.EXE LAB2.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [LAB2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
LAB2.ASM(49): error A2052: Improper operand type
LAB2.ASM(57): warning A4031: Operand types must match
LAB2.ASM(61): warning A4031: Operand types must match
LAB2.ASM(62): error A2055: Illegal register value
LAB2.ASM(81): error A2046: Multiple base registers
LAB2.ASM(82): error A2047: Multiple index registers
LAB2.ASM(89): error A2006: Phase error between passes

49894 + 459416 Bytes symbol space free

2 Warning Errors
5 Severe Errors
```

сообщений.

Объяснение полученных ошибок и предупреждений:

1) Строка 49: `mov mem3, [bx]`

Ошибка: `errorA2052: Improper operand type`

Причина: Неподходящий тип операндов. Нельзя читать из памяти и писать в память одной командой.

2) Строка 57: `mov cx,vec2[di]`

Предупреждение: `warning A4301: Operand types must match`

Причина: Несоответствие типов операндов. Размер элементов массива «vec2» 1 байт, а «cx» - 2 байта.

3) Строка 61: `mov cx,matr[bx][di]`

Предупреждение: `warning A4301: Operand types must match`

Причина: Аналогично предыдущему случаю размеры приемника и источника не совпадают: `cx` - 2 байта, а `matr[bx][di]` – 1 байт.

4) Строка 62: `mov ax,matr[bx*4][di]`

Ошибка: `errorA2055: Illegal register value`

Причина: Нельзя масштабировать 2-ух байтовые регистры.

5) Строка 81: `mov ax,matr[bp+bx]`

Ошибка: `errorA2046: Multiple base registers`

Причина: Множественно использованы базовые регистры, при том Нельзя использовать более одного базового регистра.

6) Строка 82: `mov ax,matr[bp+di+si]`

Ошибка: `errorA2047: Multiple index registers`

Причина: Множественно использованы индексные регистры, при том нельзя использовать более одного индексного регистра.

7) Строка(89), Ошибка: error A2006: Phrase error between passes

Причина: Main ENDP — Показывает, что в Main есть ошибки.

Ошибки были закомментированы. Программа снова

была протранслирована и выполнена в пошаговом режиме под управлением отладчика.

Таблица 1 – Результаты отладки

Начальные значения сегментных регистров (CS)=1A0A,

(DS)=19F5.(ES)=19F5, (SS)=1A05

Адрес команды	Символьный код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(SP) = 0018 (IP) = 0000 STACK +0 0000	(SP) = 0016 (IP) = 0001 STACK +0 19F5
0001	SUB AX, AX	2BC0	(IP)= 0001	IP = 0003
0003	PUSH AX	50	(SP)= 0016 (IP)= 0003 STACK +0 = 19F5 STACK +2 = 0000	(SP)= 0014 (IP)= 0004 STACK +0 0000 STACK +2 19F5
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	(IP)=000C (CX)=00B0	(IP)=000E (CX)=01F4
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	C7060200CEF F	(IP)=0012 DS:0000 00 00 00 00 00 00 0B 0C	(IP)=0018 DS:0000 00 00 CE FF 00 00 0B 0C
0018	MOV BX,0006	BB0600	(BX)=CE24 (IP)=0018	(BX)=0006 (IP)=001B
001B	MOV [0000],AX	A30000	(IP)=001B DS:0000 00 00 CE FF 00 00 0C 0B	(IP)=001E DS:0000 F4 01 CE FF 00 00 0C 0B
001E	MOV AL,[BX]	8A07	(AX)=01F4 (IP)=001E	(AX)=010B (IP)=0020
0020	MOV AL, [BX+03]	8A4703	(IP) = 0020 (AX) = 010B	(IP)= 0023 (AX) = 010E
0023	MOV CX, [BX+03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 120E (IP) = 0026
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [DI+000E]	8A850E00	(AX) = 010E (IP) = 0029	(AX)= 01F6 (IP)= 002D
002D	MOV CX, [DI+000E]	8B8D0E00	(CX) = 120E (IP) = 002D	(CX) = ECF6 (IP) = 0031
0031	MOV BX, 0003	BB03000	(IP) = 0031 (BX) = 0006	(IP) = 0034 (BX) = 0003
0034	MOV AL, [BX+DI+0016]	8A811600	(IP) = 0034 (AX) = 01F6	(IP) = 0038 (AX) = 0104
0038	MOV CX, [BX+DI+0016]	8B891600	(CX) = ECF6 (IP) = 0038	(CX) = FE04 (IP) = 003C
003C	MOV AX, 1A07	B8071A	(AX) = 0104 (IP)= 003C	(AX) = 1A07 (IP)= 003F
003F	MOV ES, AX	8EC0	(ES) = 19F5 (IP)= 003F	(ES) = 1A07 (IP)= 0041
0041	MOV AX, ES: [BX]	268B07	(AX) = 1A07 (IP) = 0041	(AX)= 00FF (IP) = 0044
0044	MOV AX, 0000	B80000	(AX) = 00FF (IP) = 0044	(AX)=0000 (IP) = 0047
0047	MOV ES, AX	8EC0	(ES) = 1A07 (IP) = 0047	(ES)= 0000 (IP)= 0049

0049	PUSH DS	1E	(IP) = 0049 (SP)= 0014 STACK +0 0000 STACK +2 19F5 STACK +4 0000	(IP)= 004A (SP)= 0012 STACK +0 1A07 STACK +2 0000 STACK +4 19F5
004A	POP ES	07	(SP)= 0012 (ES)=0000 (IP)= 004A STACK +0 = 1A07 STACK +2 = 0000 STACK +4 =19F5	(SP) = 0014 (ES)=1A07 (IP)= 004B STACK +0 0000 STACK +2 19F5 STACK +4 0000
004B	MOV CX, ES: [BX—01]	268B4FFF	(CX) = FE04 (IP) = 004B	(CX)= FFCE (IP)= 004F
004F	XCHG AX, CX	91	(AX) = 0000 (CX) = FFCE (IP)=004F	(AX) = FFCE (CX) = 0000 (IP)=0050
0050	MOV DI, 0002	BF0200	(IP) = 0050 (DI) = 0002	(IP) = 0053 (DI) = 0002
0053	MOV ES: [BX+DI], AX	268901	(IP) = 0053 DS:0000 F4 01 CE FF 00 00 0B 0C	(IP) = 0056 DS:0000 F4 01 CE FF 00 CE FF 0C
0056	MOV BP, SP	8BEC	(IP) = 0056 (BP) = 0000	(IP) = 0058 (BP) = 0014
0058	PUSH [0000]	FF360000	(IP) = 0058 (SP)=0014 STACK +0 = 0000 STACK +2 = 19F5 STACK +4 =0000	(IP) = 005C (SP) = 0012 STACK +0 01F4 STACK +2 0000 STACK +4 19F5
005C	PUSH [0002]	FF360200	(IP) = 005C (SP) = 0012 STACK +0 01F4 STACK +2 0000 STACK +4 19F5 STACK + 6 0000	(IP) = 0060 (SP) = 0010 STACK +0 FFCE STACK +2 01F4 STACK +4 0000 STACK + 6 19F5
0060	MOV BP, SP	8BEC	(IP) = 0060 (BP) = 0014	(IP) = 0062 (BP) = 0010

0062	MOX DX, [BP+02]	8B5602	(IP) = 0062 (DX) = 0000	(IP) = 0065 (DX) = 01F4
0065	RET Far	CB	(IP) = 0065 (SP) = 0010 (CS) = 1A0A STACK +0 FFCE STACK +2 01F4 STACK +4 0000 STACK + 6 19F5	(IP) = FFCE (SP) = 0016 (CS) = 01F4 STACK +0 19F5 STACK +2 0000 STACK +4 0000 STACK + 6 0000

### **Вывод.**

В ходе данной работы были изучены режимы адресации и формирование исполнительного адреса. В ходе работы был исправлен и пошагового отлажен исходных файл.



## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

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

```

## ПРИЛОЖЕНИЕ В

### ЛИСТИНГИ.

Название файла: listing\_lr2.LST



0010 B7 CE

```
mov bh,n2  
; ÐÑÑÐÐ°Ñ Ð°ŽÑµÑÐ°ÑžÑ
```



END Main

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK . . . . .	0018	PARA	STACK
CODE . . . . .	0068	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 = 0068
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	

91 Source Lines  
91 Total Lines  
19 Symbols

47796 + 459463 Bytes symbol space free

2 Warning Errors  
0 Severe Errors