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(Вариант 5).

Студентка гр. 1383 _____ Чернякова А.Д.
Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

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.

Ход работы.

1. Изменение набора значений исходных данных (массивов) vec1, vec2 и matr из файла lr2.dat, согласно своему варианту.
2. Трансляция программы с созданием файла диагностических сообщений. Объяснение обнаруженных ошибок и предупреждений и закомментирование операторов с ошибками в тексте программы.

· Ошибка lb2.asm(41): error A2052: Improper operand type (Неверный тип операнда)

Строка: mov mem3,[bx]

Нельзя одновременно читать из памяти и писать в память. Нужно сначала перенести данные из памяти в регистр, а уже потом из регистра в необходимый сегмент.

· Предупреждение lb2.asm(48): warning A4031: Operand types must match (Несоответствие типов операндов)

Строка: mov cx, vec2[di]

Типы операндов должны совпадать, а в данном случае, cx – 1 слово, элемент vec2 – 1 байт.

· Предупреждение lb2.asm(52): warning A4031: Operand types must match (Несоответствие типов операндов)

Строка: mov cx, matr[bx][di]

Типы операндов должны совпадать, а в данном случае, cx – 1 слово, элемент matr – 1 байт.

· Ошибка lb2.asm(53): error A2055: Illegal register value (Незаконное использование регистра)

Строка: mov ax,matr[bx*4][di]

В данном случае используется базово-индексная адресация. В таких случаях в регистре хранится адрес начала структуры данных, а доступ осуществляется к какому-нибудь элементу этой структуры. При данном типе адресации надо сначала изменить значение регистра, а уже потом переводить информацию.

- Ошибка lb2.asm(72): error A2046: Multiple base registers (несколько базовых регистров)

Строка: mov ax,matr[bp+bx]

Регистры bp и bx базовые, поэтому сначала складываются значения регистров, а уже затем данные передается указателю одного из регистров. Таким образом, сначала нужно в регистр bp занести общую сумму, а потом производить смещение.

- Ошибка lb2.asm(73): error A2047: Multiple index registers (несколько индексных регистров)

Строка: mov ax,matr[bp+di+si]

Регистры di и si индексные, поэтому сначала складываются их значения, а потом данные передаются указателю из одного регистра. Сначала в регистр di заносится общая сумма, а потом производится смещение.

- Ошибка lb2.asm(80): error A2006: Phase error between passes Строка: Main ENDP

Данная ошибка свидетельствует о том, что в функции main содержатся ошибки.

3. Повторная трансляция программы и компоновка загрузочного модуля.

```
D:\>masm.exe lb2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.

Object filename [lb2.OBJ]:
Source listing  [NUL.LST]: lb2
Cross-reference [NUL.CRF]:
lb2.asm(48): warning A4031: Operand types must match
lb2.asm(52): warning A4031: Operand types must match

  47842 + 459418 Bytes symbol space free

      2 Warning Errors
      0 Severe  Errors
```

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

Изначальные значения:

SP = 0018

IP = 0000

DS = 19F5

CX = 00B0

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	(DS)=1A07

			(IP)= 0007	(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	C7060200CEFF	(IP)=0012	(IP)=0018
0018	MOV BX,0006	BB0600	(BX)=CE24 (IP)=0018	(BX)=0006 (IP)=001B
001B	MOV [0000],AX	A30000	(IP)=001B	(IP)=001E
001E	MOV AL,[BX]	8A07	(AX)=01F4 (IP)=001E	(AX)=0105 (IP)=0020
0020	MOV AL, [BX+03]	8A4703	(IP) = 0020 (AX) = 0105	(IP)= 0023 (AX) = 0108
0023	MOV CX, [BX+03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 0C08 (IP) = 0026
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [DI+ 000E]	8A850E00	(AX) = 0108 (IP) = 0029	(AX)= 0114 (IP)= 002D
002D	MOV CX, [000E+DI]	8B8D0E00	(IP) = 002D CX = 0C0B	(IP) = 0030 CX = 1E14
0031	MOV BX, 0003	BB0300	(IP) = 0031 (BX) = 0006	(IP) = 0034 (BX) = 0003

0034	MOV AL, [0016+BX+DI]	8A811600	(AX) = 0114 (IP)= 0034	(AX) = 0103 (IP)= 0037
0038	MOV CX, [0016+BX+DI]	8B891600	(CX) = 0C0B (IP)= 0038	(CX) = 0203 (IP)= 003C
003C	MOV AX, 1A07	B8071A	(AX) = 0103 (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	(IP)= 0041 AX = 0114	(IP)= 0044 AX = 00FF
0044	MOV AX,0000	8B80000	(IP)= 0044 AX = 00FF	(IP)= 0043 AX = 0000
0047	MOV CX, ES: [BX—01]	268B4FFF	(CX) = 120E (IP) = 0043	(CX)= FFCE (IP)= 0047
0047	XCHG AX, CX	91	(AX) = 0000 (CX) = FFCE (IP)=0047	(AX) = FFCE (CX) = 0000 (IP)=0048
0047	MOV ES, AX	8EC0	(ES) = 1A07 (IP)= 0047	(ES)= 0000 (IP)= 0049
0049	PUSH DS	1E	(IP) = 0049	(IP) = 004A
004A	POP ES	07	(IP) = 004A ES = 0000	(IP) = 004B ES = 1A07
004B	MOV CX,ES: [BX-01]	268B4FF	(IP) = 004B CX = 0203	(IP) = 004F CX = FFCE
004F	XCHG AX,CX	91	(IP) = 004F AX = 0000	(IP) = 0050AX = FFCE

0050	MOV DI,0002	BF0200	(IP) = 0050	(IP) = 0053
0053	MOX ES: [BX+DI],AX	268901	(IP) = 0053 B	(IP) = 0056
0056	MOV BP,SP	8BEC	(IP) = 0056 BP = 0000	(IP) = 0058 BBP = 0014
0058	PUSH[0000]	FF360000	(IP) = 0058 STACK + 0 000 +2 19F5 +4 0000 +6 0000	(IP) = 005C STACK +0 01F4 +2 0000 +4 19F5 +6 0000
005C	PUSH[0002]	FF36200	(IP) = 005C SP = 0012	(IP) = 0060 SP = 0010
0060	MOV BP,SP	8BEC	(IP) = 0060 (BP) = 0014	(IP) = 0062 BP = 0010
0062	MOV DX, [BP+02]	8B5602	(IP) = 0062 DX = 0000	(IP) = 0065 DX = 01F4
0065	RET	FAR 0002	(IP) = 0065 (CS) = 1A0A STACK +0 FFCE +2 01F4 +4 0000 +6 19F5	(IP) = 0068 (CS) = 01F4 STACK +0 19F5 +2 0000 +4 0000 +6 0000

Выводы.

В ходе выполнения лабораторной работы были изучены основные режимы адресации памяти.

Приложение А. Код программы lb2.asm

```
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,-4
        0
matr
DB1,2,-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 Б. Листинг успешной трансляции программы с закомментированными ошибочными операторами

```
#Microsof  (R) Macro Assembler Version 5.10                    10/30/22
t
13:56:4
                                                              Page     1-1


 =                          EOL EQU '$'
 0024
 =                          ind EQU 2
 0002                       n1 EQU 500
 =                          n2 EQU -50
 01F4                  ; Стек программы
 =-003                 AStack SEGMENT STACK
 2

 0000
 0000   000C                DW 12 DUP(?)
           [
        ????     ]

 0018                  AStack ENDS
                       ; Данные программы
 0000                  DATA SEGMENT
                       ; Директивы описания даннэ
                       ‹x
 0000  0000                 mem1 DW 0
 0002  0000                 mem2 DW 0
 0004  0000                 mem3 DW 0
 0006  0B 0C 0D 0E 12 11   vec1 DB 11,12,13,14,18,17,16,15
```

```
      10 0F
000E  0A 14 F6 EC 1E 28   vec2 DB 10,20,-10,-20,30,40,-30,-40
      E2 D8
0016  01 02 FC FD 03 04   matr DB1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-6,-
                               5
```

```
       FE FF 05 06 F8 F9
       07 08 FA FB
 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 АДРЕСАЙ
                       ¦ИИ 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
#Microsoft (R) Macro Assembler Version 5.10
10/30/22 13:56:4
                                                              Page     1-2


 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]
 002D  8B 8D 000E R         mov cx,vec2[di]
lb2.asm(48): warning A4031: Operand types must match
                       ; Адресация с базированиеЙ
                       ¼ 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]
lb2.asm(52): warning A4031: Operand types must match
                       ;mov ax,matr[bx*4][di]
                       ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЙ
                       ¦ИИ С УЧЕТОМ СЕГМЕНТОВ
                       ; Переопределение
                       сегмент а
                       ; ------ вариант
 1 003C B8 ---- R      mov ax, SEG vec2
 003F 8E C0                 mov es, ax
```

```
0041  26: 8B 07      mov ax, es:[bx]
0044  B8 0000            mov ax, 0
                       ; ------ вариант 2
0047  8E C0              mov es, ax
0049  1E               push
004A  07               ds pop
004B  26: 8B 4F  F     es
004F  91         F           mov cx,
                       es:[bx-1] xchg cx,ax
0050  BF 0002          ; ------ вариант 3
                             mov di,ind
0053  26: 89 01        mov es:[bx+di],ax
                       ; ------ вариант 4
0056  8B EC                  mov bp,sp
                       ;mov ax,matr[bp+bx]
                       ;mov ax,matr[bp+di+si]
                       ; Использование          э
                       сегмента
0058  FF 36      R     тека
      0000
                             push mem1
005C  FF 36      R           push mem2
      0002
0060  8B EC                  mov bp,sp
0062  8B 56 02               mov dx,[bp]+2
0065  CA 0002                ret 2
0068                   Main ENDP
0068                   CODE ENDS
                       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	lb2	
@VERSION	TEXT	510	

82 Source Lines
82 Total Lines
19 Symbols

47842 + 459418 Bytes symbol space free

2 Warning Errors
0 Severe Errors