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

ОТЧЕТ

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

Студент гр. 9383 _____ Моисейченко К.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2020

Цель работы.

Исправить ошибки в данной программе на языке Ассемблер и применить на практике знания о режимах адресации в языке Ассемблер.

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.

Исходные данные.

Исходный код см. в приложении А.

Вариант 2.

vec1 5,6,7,8,12,11,10,9

vec2 -20,-30,20,30,-40,-50,40,50

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

Полученные ошибки и предупреждения

1. Строчка 42: mov mem3,[bx]

Ошибка: error A2052: Improper operand type

Нельзя прямо передавать объекты из памяти в память. Если нужно передать данные из ячейки bx в ячейку, на которую ссылается переменная mem3, то это нужно делать через регистр AX.

2. Строчка 49: mov cx,vec2[di]

   Ошибка: warning A4031: Operand types must match

   Переменная vec2 - массив, каждая его ячейка имеет тип DB и занимает 1 байт. Регистр CX занимает 2 байта, а место, которое занимают операнды, должно быть одинаковым.

3. Строчка 53: mov cx,matr[bx][di]

   Ошибка: warning A4031: Operand types must match

   То же самое, что и в прошлой ошибке, размер операндов различный.

4. Строчка 54: mov ax,matr[bx*4][di]

   Ошибка: error A2055: Illegal register value

   Операцию умножение на число можно применять только к регистрам с префиксом E.

5. Строчка 73: mov ax,matr[bp+bx]

   Ошибка: error A2046: Multiple base registers

   Нельзя использовать более одного базового регистра. Размер элементов матрицы matr 1 байт, а AX – 2 байта.

6. Строчка 74: mov ax,matr[bp+di+si]

   Ошибка: error A2047: Multiple index registers

   Нельзя использовать более одного индексного регистра. Размер элементов матрицы matr 1 байт, а AX - 2 байта.

ПРОТОКОЛ

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения.	после выполнения
0000	PUSH DS	1E	CS=1A0A DS=19F5 ES=19F5 SS-1A05 SP=0018 STACK=+0 0000 IP=0000	DS=19F5 SP=0016 STACK=+0 19F5 IP=0001
0001	SUB AX,AX	2BC0	AX=0000 IP=0001	AX=0000 IP=0003
0003	PUSH AX	50	AX=0000 SP=0016 STACK=+0 19F5 IP=0003	AX=0000 SP=0014 STACK=+0 0000 +2 19F5 IP=0004
0004	MOV AX,1A07	B8071A	AX=0000 IP=0004	AX=1A07 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 IP=0009	AX=01F4 IP=000C
000C	MOV CX,AX	8BC8	AX=01F4 CX=00B0 IP=000C	AX=01F4 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	C706200CEFF	DS[0002]=00 DS[0003]=00 IP=0012	DS[0002]=CE DS[0003]=FF IP=0018

0018	MOV BX,0006	BB0600	BX=CE24 IP=0018	BX=0006 IP=001B
001B	MOV [0000], AX	A30000	AX=01F4 DS[0000]=00 DS[0001]=00 IP=001B	AX=01F4 DS[0000]=F4 DS[0001]=01 IP=001E
001B	MOV AL,[BX]	8A07	AX=01F4 BX=0006 DS[0006]=05 IP=001E	AX=0105 BX=0006 DS[0006]=05 IP=0020
0020	MOV AL,[BX+03]	8A4703	AX=0101 BX=0006 DS[0009]=08 IP=0020	AX=0108 BX=0006 DS[0009]=08 IP=0023
0023	MOV CX,[BX+03]	8B4F03	CX=01F4 BX=0006 DS[0009]=08 DS[000A]=0C IP=0023	CX=080C BX=0006 DS[0009]=08 DS[000A]=0C IP=0026
0026	MOV DI, 0002	BF0200	DI=0000 IP=0026	DI=0002 IP=0029
0029	MOV AL,[000E+DI]	8A850E00	AX=0108 DI=0002 DS[0010]=14 IP=0029	AX=0114 DI=0002 DS[0010]=14 IP=002D
002D	MOV BX,0003	BB0300	BX=0006 IP=002D	BX=0003 IP=0030
0030	MOV AL,[0016+BX+DI]	8A811600	AX=0114 BX=0003 DI=0002 DS[001B]=03 IP=0030	AX=0103 BX=0003 DI=0002 DS[001B]=03 IP=0034
0034	MOV AX,1A07	B8071A	AX=0103 IP=0034	AX=1A07 IP=0037
0037	MOV ES,AX	8EC0	AX=1A07 ES=19F5 IP=0037	AX=1A07 ES=1A07 IP=0039
0039	MOV AX,ES:[BX]	268B07	AX=1A07 ES=1A07 BX=0003 ES[0003]=FF	AX=00FF ES=1A07 BX=0003 ES[0003]=FF

			ES[0004]=00 IP=0039	ES[0004]=00 IP=003C
003C	MOV AX,0000	B80000	AX=00FF IP=003C	AX=0000 IP=003F
003F	MOV ES,AX	0EC0	AX=0000 ES=1A07 IP=003F	AX=0000 ES=0000 IP=0041
0041	PUSH DS	1E	DS=1A07 STACK=+0 0000 +2 19F5 IP=0041	DS=1A07 STACK=+0 1A07 +2 0000 +4 19F5 IP=0042
0042	POP ES	07	ES=0000 STACK=+0 1A07 +2 0000 +4 19F5 SP=0012 IP=0042	ES=1A07 STACK=+0 0000 +2 19F5 SP=0014 IP=0042
0043	MOV CX, ES:[BX-01]	268B4FFF	CX=0C08 ES=1A07 BX=0003 ES[0002]=CE ES[0003]=FF IP=0043	CX=FFCE ES=1A07 BX=0003 ES[0002]=CE ES[0003]=FF 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 BX=0003 DI=0002 AX=FFCE ES[0005]=00 ES[0006]=05 IP=004B	ES=1A07 BX=0003 DI=0002 AX=FFCE ES[0005]=CE ES[0006]=FF IP=004E
004E	MOV BP,SP	8BEC	BP=0000 SP=0014 IP=004E	BP=0014 SP=0014 IP=0050
0050	PUSH [0000]	FF360000	DS[0000]=F4 DS[0001]=01	DS[0000]=F4 DS[0001]=01

			SP=0014 STACK=+0 0000 +2 19F5 IP=0050	SP=0012 STACK=+0 01F4 +2 0000 +4 19F5 IP = 0054
0054	PUSH [0002]	FF360200	DS[0002]=CE DS[0003]=FF SP=0012 STACK=+0 01F4 +2 0000 +4 19F5 IP = 0054	DS[0002]=CE DS[0003]=FF SP=0012 STACK=+0 FFCE +2 01F4 +4 0000 +6 19F5 IP = 0058
0058	MOV BP,SP	9BEC	SP=0010 BP=0014 IP=0058	SP=0010 BP=0010 IP=005A
005A	MOV DX,[BP+02]	8B5602	DX=0000 BP=0010 SS[0012]=F4 SS[0013]=01 IP=005A	DX=01F4 BP=0010 SS[0012]=F4 SS[0013]=01 IP=005D
005D	RET FAR 0002	CA0200	CS=1A0A SP=0010 STACK=+0 FFCE +2 01F4 +4 0000 +6 19F5 IP=005D	CS=01F4 SP=0016 STACK=+0 19F5

Выводы.

Были найдены и исправлены ошибки в готовой программе на языке Ассемблер. Применены навыки отладки программы на языке Ассемблер, усвоены знания в области регистровой адресации

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл lr2_comp.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 5,6,7,8,12,11,10,9
vec2 DB -20,-30,20,30,-40,-50,40,50
matr DB -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4,8,7,6,5
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]
; Использование сегмента стека
push mem1
push mem2
mov bp,sp
mov dx,[bp]+2
ret 2
Main ENDP
CODE ENDS
END Main
```

Файл lr2_comp.lst:

```
Microsoft (R) Macro Assembler Version 5.10                  11/26/20
03:48:4
                                                                Page
1-1


                        ; Программа изучения режимов адресации
                      процессора 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
```

```
            16 0000  0000            mem1 DW 0
            17 0002  0000            mem2 DW 0
            18 0004  0000            mem3 DW 0
            19 0006  05 06 07 08 0C 0B    vec1 DB 5,6,7,8,12,11,10,9
            20       0A 09
            21 000E   EC E2 14 1E D8 CE     vec2 DB -20,-30,20,30,-40,-
50,40,50
            22       28 32
            23 0016  FB FA F9 F8 04 03    matr DB -5,-6,-7,-8,4,3,2,1,-
1,-2,-3,-4
                     ,8,7,6,5
            24       02 01 FF FE FD FC
            25       08 07 06 05
            26 0026              DATA ENDS
            27                 ; Код программы
            28 0000              CODE SEGMENT
            29                  ASSUME CS:CODE, DS:DATA, SS:AStack
            30                 ; Головная процедура
            31 0000              Main PROC FAR
            32 0000  1E           push DS
            33 0001  2B C0           sub AX,AX
            34 0003  50           push AX
            35 0004  B8 ---- R       mov AX,DATA
            36 0007  8E D8           mov DS,AX
            37                 ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ
                  СМЕЩЕНИЙ
            38                 ; Регистровая адресация
            39 0009  B8 01F4           mov ax,n1
            40 000C  8B C8             mov cx,ax
            41 000E  B3 24             mov bl,EOL
            42 0010  B7 CE             mov bh,n2
            43                 ; Прямая адресация
            44 0012  C7 06 0002 R FFCE     mov mem2,n2
            45 0018  BB 0006 R         mov bx,OFFSET vec1
            46 001B  A3 0000 R         mov mem1,ax
            47                 ; Косвенная адресация
            48 001E  8A 07             mov al,[bx]
            49                 ;; mov mem3,[bx]
            50                 ; Базированная адресация
            51 0020  8A 47 03             mov al,[bx]+3
     Microsoft (R) Macro Assembler Version 5.10                11/26/20
03:48:4
                                                                     Page
1-2


            52 0023  8B 4F 03             mov cx,3[bx]
            53                 ; Индексная адресация
            54 0026  BF 0002             mov di,ind
            55 0029  8A 85 000E R         mov al,vec2[di]
            56                 ;; mov cx,vec2[di]
            57                 ; Адресация с базированием и индексиров
                  анием
            58 002D  BB 0003             mov bx,3
            59 0030  8A 81 0016 R         mov al,matr[bx][di]
            60                 ;; mov cx,matr[bx][di]
            61                 ;; mov ax,matr[bx*4][di]
```

```
            62                  ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ С
                              ЕГМЕНТОВ
            63                  ; Переопределение сегмента
            64                  ; ------ вариант 1
            65 0034  B8 ---- R          mov ax, SEG vec2
            66 0037  8E C0              mov es, ax
            67 0039  26: 8B 07          mov ax, es:[bx]
            68 003C  B8 0000              mov ax, 0
            69                  ; ------ вариант 2
            70 003F  8E C0              mov es, ax
            71 0041  1E               push ds
            72 0042  07               pop es
            73 0043  26: 8B 4F FF           mov cx, es:[bx-1]
            74 0047  91               xchg cx,ax
            75                  ; ------ вариант 3
            76 0048  BF 0002              mov di,ind
            77 004B  26: 89 01          mov es:[bx+di],ax
            78                  ; ------ вариант 4
            79 004E  8B EC              mov bp,sp
            80                  ;; mov ax,matr[bp+bx]
            81                  ;; mov ax,matr[bp+di+si]
            82                  ; Использование сегмента стека
            83 0050  FF 36 0000 R           push mem1
            84 0054  FF 36 0002 R           push mem2
            85 0058  8B EC              mov bp,sp
            86 005A  8B 56 02              mov dx,[bp]+2
            87 005D  CA 0002              ret 2
            88 0060                 Main ENDP
            89 0060                 CODE ENDS
            90                    END Main
      Microsoft (R) Macro Assembler Version 5.10                  11/26/20
03:48:4

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  lr2_comp
@VERSION . . . . . .  TEXT  510


     83 Source  Lines
     83 Total   Lines
     19 Symbols

  47260 + 460000 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```