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. 9383 _____ Лысова А.М.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить режимы адресации и формирования исполнительного адреса, научиться проводить диагностику asm-файла

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 10.

vec1 38,37,36,35,31,32,33,34

vec2 70,80,-70,-80,50,60,-50,-60

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

Полученные ошибки и предупреждения:

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

Неподходящий тип операндов: запрещено перемещение из области памяти в область памяти.

2. mov cx,vec2[di] (warning A4031: Operand types must match)

Несоответствие типов операндов: vec2(слово) — 1 байта, cx(регистр) — 2 байта.

3. mov cx,matr[bx][di] (warning A4031: Operand types must match)

Несоответствие типов операндов: matr[bx][di](слово) — 1 байт, cx(регистр)- 2 байта.

4. mov ax,matr[bx*4][di] (error A2055: Illegal register value)

Незаконное использование регистра: нельзя умножать 16-битовый регистр (+ несоответствие типов операндов)

5. mov ax,matr[bp+bx] (error A2046: Multiple base registers)

Слишком много базовых регистров: нельзя использовать сразу несколько базовых регистров

6. mov ax,matr[bp+di+si] (error A2047: Multiple index registers)

Слишком много индексных регистров: нельзя использовать сразу несколько индексных регистров

ПРОТОКОЛ

Адрес Команды	Символический 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	(AX) = 0000 (IP) = 0001	(AX) = 0000 (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	(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 DS: 0002 = 00 DS: 0003 = 00	(IP) = 0018 DS: 0002 = CE DS: 0003 = FF
0018	MOV BX, 0006	BB0600	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	MOV [0000], AX	A30000	(IP) = 001B DS: 0000 = 00 DS: 0001 = 00	(IP) = 001E DS: 0000 = F4 DS: 0001 = 01
001E	MOV AL, [BX]	8A07	(AX) = 01F4 (IP) = 001E	(AX) = 0101 (IP) = 0020

0020	MOV AL, [BX+03]	8A4703	(AX) = 0101 (IP) = 0020	(AX) = 0104 (IP) = 0023
0023	MOV CX, [BX+03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 0804 (IP) = 0026
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E+DI]	8A850E00	(AX) = 0104 (IP) = 0029	(AX) = 010A (IP) = 002D
002D	MOV BX, 0003	BB0300	(BX) = 0006 (IP) = 002D	(BX) = 0003 (IP) = 0030
0030	MOV AL, [0016+BX+DI]	8A811600	(AX) = 010A (IP) = 0030	(AX) = 01FD (IP) = 0034
0034	MOV AX, 1A07	B8071A	(AX) = 01FD (IP) = 0034	(AX) = 1A07 (IP) = 0037
0037	MOV ES, AX	8EC0	(ES) = 19F5 (IP) = 0037	(ES) = 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 (IP) = 003F	(ES) = 0000 (IP) = 0041
0041	PUSH DS	1E	(SP) = 0014 (IP) = 0041 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000	(SP) = 0012 (IP) = 0042 Stack: +0 1A07 Stack: +2 0000 Stack: +4 19F5
0042	POP ES	07	(SP) = 0012 (ES) = 0000 (IP) = 0042 Stack: +0 1A07 Stack: +2 0000 Stack: +419F5	(SP) = 0014 (ES) = 1A07 (IP) = 0043 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000
0043	MOV CX, ES: [BX-01]	268B4FFF	(CX) = 0804 (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	(IP) = 004B DS: 0005 = 00 DS: 0006 = 01	(IP) = 004E DS: 0005 = CE DS: 0006 = FF
004E	MOV BP, SP	8BEC	(BP) = 0000 (IP) = 004E	(BP) = 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: +4 0000 Stack: +6 19F5
0058	MOV BP, SP	8BEC	(BP) = 0014 (IP) = 0058	(BP) = 0010 (IP) = 005A
005A	MOV DX, [BP+02]	8B5602	(DX) = 0000 (IP) = 005A	(DX) = 01F4 (IP) = 005D
005D	RET Far 0002	CA0200	(SP) = 0010 (CS) = 1A0A Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5	(SP) = 0016 (CS) = 01F4 Stack: +2 19F5 Stack: +2 0000 Stack: +4 0000 Stack: +6 0000

Выводы:

Были изучены режимы адресации и формирования исполнительного адреса, так же проведена диагностика файла.

ПРИЛОЖЕНИЕ А.

Название файла: lb2_ex.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      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

; 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.

Название файла: lb_ex.lst

```
#Microsoft (R) Macro Assembler Version 5.10              10/13/20 21:44:2
                                                        Page     1-1


                                ; Ð Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÐ° ÐžÐ·Ñ Ñ ÐµÐœÐžÑ
Ñ ÐµÐ¶ÐžÐ
                                ŒÐŸÐ² Ð°ÐŽÑ ÐµÑ Ð°Ñ‘ÐžÐž
Ð¿Ñ ÐŸÑ‘ÐµÑ Ñ ÐŸÑ Ð° I
                                ntelX86
 = 0024                                 EOL EQU '$'
 = 0002                                 ind EQU 2
 = 01F4                                 n1 EQU 500
 =-0032                                 n2 EQU -50

                                ; Ð¡Ñ ÐµÐº Ð¿Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÑ
 0000                           AStack SEGMENT STACK
 0000  000C[                            DW 12 DUP(?)
         ????
                        ]

 0018                           AStack ENDS

                                ; Ð Ð°ÐœÐœÑ Ðµ Ð¿Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÑ
 0000                           DATA SEGMENT

                                ; Ð ÐžÑ ÐµÐºÑ ÐžÐ²Ñ ÐŸÐ¿ÐžÑ Ð°ÐœÐžÑ
ÐŽÐ°ÐœÐœÑ
                                Ñ
 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

                                ; Ð ÐŸÐŽ Ð¿Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÑ
 0000                           CODE SEGMENT
                                        ASSUME CS:CODE, DS:DATA, SS:AStack

                                ; Ð ÐŸÐ»ÐŸÐ²ÐœÐ°Ñ Ð¿Ñ ÐŸÑ‘ÐµÐŽÑ Ñ Ð°
 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

                          ; Ð Ð Ð Ð Ð Ð Ð  Ð Ð Ð Ð Ð Ð Ð
Ð Ð Ð Ð Ð¡Ð Ð
                          ŠÐ Ð  Ð Ð  Ð£Ð Ð Ð Ð Ð
Ð¡Ð Ð Ð©Ð Ð Ð Ð Ð
                          ; Ð ÐµÐ³ÐžÑ Ñ Ñ ÐŸÐ²Ð°Ñ
Ð°ÐŽÑ ÐµÑ Ð°Ñ ÐžÑ
 0009  B8 01F4                       mov ax,n1
 000C  8B C8                    mov cx,ax
 000E  B3 24                    mov bl,EOL
 0010  B7 CE                    mov bh,n2

#Microsoft (R) Macro Assembler Version 5.10          10/13/20 21:44:2
                                Page     1-2


                          ; Ð Ñ Ñ ÐŒÐ°Ñ  Ð°ÐŽÑ ÐµÑ Ð°Ñ ÐžÑ
 0012  C7 06 0002 R FFCE        mov mem2,n2
 0018  BB 0006 R                mov bx,OFFSET vec1
 001B  A3 0000 R                mov mem1,ax

                          ; Ð ÐŸÑ Ð²ÐµÐœÐœÐ°Ñ  Ð°ÐŽÑ ÐµÑ Ð°Ñ ÐžÑ
 001E  8A 07                    mov al,[bx]
                                mov mem3,[bx]
LB2_EX.ASM(50): error A2052: Improper operand type

                          ; Ð Ð°Ð·ÐžÑ ÐŸÐ²Ð°ÐœÐœÐ°Ñ
Ð°ÐŽÑ ÐµÑ Ð°Ñ ÐžÑ
 0020  8A 47 03                      mov al,[bx]+3
 0023  8B 4F 03                      mov cx,3[bx]

                          ; Ð ÐœÐŽÐµÐºÑ ÐœÐ°Ñ  Ð°ÐŽÑ ÐµÑ Ð°Ñ ÐžÑ
 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_EX.ASM(59): warning A4031: Operand types must match

                          ; Ð ÐŽÑ ÐµÑ Ð°Ñ ÐžÑ  Ñ
Ð±Ð°Ð·ÐžÑ ÐŸÐ²Ð°ÐœÐŽÐµÐ
                          Œ Ðž ÐžÐœÐŽÐµÐºÑ ÐžÑ ÐŸÐ²Ð°ÐœÐžÐµÐŒ
 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_EX.ASM(64): warning A4031: Operand types must match
 003C  8B 85 0022 R             mov ax,matr[bx*4][di]
LB2_EX.ASM(65): error A2055: Illegal register value
```

```
                        ; Ð Ð Ð Ð Ð Ð Ð Ð Ð  Ð Ð Ð Ð Ð Ð Ð Ð
Ð Ð Ð Ð Ð¡Ð Ð
                        ŠÐ Ð Ð¡ Ð£Ð§Ð Ð¢Ð Ð Ð¡Ð Ð Ð Ð Ð Ð
¢Ð Ð
                        ; Ð ÐµÑ ÐµÐŸÐ¿Ñ ÐµÐŽÐµÐ»ÐµÐœÐžÐµ
Ñ ÐµÐ³ÐŒÐµÐœÑ
                        Ð°
                        ; ------ 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]
LB2_EX.ASM(88): error A2046: Multiple base registers
 0061  3E: 8B 83 0016 R            mov ax,matr[bp+di+si]
LB2_EX.ASM(89): error A2047: Multiple index registers

                        ;  Ð Ñ Ð¿ÐŸÐ»ÑÐ·ÐŸÐ²Ð°ÐœÐžÐµ
Ñ ÐµÐ³ÐŒÐµÐœÑ Ð°
                        Ñ Ñ ÐµÐºÐ°
 0066  FF 36 0000 R        push  mem1
#Microsoft (R) Macro Assembler Version 5.10              10/13/20 21:44:2
                                          Page     1-3


 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
LB2_EX.ASM(98): error A2006: Phase error between passes
 0076                   CODE    ENDS
                              END Main
#Microsoft (R) Macro Assembler Version 5.10              10/13/20 21:44:2
                                          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	LB2_EX	
@VERSION	TEXT	510	

100 Source Lines
100 Total Lines
19 Symbols

47798 + 459462 Bytes symbol space free

2 Warning Errors
5 Severe Errors

ПРИЛОЖЕНИЕ В.

Название файла: lb_res.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      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

; 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
```