

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 8

Студентка гр. 1383

Сардинов М.Г.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучение режимов адресации на языке Ассемблера.

Задание.

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу `lr2_compr.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.

Выполнение работы.

1. Были изменены значения `vec1`, `vec2`, `matr` согласно варианту.

2. Были обнаружены следующие ошибки:

1) `mov mem3,[bx]` (`lab_2.asm(41)`: error A2052: Improper operand type)

Неподходящий тип операнда. Невозможно одновременное считывание из памяти и запись данных в память.

2) `mov cx,vec2[di]` (`lab_2.asm(48)`: warning A4031: Operand types must match) Несоответствие типов операндов. Операнды должны быть одинакового размера. Размерность регистра `'cx'` - 2 байта, а `'vec2'` 1 байт.

3) mov cx,matr[bx][di](lab_2.asm(52): warning A4031: Operand types must match) Несоответствие типов операндов. Размер элементов операнда 'matr' 1 байт, а 'cx' - 2 байта

4) mov ax,matr[bx*4][di](lab_2.asm(53): error A2055: Illegal register value) Незаконное использование регистра. Нельзя масштабировать 16-битные регистры.

5) mov ax,matr[bp+bx](lab_2.asm(72): error A2046: Multiple base registers) Слишком много базовых регистров. Нельзя использовать более одного базового регистра.

6) mov ax,matr[bp+di+si](lab_2.asm(73): error A2047: Multiple index registers) Слишком много индексных регистров. Нельзя использовать более одного индексного регистра.

3.Строки с ошибками были закомментированы.

4.Начальное (содержимое сегментных регистров) состояние режимов (CS) = 1A0A, (DS) = 19F5, (ES) = 19F5, (SS) = 1A05.

Таблица 2- исполнение файла Assemb_lb2.asm

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	Stack: +0 0000 +2 0000 +4 0000 +6 0000 (SP) = 0018 (IP) = 0000	Stack: +0 19F5 +2 0000 +4 0000 +6 0000 (SP) = 0016 (IP) = 0001
0001	SUB AX, AX	2BC0	(AX) = 0000	(AX) = 0000

			(IP) = 0001	(IP) = 0003
0003	PUSH AX	50	Stack: +0 19F5 +2 0000 +4 0000 +6 0000 (SP) = 0016 (IP) = 0003	Stack: +0 0000 +2 19F5 +4 0000 +6 0000 (SP) = 0014 (IP) = 0004
0004	MOV AX, 1A07	B8071A	(IP) = 0004 (AX) = 0000	(IP) = 0007 (AX) = 1A07
0007	MOV DS,AX	8ED8	(IP) = 0007 (DS) = 19F5	(IP) = 0009 (DS) = 1A07
0009	MOV AX, 01F4	B8F401	(IP) = 0009 (AX) = 1A07	(IP) = 000C (AX) = 01F4
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	(IP) = 001B	(IP) = 001E
001E	MOV AL, [BX]	8A07	(AX) = 01F4 (IP) = 001E	(AX) = 011C (IP) = 0020
0020	MOV AL, [BX	8A4703	(AX) = 011C	(AX) = 0119

	+ 03]		(IP) = 0020	(IP) = 0023
0023	MOV CX, [BX, + 03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 1519 (IP) = 0026
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E+ DI]	8A850E00	(AX) = 0119 (IP) = 0029	(AX) = 01EC (IP) = 002D
002D	MOV CX, [000E+DI]	8B8D0E00	(CX) = 1519 (IP) = 2D	(CX) = E2EC (IP) = 31
0031	MOV BX, 0003	BB0300	(BX) = 0006 (IP) = 0031	(BX) = 0003 (IP) = 0034
0034	MOV AL, [0016 + BX + DI]	8A811600	(AX) = 01EC (IP) = 0034	(AX) = 01FB (IP) = 0038
0038	MOV CX, [0016+BX+DI]	8B891600	(CX) = E2EC (IP) = 0038	(CX) = 01FB (IP) = 003C
003C	MOV AX, 1A07	B8071A	(AX) = 01FB (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	Stack: +0 0000 +2 19F5 +4 0000	Stack: +0 1A07 +2 0000 +4 19F5

			+6 0000 (SP) = 0014 (IP) = 0049	+6 0000 (SP) = 0012 (IP) = 004A
004A	POP ES	07	Stack: +0 1A07 +2 0000 +4 19F5 +6 0000 (SP) = 0012 (ES) = 0000 (IP) = 004A	Stack: +0 0000 +2 19F5 +4 0000 +6 0000 (SP) = 0014 (ES) = 1A07 (IP) = 004B
004B	MOV CX, ES: [BX - 01]	268B4FFF	(CX) = 01FB (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	(IP) = 0053
0053	MOV ES: [BX + DI], AX	268901	(IP) = 0053	(IP) = 0056
0056	MOV BP, SP	8BEC	(BP) = 0000 (IP) = 0056	(BP) = 0014 (IP) = 0058
0058	PUSH [0000]	FF360000	Stack: +0 0000 +2 19F5 +4 0000	Stack: +0 01F4 +2 0000 +4 19F5

			+6 0000 (SP) = 0014 (IP) = 0058	+6 0000 (SP) = 0012 (IP) = 005C
005C	PUSH [0002]	FF360200	Stack: +0 01F4 +2 0000 +4 19F5 +6 0000 (SP) = 0012 (IP) = 005C	Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 (SP) = 0010 (IP) = 0060
0060	MOV BP, SP	8BEC	(BP) = 0014 (IP) = 0060	(BP) = 0010 (IP) = 0062
0062	MOV DX, [BP + 02]	8B5602	(DX) = 0000 (IP) = 0062	(DX) = 01F4 (IP) = 0065
0065	POP AX	58	Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 (SP) = 0010 (AX) = FFCE (IP) = 0065	Stack: +0 01F4 +2 0000 +4 19F5 +6 0000 (SP) = 0012 (AX) = FFCE (IP) = 0066

0066	POP BX	5B	Stack: +0 01F4 +2 0000 +4 19F5 +6 0000 (SP) = 0012 (BX) = 0003 (IP) = 0066	Stack: +0 0000 +2 19F5 +4 0000 +6 0000 (SP) = 0014 (BX) = 01F4 (IP) = 0067
0067	RET Far 0002	CA0200	Stack: +0 0000 +2 19F5 +4 0000 +6 0000 (SP) = 0010 (CS) = 1A0A (IP) = 0067	Stack: +0 0000 +2 0000 +4 0000 +6 01F4 (SP) = 0016 (CS) = 01F4 (IP) = 0000

Программный код см. в приложении А.

Файлы диагностических сообщений см. в приложении Б.

Выводы.

В ходе выполнения работы были изучены режимы адресации на языке Ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Assemb_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 28, 27, 26, 25, 21, 22, 23, 24
vec2 DB 20, 30, -20, -30, 40, 50, -40, -50
matr DB -8, -7, 3, 4, -6, -5, 1, 2, -4, -3, 7, 8, -2, -1, 5, 6
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
pop ax
pop bx
ret 2
Main ENDP
CODE ENDS
END Main

```

ПРИЛОЖЕНИЕ Б

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: Assemb_lb2.lst

Microsoft (R) Macro Assembler Version 5.10

11/27/22 13:19:5

Page 1-1

```
= 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
0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 1C 1B 1A 19 15 16  vec1 DB 28, 27, 26, 25, 21, 22, 23, 24
    17 18
000E 14 1E EC E2 28 32  vec2 DB 20, 30, -20, -30, 40, 50, -40, -50
    D8 CE
0016 F8 F9 03 04 FA FB  matr DB -8, -7, 3, 4, -6, -5, 1, 2, -4, -3, 7,
8, -2, -1, 5, 6
    01 02 FC FD 07 08
    FE FF 05 06
```

```

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

; Индексная адресация
0026 BF 0002                       mov di, ind

Microsoft (R) Macro Assembler Version 5.10      11/27/22 13:19:5

                                Page   1-2

0029 8A 85 000E R                  mov al, vec2[di]
002D 8B 8D 000E R                  mov cx, vec2[di]

```

```

ass_2.asm(48): warning A4031: Operand types must match
; Адресация с базированием 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]
ass_2.asm(52): warning A4031: Operand types must match
;mov ax, matr[bx * 4][di]
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмента
; -----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 ds
004A 07                  pop es
004B 26: 8B 4F FF          mov cx, es: [bx - 1]
004F 91                  xchg cx, ax
; -----вариант 3
0050 BF 0002                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 сегмента стека
0058 FF 36 0000 R          push mem1
005C FF 36 0002 R          push mem2
0060 8B EC                mov bp, sp
0062 8B 56 02                mov dx, [bp] + 2
0065 58                  pop ax
0066 5B                  pop bx
0067 CA 0002                ret 2

```

```

006A          Main ENDP
006A          CODE ENDS
END Main

```

```

Microsoft (R) Macro Assembler Version 5.10      11/27/22 13:19:5

```

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	006A	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 = 006A
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 ass_2
@VERSION TEXT 510

84 Source Lines

84 Total Lines

19 Symbols

47828 + 461479 Bytes symbol space free

2 Warning Errors

0 Severe Errors