

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. 1383

Ермакова В.М.

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_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.

Выполнение работы.

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

2. Были обнаружены следующие ошибки:

1) `mov mem3,[bx]` (`lab_2.asm(42): error A2052: Improper operand type`)

Неподходящий тип операнда. Нельзя напрямую копировать значения из памяти в память.

2) `mov cx,vec2[di]` (`lab_2.asm(49): warning A4031: Operand types must match`) Несоответствие типов операндов. Операнды должны быть одинакового размера. В данном случае размерность регистра `'cx'` - слово, 2 байта, а `'vec2'` 1 байт – попытка поместить байт в слово.

3) `mov cx,matr[bx][di]`(lab_2.asm(53): warning A4031: Operand types must match) Несоответствие типов операндов. Размер элементов операнда 'matr' – размерностью в 1 байт, а 'cx' - 2 байта, слово.

4) `mov ax,matr[bx*4][di]`(lab_2.asm(54): error A2055: Illegal register value) Незаконное использование регистра. Нельзя масштабировать 16-битные регистры.

5) `mov ax,matr[bp+bx]`(lab_2.asm(73): 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(74): 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 hello2.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) = 011F (IP) = 0020

0020	MOV AL, [BX + 03]	8A4703	(AX) = 011F (IP) = 0020	(AX) = 0122 (IP) = 0023
0023	MOV CX, [BX, + 03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 2622 (IP) = 0026
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E+ DI]	8A850E00	(AX) = 0122 (IP) = 0029	(AX) = 01CE (IP) = 002D
002D	MOV BX, 0003	BB0300	(BX) = 0006 (IP) = 002D	(BX) = 0003 (IP) = 0030
0030	MOV AL, [0016 + BX + DI]	8A811600	(AX) = 01CE (IP) = 0030	(AX) = 01FF (IP) = 0034
0034	MOV CX, [0016+BX+DI]	8B891600	(CX) = 2622 (IP) = 0034	(CX) = 05FF (IP) = 0037
0037	MOV AX, 1A07	B8071A	(AX) = 01FF (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	(IP) = 003F (ES) = 1A07	(IP) = 0041 (ES) = 0000
0041	PUSH DS	1E	Stack: +0 0000 +2 19F5 +4 0000 +6 0000	Stack: +0 1A07 +2 0000 +4 19F5 +6 0000

			(SP) = 0014 (IP) = 0041	(SP) = 0012 (IP) = 0042
0042	POP ES	07	Stack: +0 1A07 +2 0000 +4 19F5 +6 0000 (SP) = 0012 (ES) = 0000 (IP) = 0042	Stack: +0 0000 +2 19F5 +4 0000 +6 0000 (SP) = 0014 (ES) = 1A07 (IP) = 0043
0043	MOV CX, ES: [BX - 01]	268B4FFF	(IP) = 0043 (CX) = 2622	(IP) = 0047 (CX) = FFCE
0047	XCHG AX, CX	91	(AX) = 0000 (IP) = 0047 (CX) = FFCE	(AX) = FFCE (IP) = 0048 (CX) = 0000
0048	MOV DI, 0002	BF0200	(IP) = 0048	(IP) = 004B
004B	MOV ES: [BX + DI], AX	268901	(IP) = 004B	(IP) = 004E
004E	MOV BP, SP	8BEC	(IP) = 004E (BP) = 0000	(IP) = 0050 (BP) = 0014
0050	PUSH [0000]	FF360000	Stack: +0 0000 +2 19F5 +4 0000 +6 0000	Stack: +0 01F4 +2 0000 +4 19F5 +6 0000

			(IP) = 0050 (SP) = 0014	(IP) = 0054 (SP) = 0012
0054	PUSH [0002]	FF360200	Stack: +0 01F4 +2 0000 +4 19F5 +6 0000 (IP) = 0054 (SP) = 0012	Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 (IP) = 0058 (SP) = 0010
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	Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 (SP) = 0010 (CS) = 1A0A (IP) = 005D	Stack: +0 19F5 +2 0000 +4 0000 +6 0000 (SP) = 0016 (CS) = 01F4 (IP) = FFCE

Программный код см. в приложении А.

Файлы диагностических сообщений см. в приложении Б.

Выводы.

В ходе выполнения лабораторной работы были исследованы основные принципы работы с режимами адресации на языке программирования Ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab_2.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 31,32,33,34,38,37,36,35
    vec2 DB 50,60,-50,-60,70,80,-70,-80
    matr DB -4,-3,7,8,-2,-1,5,6,-8,-7,3,4,-6,-5,1,2
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 РЕЖИВОВ 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 РЕЖИВОВ АДРЕСАЦИИ С УЧЁТОМ 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

```