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»

Студент гр. 1303 _____ Коренев Д.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Изучить режимы адресации и формирование исполнительного адреса.

Задание.

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу lab2.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.

Порядок выполнения работы.

1. Получить у преподавателя вариант набора значений исходных данных (массивов) vec1, vec2 и matr из файла lr2.dat, приведенного в каталоге Задания и занести свои данные вместо значений, указанных в приведенной ниже программе.

2. Протранслировать программу с созданием файла диагностических сообщений; объяснить обнаруженные ошибки и закомментировать соответствующие операторы в тексте программы.

3. Снова протранслировать программу и скомпоновать загрузочный модуль.

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

5. Результаты прогона программы под управлением отладчика должны быть подписаны преподавателем и представлены в отчете.

Выполнение работы.

1. Описание обнаруженных при первоначальной трансляции ошибок и их объяснение :

При первоначальной трансляции были обнаружены следующие ошибки:

1. LB2.ASM(53): error A2052: Improper operand type

строка 53: mov mem3,[bx] - Перемещение данных из памяти невозможны на языке ассемблер

2. LB2.ASM(62): warning A4031: Operand types must match

строка 62: mov cx,vec2[di] — Некорректное использование операндов — с разной размерностью. Размер регистра cx — 2 байта, размер элемента массива — 1 байт.

3. LB2.ASM(67): warning A4031: Operand types must match

строка 67: mov cx,matr[bx][di] — Некорректное использование операндов — с разной размерностью. Размер регистра cx — 2 байта, размер элемента массива — 1 байт.

4. LB2.ASM(68): error A2055: Illegal register value

строка 68: mov ax,matr[bx*4][di] — Нельзя масштабировать на наборе инструкций, так как нет директивы .386 или старше.

5. LB2.ASM(92): error A2046: Multiple base registers

строка 92: mov ax,matr[bp+bx] — Нельзя использовать больше одного базового регистра.

6. LB2.ASM(93): error A2047: Multiple index registers

строка 93: mov ax,matr[bp+di+si] — Нельзя использовать больше одного базового регистра.

7. LB2.ASM(101): error A2006: Phase error between passes

строка 101: Main ENDP — Показывает, что в Main есть ошибки.

Ошибки были закомментированы. Программа снова была протранслирована и выполнена в пошаговом режиме под управлением отладчика.

Результаты выполнения программы под управлением отладчика представлены в Таблице 1.

Начальные значения регистров			
CS = 1A0A	DS = 19F5	ES = 19F5	SS = 1A05

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До	После
0000	PUSH DS	1E	SP = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0001	SUB AX,AX	2BC0	AX = 0000	AX = 0000
0003	PUSH AX	50	SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0004	MOV AX,1A07	B8071A	AX = 0000	AX = 1A07
0007	MOV DS,AX	8ED8	DS = 19F5	DS = 1A07
0009	MOV AX,01F4	B8F401	AX = 1A07	AX = 01F4

000C	MOV CX,AX	8BC8	CX = 00B0	CX = 01F4
000E	MOV BL,24	B324	BX = 0000	BX = 0024
0010	MOV BH,CE	B7CE	BX = 0024	BX = CE24
0012	MOV [0002],FFCE	C7060200CE FF	DS:0002 = 00 DS:0003 = 00	DS:0002 = CE DS:0003 = FF
0018	MOV BX,0006	BB0600	BX = CE24	BX = 0006
001B	MOV [0000],AX	A30000	DS:0000 = 00 DS:0001 = 00	DS:0000 = F4 DS:0001 = 01
001E	MOV AL,[BX]	8A07	AX = 01F4	AX = 0126
0020	MOV AL,[BX+03]	8A4703	AX = 0126	AX = 0123
0023	MOV CX,[BX+03]	8B4F03	CX = 01F4	CX =0123
0026	MOV DI,0002	BF0200	DI = 0000	DI = 0002
0029	MOV AL, [000E+DI]	8A850E00	AX = 0123	AX = 01BA
002D	MOV BX,0003	BB0300	BX = 0006	BX = 0003
0030	MOV AL, [0016+BX+DI]	8A811600	AX = 01BA	AX = 01F9
0034	MOV AX,1A07	B8071A	AX = 01F9	AX = 1A07
0037	MOV ES,AX	83C0	ES = 19F5	ES = 1A07
0039	MOV AX,ES:[BX]	268B07	AX = 1A07	AX = 00FF
003C	MOV AX,0000	B80000	AX = 00FF	AX = 0000
003F	MOV ES,AX	8EC0	ES = 1A07	ES = 0000
0041	PUSH DS	1E	stack +0 0000 +2 19F5 +4 0000 +6 0000	stack +0 1A07 +2 0000 +4 19F5 +6 0000
0042	POP ES	07	SP = 0012 ES = 0000 stack +0 1A07 +2 0000	SP = 0014 ES = 1A07 stack +0 0000 +2 19F5

			+4 19F5 +6 0000	+4 0000 +6 0000
0043	MOV CS,ES:[BX-01]	268B4FFF	CX = 1F23	CX = FFCE
0047	XCHG AX,CX	91	AX = 0000 CX = FFCE	AX = FFCE CX = 0000
0048	MOV DI,0002	BF0200	DI = 0002	DI = 0002
004B	MOV ES:[BX+DI,AX	268901	DS:0005 = 00 DS:0006 = 26	DS:0005 = CE DS:0006 = FF
004E	MOV BP,SP	8BEC	BP = 0000	BP = 0014
0050	PUSH [0000]	FF360000	stack +0 0000 +2 19F5 +4 0000 +6 0000	stack +0 01F4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF360200	stack +0 01F4 +2 0000 +4 19F5 +6 0000	stack +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	MOV BP,SP	8BEC	BP = 0014	BP = 0010
005A	MOV DX,[BP+02]	8B5602	DX = 0000	DX = 01F4
005D	RET Far 0002	CA0200	SP = 0010 CS = 1A0A stack +0 FFCE +2 01F4 +4 0000 +6 19F5	SP = 0016 CS = 01F4 Stack +0 19F5 +2 0000 +4 0000 +6 0000

Вывод.

Изучены режимы адресации и формирование исполнительного адреса. В ходе работы был исправлен и пошагово отлажен исходных файл.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.asm

```
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50


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 38,37,36,35,31,32,33,34
vec2  DB 70,80,-70,-80,50,60,-50,-60
matr  DB -2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-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 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 AX
      pop AX
      ret
Main  ENDP
CODE  ENDS
      END Main
```