

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

Токун Г.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы

Изучить режимы адресации и формирование исполнительного адреса.

Общая формулировка задачи

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

В ходе выполнения лабораторной работы сначала были изменены наборы значений исходных данных в соответствии с вариантом. Затем программа была протранслирована с созданием файла листинга (приложение А), и были получены следующие ошибки:

- 1) `lr2_comp.asm(41): error A2052: Improper operand type`
`mov mem3,[bx]`

Машинные команды не могут использовать сразу два операнда из оперативной памяти, поэтому нельзя производить чтение из памяти и запись в память одновременно. Следует сначала прочитать в регистр, и затем из регистра записывать в память.

2) lr2_comp.asm(48): warning A4031: Operand types must match

```
mov cx,vec2[di]
```

3)

Предупреждение из-за несоответствия размеров операндов: cx – 2 байта, vec2[di] – 1 байт

4) lr2_comp.asm(52): warning A4031: Operand types must match

```
mov cx,matr[bx][di]
```

Предупреждение из-за несоответствия размеров операндов: cx – 2 байта, matr[bx][di] – 1 байт

5) lr2_comp.asm(53): error A2055: Illegal register value

```
mov ax,matr[bx*4][di]
```

Нельзя масштабировать адрес на 086 наборе инструкций.

6) lr2_comp.asm(72): error A2046: Multiple base registers

```
mov ax,matr[bp+bx]
```

Сложение двух базовых регистров, вместо базового и индексного.

7) lr2_comp.asm(73): error A2047: Multiple index registers

```
mov ax,matr[bp+di+si]
```

Сложение базового и двух индексных регистров вместо базового и индексного.

8) lr2_comp.asm(80): error A2006: Phase error between passes
Main ENDP

Ошибка говорит о наличии ошибок в Main.

Исправленная программа (файл программы – приложение А) была протранслирована (файл листинга – приложение Б), и был создан загрузочный модуль:

Начальные значения регистров:

AX = 0000 SI = 0000 CS = 1D9E IP = 0000 Stack = 0000

BX = 0000 DI = 0000 DS = 1D89

CX = 00B0 BP = 0000 ES = 1D89

DX = 0000 SP = 0018 SS = 1D99

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	IP = 0000 DS = 1D89 Stack = 0000 SP = 0018	IP = 0001 DS = 1D89 Stack = 1D89 SP = 0016
0001	SUB AX, AX	2BC0	AX = 0000 IP = 0001	AX = 0000 IP = 0003
0003	PUSH AX	50	AX = 0000 SP = 0016 Stack = 1D89 IP = 0003	AX = 0000 SP = 0014 Stack = 0000 IP = 0004
0004	MOV AX, 1D9B	B89B1D	AX = 0000 IP = 0004	AX = 1D9B IP = 0007
0007	MOV DS, AX	8ED8	AX = 1D9B DS = 1D89 IP = 0007	AX = 1D9B DS = 1D9B IP = 0009

0009	MOV AX, 01F4	B8F401	AX = 1D9B 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	BL = 00 BX = 0000 IP = 000E	BL = 24 BX = 0024 IP = 0010
0010	MOV BH, CE	B7CE	BH = 00 BX = 0024 IP = 0010	BH = CE 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	AX = 01F4 IP = 001B	AX = 01F4 IP = 001E
001E	MOV AL, [BX]	8A07	AL = F4 AX = 01F4 BX = 0006 IP = 001E	AL = 12 AX = 0112 BX = 0006 IP = 0020
0020	MOV AL, [BX + 03]	8A4703	AL = 12 AX = 0112 BX = 0006 IP = 0020	AL = 0F AX = 010F BX = 0006 IP = 0023
0023	MOV CX, [BX + 03]	8B4F03	CX = 01F4 BX = 0006 IP = 0023	CX = 0B0F BX = 0006 IP = 0026
0026	MOV DI, 0002	BF0200	DI = 0000	DI = 0002

			IP = 0026	IP = 0029
0029	MOV AL, [000E + DI]	8A850E00	AL = 0F AX = 010F DI = 0002 IP = 0029	AL = E2 AX = 01E2 DI = 0002 IP = 002D
002D	MOV BX, 0003	BB0300	BX = 0006 IP = 002D	BX = 0003 IP = 0030
0030	MOV AL, [0016 + BX + DI]	8A811600	AL = E2 AX = 01E2 BX = 0003 DI = 0002 IP = 0030	AL = FF AX = 01FF BX = 0003 DI = 0002 IP = 0034
0034	MOV AX, 1D9B	B89B1D	AX = 01FF IP = 0034	AX = 1D9B IP = 0037
0037	MOV ES, AX	8EC0	ES = 1D89 AX = 1D9B IP = 0037	ES = 1D9B AX = 1D9B IP = 0039
0039	MOV AX, ES:[BX]	268B07	AX = 1D9B ES = 1D9B BX = 0003 IP = 0039	AX = 00FF ES = 1D9B BX = 0003 IP = 003C
003C	MOV AX, 0000	B80000	AX = 00FF IP = 003C	AX = 0000 IP = 003F
003F	MOV ES, AX	8EC0	ES = 1D9B AX = 0000 IP = 003F	ES = 0000 AX = 0000 IP = 0041
0041	PUSH DS	1E	DS = 1D9B SP = 0014 Stack = 0000	DS = 1D9B SP = 0012 Stack = 1D9B

			IP = 0041	IP = 0042
0042	POP ES	07	ES = 0000 SP = 0012 Stack = 1D9B IP = 0042	ES = 1D9B SP = 0014 Stack = 0000 IP = 0043
0043	MOV CX, ES:[BX - 01]	268B4FFF	CX = 0B0F ES = 1D9B BX = 0003 IP = 0043	CX = FFCE ES = 1D9B BX = 0003 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 = 1D9B BX = 0003 DI = 0002 AX = FFCE IP = 004B	ES = 1D9B BX = 0003 DI = 0002 AX = FFCE IP = 004E
004E	MOV BP, SP	8BEC	BP = 0000 SP = 0014 IP = 004E	BP = 0014 SP = 0014 IP = 0050
0050	PUSH [0000]	FF360000	SP = 0014 Stack = 0000 IP = 0050	SP = 0012 Stack = 01F4 IP = 0054
0054	PUSH [0002]	FF360200	SP = 0012 Stack = 01F4 IP = 0054	SP = 0010 Stack = FFCE IP = 0058
0058	MOV BP, SP	8BEC	BP = 0014	BP = 0010

			SP = 0010 IP = 0058	SP = 0010 IP = 005A
005A	MOV DX, [BP + 02]	8B5602	DX = 0000 BP = 0010 IP = 005A	DX = 01F4 BP = 0010 IP = 005D
005D	RET Far 0002	CA0200	SP = 0010 Stack = FFCE CS = 1D9E IP = 005D	SP = 0016 Stack = 1D89 CS = 01F4 IP = FFCE

Выводы

В ходе выполнения лабораторной работы были приобретены знания о режимах адресации и формировании исполнительного адреса.

ПРИЛОЖЕНИЕ А

ФАЙЛ_COMP.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 18,17,16,15,11,12,13,14
VEC2 DB 30,40,-30,-40,10,20,-10,-20
MATR DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-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

ФАЙЛ_COMP.LST

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
 10/8/22 21:22:52
 PAGE

1-1

```

= 0024          EOL EQU '$'
= 0002          IND EQU 2
= 01F4          N1 EQU 500
=-0032          N2 EQU -50
; ПЎС, РµРЄ ПЇСБПСПІСБР°РЈРЈС<
0000            ASTACK SEGMENT STACK
0000 000C[      DW 12 DUP(?)
      ????
      ]

0018            ASTACK ENDS
; Р"Р°РSPSC< Рµ ПЇСБПСПІСБР°РЈРЈС<
0000            DATA SEGMENT
;      Р"РЄСБРµРЄС, РЄРІС<      PSPІРЄСГР°РSPЄСЦ
РГР°РSPSC
< C...
0000 0000      MEM1 DW 0
0002 0000      MEM2 DW 0
0004 0000      MEM3 DW 0
0006 12 11 10 0F 0B 0C VEC1 DB 18,17,16,15,11,12,13,14
      0D 0E
000E 1E 28 E2 D8 0A 14 VEC2 DB 30,40,-30,-40,10,20,-10,-20
      F6 EC
0016 FC FD 01 02 FE FF MATR DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-
8,-7,-6,-5
      03 04 05 06 07 08
      F8 F9 FA FB
0026            DATA ENDS
; РЉРSPГ ПЇСБПСПІСБР°РЈРЈС<
0000            CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:ASTACK
; Р"РSP»РSPІРSP°СЦ ПЇСБПSC†РµРГГСГСБР°
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
;      РЦР РЃР'Р•Р РЉР      Р Р•Р-Р□РЃРЃР'
РЃР"Р Р•РЎРЃР
!Р□Р□ РЃРЃ РЈР РЃР'РЃР• РЎРЃР•РРР•РЃР□Р™
; Р РµРІРІЄСГС,СБРSPІР°СЦ Р°РГСБРµСГР°С†РЄСЦ
0009 B8 01F4      MOV AX,N1
000C 8B C8        MOV CX,AX
000E B3 24        MOV BL,EOL
0010 B7 CE        MOV BH,N2

```



```

0050  FF 36 0000 R      PUSH MEM1
0054  FF 36 0002 R      PUSH MEM2
0058  8B EC            MOV BP,SP
005A  8B 56 02          MOV DX,[BP]+2
005D  CA 0002           RET 2
0060                                MAIN ENDP
0060                                CODE ENDS
                                END MAIN

```

```

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
10/8/22 21:22:52

```

SYMBOLS-1

SEGMENTS AND GROUPS:

CLASS	N A M E	LENGTH	ALIGN	COMBINE
	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	
0060	MAIN	F PROC	0000	CODE LENGTH =
	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	

```

82 SOURCE  LINES
82 TOTAL  LINES
19 SYMBOLS

```

47776 + 444860 BYTES SYMBOL SPACE FREE

0 WARNING ERRORS
0 SEVERE ERRORS