

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Организация ЭВМ и систем»
Тема: Изучение режимов адресации и формирования
исполнительного адреса
Вариант 8

Студент гр. 0382

Кондратов Ю.А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучение режимов адресации процессора Intel X86 при помощи программы на языке Ассемблера.

Задание.

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.

Данные для варианта 8 представлены на рисунке 1.

8	<code>vec1</code>	<code>28,27,26,25,21,22,23,24</code>
	<code>vec2</code>	<code>20,30,-20,-30,40,50,-40,-50</code>
	<code>matr</code>	<code>-8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-1,5,6</code>

Рисунок 1 – Данные варианта 8

Файл диагностических сообщений, созданный при трансляции программы с измененными данными представлен в приложении Б. Исходный код команды с исправлениями см. в приложении А. Все ошибки с объяснениями причин занесены в таблицу 1.

Таблица 1 – Ошибки при первой трансляции

Ошибка	Объяснение
<code>mov mem3, [bx]</code>	Перемещение из памяти в память запрещено на архитектурном уровне.

mov cx, vec2[di]	Попытка перемещения DB в регистр размера DW.
mov cx, matr[bx][di]	См. предыдущую строку.
mov ax, matr[bx*4][di]	Умножение 2-х байтовых регистров запрещено.
mov ax, matr[bp+bx]	Нельзя использовать несколько базовых регистров для адресации.
mov ax, matr[bp+di+si]	Нельзя использовать несколько индексных регистров для адресации.

Протокол отладки программы представлен в таблице 2.

Таблица 2 – Протокол отладки программы

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	SP 0018 Stack +0 0000	SP 0016 Stack +0 19F5
0001	SUB AX, AX	2BC0	AX 0000	AX 0000
0003	PUSH AX	50	Stack +0 19F5 SP 0016	Stack +0 0000 +2 19F5 SP 0014
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	MAX BH, CE	B7CE	BX 0024	BX CE24
0012	MOV [0002], FFCE	C7060200CEFF	DS:0000 00 00 00 00 00 00	DS:0000 00 00 CE FF 00 00
0018	MOV BX, 0006	BB0600	BX CE24	BX 0006
001B	MOV [0000], AX	A30000	DS:0000 00 00 CE FF 00 00	DS:0000 F4 01 CE FF 00 00
001E	MOV AL, [BX]	8A07	AX 01F4	AX 011C
0020	MOV AL, [BX+03]	8A4703	AX 011C	AX 0119

0023	MOV CX, [BX+03]	8B4F03	CX 01F4	CX 1519
0026	MOV DI, 0002	BF0200	DI 0000	DI 0002
0029	MOV AL, [000E+DI]	8A850E00	AX 0119	01EC
002D	MOV CS, [000E+DI]	8B8D0E00	CX 1519	CX E2EC
0031	MOV BX, 0003	BB0300	BX 0006	BX 0003
0034	MOV AL, [0016+BX+DI]	8A811600	AX 01EC	AX 01FB
0038	MOV CX, [0016+BX+DI]	8B891600	CX E2EC	CX 01FB
003C	MOV AX, 1A07	B8071A	AX 01FB	AX 1A07
003F	MOV ES, AX	8EC0	ES 19F5	ES 1A07
0041	MOV AX, ES:[BX]	268B07	AX 1A07	AX 00FF
0044	MOV AX, 0000	B80000	AX 00FF	AX 0000
0047	MOV ES, AX	8EC0	ES 1A07	ES 0000
0049	PUSH DS	1E	SP 0014 Stack +0 0000 +2 19F5	SP 0012 Stack +0 1A07 +2 0000 +4 19F5
004A	POP ES	07	ES 0000 SP 0012 Stack +0 1A07 +2 0000 +4 19F5	ES 1A07 SP 0014 Stack +0 0000 +2 19F5
004B	MOV CX,ES:[BX-01]	268B4FFF	CX 1519	CX FFCE
004F	XCHG AX, CX	91	AX 0000 CX FFCE	AX FFCE CX 0000
0050	MOV DI, 0002	BF0200	DI 0002	DI 0002
0053	MOV ES:[BX+DI], AX	268901	DS:0000 F4 01 CE FF 00 00 00	DS:0000 F4 01 CE FF 00 CE FF
0056	MOV BP, SP	8BEC	BP 0000	BP 0014
0058	PUSH [0000]	FF360000	SP 0014 Stack +0 0000 +2 19F5	SP 0012 Stack +0 01F4 +2 0000

				+4 19F5
005C	PUSH [0002]	FF360200	SP 0012 Stack +0 01F4 +2 0000 +4 19F5	SP 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
0060	MOV BP, SP	8BEC	BP 0014	BP 0010
0062	MOV DX, [BP+02]	8B5602	DX 0000	DX 01F4
0065	RET Far 0002	CA0200	CS 1A0A IP 0065 SP 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	CS 01F4 IP FFCE SP 0016 Stack +0 19F5 +2 0000

Выводы.

В ходе работы были изучены режимы адресации процессоров Intel X86, исправлены ошибки в программе, составлен протокол отладки программы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

correct.asm:

```
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
DATA SEGMENT
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
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
Main PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX
    mov ax,n1
    mov cx,ax
    mov bl,EOL
    mov bh,n2
    mov mem2,n2
    mov bx,OFFSET vec1
    mov mem1,ax
    mov al,[bx]
;    mov mem3,[bx]
    mov al,[bx]+3
    mov cx,3[bx]
    mov di,ind
    mov al,vec2[di]
    mov cx,vec2[di]
    mov bx,3
    mov al,matr[bx][di]
    mov cx,matr[bx][di]
;    mov ax,matr[bx*4][di]
    mov ax, SEG vec2
    mov es, ax
    mov ax, es:[bx]
    mov ax, 0
    mov es, ax
    push ds
    pop es
    mov cx, es:[bx-1]
    xchg cx,ax
    mov di,ind
    mov es:[bx+di],ax
```

```
    mov bp,sp
;    mov ax,matr[bp+bx]
;    mov ax,matr[bp+di+si]
    push mem1
    push mem2
    mov bp,sp
    mov dx,[bp]+2
    ret 2
Main ENDP
CODE ENDS
    END Main
```

ПРИЛОЖЕНИЕ Б

ЛИСТИНГИ

До исправления:

Microsoft (R) Macro Assembler Version 5.10 10/2/21
12:41:13

```
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50
0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?) ????      ]

0018          AStack ENDS
0000          DATA SEGMENT
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
0000          CODE SEGMENT
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
0009 B8 01F4          mov ax,n1
000C 8B C8          mov cx,ax
000E B3 24          mov bl,EOL
0010 B7 CE          mov bh,n2
0012 C7 06 0002 R FFCE  mov mem2,n2
0018 BB 0006 R      mov bx,OFFSET vec1
001B A3 0000 R      mov mem1,ax
001E 8A 07          mov al,[bx]
                        mov mem3,[bx]
MAIN.ASM(41): error A2052: Improper operand type
0020 8A 47 03          mov al,[bx]+3
0023 8B 4F 03          mov cx,3[bx]
0026 BF 0002          mov di,ind
0029 8A 85 000E R      mov al,vec2[di]
002D 8B 8D 000E R      mov cx,vec2[di]
MAIN.ASM(48): warning A4031: Operand types must match
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]
MAIN.ASM(52): warning A4031: Operand types must match
```



```

003C 8B 85 0022 R          mov ax,matr[bx*4][di]
MAIN.ASM(53): error A2055: Illegal register value
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
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
0054 BF 0002          mov di,ind
0057 26: 89 01          mov es:[bx+di],ax
005A 8B EC              mov bp,sp
005C 3E: 8B 86 0016 R    mov ax,matr[bp+bx]
MAIN.ASM(72): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R    mov ax,matr[bp+di+si]
MAIN.ASM(73): error A2047: Multiple index registers
0066 FF 36 0000 R      push mem1
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
MAIN.ASM(80): error A2006: Phase error between passes
0076                      CODE ENDS
                          END Main

```

Microsoft (R) Macro Assembler Version 5.10
12:41:13

10/2/21

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	
0076 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 MAIN
@VERSION . . . . . TEXT 510

```

```

82 Source Lines
82 Total Lines
19 Symbols

```

47820 + 459440 Bytes symbol space free

```

2 Warning Errors
5 Severe Errors

```

После исправления:

Microsoft (R) Macro Assembler Version 5.10
18:41:2

10/11/21

Page

1-1

```

= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50
0000            AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
      ????
      ]

0018            AStack ENDS
0000            DATA SEGMENT
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
0000            CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:AStack
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
0009  B8 01F4                    mov ax,n1
000C  8B C8                      mov cx,ax
000E  B3 24                      mov bl,EOL
0010  B7 CE                      mov bh,n2
0012  C7 06 0002 R FFCE         mov mem2,n2
0018  BB 0006 R                 mov bx,OFFSET vec1
001B  A3 0000 R                 mov mem1,ax
001E  8A 07                      mov al,[bx]
                                ;   mov mem3,[bx]
0020  8A 47 03                  mov al,[bx]+3
0023  8B 4F 03                  mov cx,3[bx]

```

Microsoft (R) Macro Assembler Version 5.10
18:41:2

10/11/21

Page

1-2

```

0026  BF 0002                    mov di,ind
0029  8A 85 000E R               mov al,vec2[di]
002D  8B 8D 000E R               mov cx,vec2[di]
MAIN.ASM(48): warning A4031: Operand types must match
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]
MAIN.ASM(52): warning A4031: Operand types must match
                                ;   mov ax,matr[bx*4][di]
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
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
0050  BF 0002                    mov di,ind
0053  26: 89 01                  mov es:[bx+di],ax
0056  8B EC                      mov bp,sp
                                ;   mov ax,matr[bp+bx]
                                ;   mov ax,matr[bp+di+si]
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  CA 0002                    ret 2
0068                                Main ENDP
0068                                CODE ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10
18:41:2

10/11/21

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0068	PARA	NONE
DATA	0026	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
	EOL	NUMBER	0024	
	IND	NUMBER	0002	
0068	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	MAIN	
	@VERSION	TEXT	510	

82 Source Lines
82 Total Lines
19 Symbols

47828 + 459432 Bytes symbol space free

2 Warning Errors
0 Severe Errors