

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

Студент(ка) гр. 9382

Голубева В.П.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить режимы адресации, исправить ошибки в программе.

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.

Обнаруженные ошибки:

- 1) В строке 42: `mov mem3,[bx]`
(error A2052: Improper operand type)
- 2) В строке 49: `mov cx,vec2[di]`
(warning A4031: Operand types must match)
- 3) В строке 53: `mov cx,matr[bx][di]`
(warning A4031: Operand types must match)
- 4) В строке 54: `mov ax,matr[bx*4][di]`

(error A2055: Illegal register value)

5) В строке 73: `mov ax,matr[bp+bx]`

(error A2046: Multiple base registers)

6) В строке 74: `mov ax,matr[bp+di+si]`

(error A2047: Multiple index registers)

Объяснение ошибок:

1) Неподходящий тип операндов. Нельзя читать из памяти и писать в память одной командой. Нужно использовать вспомогательный регистр, чтобы записать значение из `[bx]` в `mem3`

2) Несоответствие типов операндов. Размер элементов массива `vec2` - 1 байт, а `sx` - 2 байта. Размер операндов должен быть одинаковым.

3) Несоответствие типов операндов. Размер элементов массива `matr` - 1 байт, а `sx` - 2 байта. Размер операндов должен быть одинаковым.

4) Незаконное использование регистра. Нельзя умножать 16-битные регистры.

5) Слишком много базовых регистров. Нельзя использовать более одного базового регистра. Размер элементов массива `matr` - 1 байт, а `ax` - 2 байта.

6) Нельзя использовать более одного индексного регистра. Нельзя использовать более двух регистров. Размер элементов матрицы `matr` - 1 байт, а `ax` - 2 байта.

Ошибки были закомментированы.

Протокол выполнения исправленной программы в пошаговом режиме с помощью отладчика AFDPRO представлен в таблице 1.

Начальные значения регистров

IP=0000 CS=1A0A DS=19F5 ES=19F5 SS=1A05

Таблица 1. Выполнение программы в пошаговом режиме

Адрес команды	Шестнадцатеричный код команды	Символьный код команды	Содержимое регистров памяти	
			До выполнения/После выполнения	
0000	1E	PUSH DS	IP=0000 SP=0018 STACK +0 0000	IP=0001 SP=0016 STACK +0 19F5
0001	2BC0	SUB AX, AX	IP=0001 SP=0016	IP=0003
0003	50	PUSH AX	IP=0003	IP=0004 SP=0014 +0 0000 +4 19F5
0004	B8071A	MOV AX, 1A07	IP=0004 SP=0014 AX=0000	IP=0007 AX=1A07
0007	8ED8	MOV DS, AX	IP=0007 AX=1A07	IP=0009 DS=1A07
0009	B8F401	MOV AX, 01F4	IP=0009 DS=1A07	IP=000C AX=01F4
000C	8BC8	MOV CX, AX	IP=000C AX=01F4 CX=0000	IP=000E CX=01F4
000E	B324	MOV BL, 24	IP=000E CX=01F4	IP=0010 BX=0024

			BX=0000	
0010	B7CE	MOV BH, CE	IP=0010 BX=0024	IP=0012 BX=CE24
0012	C706020 0CEFF	MOV [0002], FFCE	IP=0012 BX=CE24	IP=0018
0018	BB0600	MOV BX, 0006	IP=0018	IP=001B BX=0006
001B	A30000	MOV [0000], AX	IP=001B BX=0006	IP=001E
001E	8A07	MOV AL,[BX]	IP=001E	IP=0020 AX=0101
0020	8A4703	MOV AL,[BX+03]	IP=0020 AX=0101	IP=0023
0023	8B4F03	MOV CX,[BX+03]	IP=0023	IP=0026 CX=0804
0026	BF0200	MOV DI, 0002	IP=0026 CX=0804	IP=0029 DS=1A07
0029	8A850E 00	MOV AL, [000E+DI]	IP=0029 DS=1A07	IP=002D AX=010A
002D	BB0300	MOV BX,0003	IP=002D AX=010A	IP=0030
0030	8A81160 0	MOV AL,[0016 + BX + DI]	IP=0030	IP=0034 BX=0003 AX=01FD
0034	B8071A	MOV	IP=0034	IP=0037

		AX,1A07	BX=0003 AX=01FD	AX=1A07
0037	8EC0	MOV ES,AX	IP=0037 AX=1A07	IP=0039 ES=1A07
0039	268B07	MOV AX,ES:[BX]	IP=0039 ES=1A07	IP=003C AX=00FF
003C	B80000	MOV AX,0000	IP=003C AX=00FF	IP=003F AX=0000
003F	8EC0	MOV ES,AX	IP=003F AX=0000	IP=0041 ES=0000
0041	1E	PUSH DS	IP=0041 ES=0000	IP=0042 SP=0012 STACK +0 1A07 +4 19F5
0042	07	POP ES	IP=0042 SP=0012	IP=0043 ES=1A07 SP=0014 STACK +0 0000 +2 19F5 +4 0000
0043	268B4F FF	MOV CX,ES:[BX- 01]	IP=0043 ES=1A07 SP=0014	IP=0047 CX=FFCE
0047	91	XCHG AX,CX	IP=0047	IP=0048

			CX=FFCE	AX=FFCE CX=0000
0048	BF0200	MOV DI,0002	IP=0048	IP=004B
004B	268901	MOV ES: [BX+DI],AX	IP=004B	IP=004E
004E	8BEC	MOV BP,SP	IP=004E	IP=0050 BP=0014
0050 0	FF36000	PUSH [0000]	IP=0050 BP=0014	IP=0054 SP=0012 STACK +0 01F4 +2 0000 +4 19F5
0054 0	FF36020	PUSH [0002]	IP=0054 SP=0012	IP=0058 SP=0010 STACK +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	8BEC	MOV BP,SP	IP=0058 SP=0010	IP=005A BP=0010
005A	8B5602	MOV DX,[BP+02]	IP=005A BP=0010	IP=005D DX=01F4
005D	CA0200	RET	IP=005D	IP=FFCE

		FAR 0002	DX=01F4	
--	--	----------	---------	--

Выводы.

Я изучила режимы адресации и научилась находить и идентифицировать ошибки в программе.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.asm

; Программа изучения режимов адресации процессора IntelX86

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 1,2,3,4,8,7,6,5

vec2 DB -10,-20,10,20,-30,-40,30,40

matr DB 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-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 **ЛИСТИНГ ПРОГРАММЫ**

Название файла: main.lst

#Microsoft (R) Macro Assembler Version 5.10

10/13/20 12:49:1

Page 1-1

```

= 0024                                EOL EQU '$'
= 0002                                ind EQU 2
= 01F4                                n1 EQU 500
=-0032                                n2 EQU -50
                                     ; Ð¡Ñ(ÐµÐ° Ð¡Ñ(ÐŸÐ³Ñ(Ð°ÐœÐœÑ)
0000                                Astack SEGMENT STACK
0000 000C[                            DW 12 DUP(?)
                                     ????
                                     ]

0018                                Astack ENDS
                                     ; Ð Ð°ÐœÐœÑ)Ðµ Ð¡Ñ(ÐŸÐ³Ñ(Ð°ÐœÐœÑ)
0000                                DATA SEGMENT
                                     ; Ð ÐžÑ(ÐµÐ°Ñ(ÐžÐ²Ñ)
ÐŸÐ¡ÐžÑ Ð°ÐœÐžÑ ÐžÐ°ÐœÐœÑ
                                     )Ñ)
0000 0000                            mem1 DW 0
0002 0000                            mem2 DW 0
0004 0000                            mem3 DW 0
0006 01 02 03 04 08 07 vec1 DB 1,2,3,4,8,7,6,5
06 05
000E F6 EC 0A 14 E2 D8    vec2 DB -10,-20,10,20,-30,-40,30,40

```

```

1E 28
0016 01 02 03 04 FC FD      matr DB 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-8,-7,-6,-5
FE FF 05 06 07 08
F8 F9 FA FB
0026                          DATA ENDS
                                ; D  DŸDŽ DĭÑ(DŸD³Ñ(D°DĈDĈÑ)
0000                          CODE SEGMENT
                                ASSUME CS:CODE, DS:DATA, SS:AStack
                                ;                                D  DŸD»DŸD²DœD°Ñ
DĭÑ(DŸÑ«DμDŽÑ)Ñ(D°
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
                                ;                                D  D D  D  D  D D  D
D D  D  D  D  D  D  D  D  D  D D  DĭD  D
                                ŠD  D          D  D          D£D D  D  D  D
DĭD  D  D©D  D  D  D
                                ;                                D DμD³DžÑ)Ñ(Ñ(DŸD²D°Ñ
D°DŽÑ(DμÑ)D°Ñ«DžÑ
0009 B8 01F4                mov ax,n1
000C 8B C8                  mov cx,ax
000E B3 24                  mov bl,EOL
0010 B7 CE                  mov bh,n2
                                ; D  Ñ(Ñ  DĈD°Ñ  D°DŽÑ(DμÑ)D°Ñ«DžÑ
0012 C7 06 0002 R FFCE      mov mem2,n2
0018 BB 0006 R              mov bx,OFFSET vec1

```

```

001B A3 0000 R      mov mem1,ax
                    ;Đ ĐŸŃ)Đ²ĐµĐœĐ°Ń
Đ°ĐŽŃ(ĐµŃ)Đ°Ń«ĐžŃ
001E 8A 07          mov al,[bx]
                    mov mem3,[bx]
MAIN.ASM(42): error A2052: Improper operand type
                    ; Đ Đ°Đ·ĐžŃ ĐŸĐ²Đ°ĐœĐœĐ°Ń
Đ°ĐŽŃ(ĐµŃ)Đ°Ń«ĐžŃ
#Microsoft (R) Macro Assembler Version 5.10      10/13/20 12:49:1
                    Page 1-2

0020 8A 47 03        mov al,[bx]+3
0023 8B 4F 03        mov cx,3[bx]
                    ; Đ ĐœĐŽĐµĐ°Ń ĐœĐ°Ń
Đ°ĐŽŃ(ĐµŃ)Đ°Ń«ĐžŃ
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(49): warning A4031: Operand types must match
                    ; Đ ĐžŃ(ĐµŃ)Đ°Ń«ĐžŃ Ń)
Đ±Đ°Đ·ĐžŃ(ĐŸĐ²Đ°ĐœĐžĐµĐ
                    Ć Đž
ĐžĐœĐŽĐµĐ°Ń)ĐžŃ(ĐŸĐ²Đ°ĐœĐžĐµĐĆ
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(53): warning A4031: Operand types must match
003C 8B 85 0022 R     mov ax,matr[bx*4][di]

```



```

;                                D  Ñ)ĐġĐŸĐ»Ñ{Đ·ĐŸĐ²Đ°ĐœĐžĐµ
Ñ)ĐµĐ³ĐĈĐµĐœÑ(Đ° Ñ
                                )Ñ(ĐµĐ°Đ°

```

```

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(81): error A2006: Phase error between passes

```

0076                   CODE ENDS
                                END Main

```

#Microsoft (R) Macro Assembler Version 5.10 10/13/20 12:49:1

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

83 Source Lines

83 Total Lines

19 Symbols

47828 + 459432 Bytes symbol space free

2 Warning Errors

5 Severe Errors