

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. 9383

Рыбников Р.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Ознакомиться с режимами адресации в Ассемблере. Исправить ошибки в программном коде.

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.

Исходный код.

; Программа изучения режимов адресации процессора IntelX86

EOL EQU '\$' ; warning A4031: Operand type must match

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 11,12,13,14,18,17,16,15

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

matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-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 СМЕЩЕНИИ

; Регистровая адресация

mov ax,n1

mov cx,ax

mov bl,EOL ; error A2009: Symbol not defiend: EOL

mov bh,n2

; Прямая адресация

mov mem2,n2

mov bx,OFFSET vec1

mov mem1,ax

; Косвенная адресация

mov al,[bx]

mov mem3,[bx] ; (1) error A2052: Improper operand type

; Базированная адресация

mov al,[bx]+3

```
mov cx,3[bx]
```

```
; Индексная адресация
```

```
mov di,ind
```

```
mov al,vec2[di]
```

```
mov cx,vec2[di] ; (2) warning A4031: Operand types must match
```

```
; Адресация с базированием и индексированием
```

```
mov bx,3
```

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

```
mov cx,matr[bx][di] ; (3) warning A4031: Operand types must match
```

```
mov ax,matr[bx*4][di] ; (4) error A2055: Illegal register value
```

```
; 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] ; (5) error A2046: Multiple base register

mov ax,matr[bp+di+si] ; (6) error A2047: Multiple index register

; Использование сегмента стека

push mem1

push mem2

mov bp,sp

mov dx,[bp]+2

ret 2

Main ENDP ; (7) error A2006: Phase error between passes

CODE ENDS

END Main

Ошибки кода.

- (2) EOL EQU '\$' ; warning A4031: Operand type must match (гласит о том, что произошло несоответствие типов операндов)
- (44) mov mem3,[bx] ; error A2052: Improper operand type (гласит о том, что запрещается запись и чтение из памяти одной инструкцией)
- (51) mov cx,vec2[di] ; warning A4031: Operand types must match (гласит о том, что произошло несоответствие типов операндов. Размер cx = 2 байта, а размер вектора vec2 и матрицы matr = 1 байт)
- (55) mov cx,matr[bx][di] ; warning A4031: Operand types must match (гласит о том, что произошло несоответствие типов операндов. Размер cx = 2 байта, а размер вектора vec2 и матрицы matr = 1 байт)
- (56) mov ax,matr[bx*4][di] ; error A2055: Illegal register value (гласит о том, что запрещается умножать 16-битовые регистры)
- (75) mov ax,matr[bp+bx] ; error A2046: Multiple base register (гласит о том, что происходит множественное использование регистров, когда разрешено использование только одного)
- (76) mov ax,matr[bp+di+si] ; error A2047: Multiple index register (гласит о том, что происходит множественное использование регистров, когда разрешено использование только одного)

Листинг исправленной программы.

Microsoft (R) Macro Assembler Version 5.10

10/21/20 18:32:2

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	005E	PARA	NONE	
DATA	0026	PARA	NONE	

Symbols:

N a m e	Type	Value	Attr
IND	NUMBER	0002	
MAIN	F PROC	0000	CODE Length = 005E
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	

85 Source Lines

85 Total Lines

18 Symbols

47814 + 459443 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Протокол отладки

DOSBox 0.74-3-2, Cpu speed: 3000 cycles, Frameskip 0, Program: AFDPRO

AX 0000 SI 0000 CS 1A0A IP 0000 Stack +0 0000 Flags 7202

BX 0000 DI 0000 DS 19F5 +2 0000

CX 00AE BP 0000 ES 19F5 HS 19F5 +4 0000 OF DF IF SF ZF AF PF CF

DX 0000 SP 0018 SS 1A05 FS 19F5 +6 0000 0 0 1 0 0 0 0 0

CMD >

0000 1E PUSH DS

0001 2BC0 SUB AX,AX

0003 50 PUSH AX

0004 B8071A MOV AX,1A07

0007 8ED8 MOV DS,AX

0009 B8F401 MOV AX,01F4

000C 8BC8 MOV CX,AX

000E B7CE MOV BH,CE

0 1 2 3 4 5 6 7

DS:0000 CD 20 FF 9F 00 EA F0 FE

DS:0008 AD DE 1B 05 C5 06 00 00

DS:0010 18 01 10 01 18 01 92 01

DS:0018 01 01 01 00 02 FF FF FF

DS:0020 FF FF FF FF FF FF FF FF

DS:0028 FF FF FF FF EB 19 C0 11

DS:0030 A2 01 14 00 18 00 F5 19

DS:0038 FF FF FF FF 00 00 00 00

DS:0040 05 00 00 00 00 00 00 00

DS:0048 00 00 00 00 00 00 00 00

2

0 1 2 3 4 5 6 7

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = Я.ЪЁ■ H|..†...

DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FFT.

DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 L.

DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 B.....Ï.

DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 ↑ 8 ↓ 9 ← 10 →

Таблица 1 – Результаты пошагового выполнения main.asm.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(SP) = 0018 (DS) = 19F5 Stack: +0 0000	(SP) = 0016 (DS) = 19F5 Stack: +0 19F5
0001	SUB AX,AX	2BC0	(AX) = 0000	(AX) = 0000
0003	PUSH AX	50	(AX) = 0000 (SP) = 0016 Stack: +0 19F5 Stack: +2 0000	(AX) = 0000 (SP) = 0014 Stack: +0 0000 Stack: +2 19F5
0004	MOV AX, 1A07	BB071A	(AX) = 0000	(AX) = 1A07
0007	MOV DS,AX	8ED8	(DS) = 19F5 (AX) = 1A07	(DS) = 1A07 (AX) = 1A07
0009	MOV AX, 01F4	B8F401	(AX) = 1A07	(AX) = 01F4
000C	MOV CX, AX	8BC8	(CX) = 00AE (AX) = 01F4	(CX) = 01F4 (AX) = 01F4
000E	MOV BL, 24	B7CE	(BX) = 0000	(BX) = CE00
0010	MOV [0002], FFCE	C7060200CEFF	(BX) = CE00	(BX) = CE00
0016	MOV BX, 0006	BB0600	DS: 0000 = 00 DS: 0008 = 0D	DS: 0002 = CE DS: 0003 = FF
0019	MOV [0000], AX	A30000	(BX) = 0006	(BX) = 0006
001C	MOV AL, [BX]	BA07	(AX) = 01F4 DS: 0000 = F4 DS: 0001 = 01	(AX) = 010B DS: 0000 = F4 DS: 0001 = 01
001E	MOV AL, [BX+03]	8A4703	(AX) = 010B (BX) = 0006	(AX) = 010E (BX) = 0006

0021	MOV CX, [BX+03]	8B4F03	(AX) = 010E (BX) = 0006	(AX) = 010E (BX) = 0006
0024	MOV DI, 0002	BF0200	(CX) = 120E (BX) = 0006	(CX) = 120E (BX) = 0006
0027	MOV AL, [000E+DI]	8A850E00	(DI) = 0002	(DI) = 0002
002B	MOV BX, 0003	BB0300	(AX) = 01F6 (DI) = 0002	(AX) = 01F6 (DI) = 0002
002E	MOV AL, [0016+BX+DI]	8A811600	(BX) = 0003	(BX) = 0003
0032	MOV AX, 1A07	B8071A	(AX) = 0104 (BX) = 0003 (DI) = 0002	(AX) = 1A07 (BX) = 0003 (DI) = 0002
0035	MOV ES, AX	8EC0	(AX) = 1A07	(AX) = 1A07
0037	MOV AX, ES:[BX]	268B07	(ES) = 1A07 (AX) = 1A07	(ES) = 1A07 (AX) = 00FF
003A	MOV AX, 0000	B80000	(AX) = 00FF (ES) = 1A07 (BX) = 0003	(AX) = 0000 (ES) = 1A07 (BX) = 0003
003D	MOV ES,AX	8EC0	(AX) = 0000	(AX) = 0000
003F	MOV DS	1E	(AX) = 0000 (ES) = 0000	(AX) = 0000 (ES) = 0000
0040	POP ES	07	(DS) = 1A07 (SP) = 0012 Stack: +0 1A07 Stack: +2 0000 Stack: +4 19F5	(DS) = 1A07 (SP) = 0014 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000

0041	MOV CX, ES:[BX-01]	268B4FFF	(ES) = 1A07 (SP) = 0014 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000	(ES) = 1A07 (SP) = 0014 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000
0045	XCHG AX, CX	91	(CX)=FFCE (ES) = 1A07 (BX) = 0003	(CX) = 0000 (ES) = 1A07 (BX) = 0003
0046	MOV DI, 0002	BF0200	(AX) = FFCE (CX) = 0000	(AX) = FFCE (CX) = 0000
0049	MOV ES:[BX_DI],AX	268901	(DI) = 0002	(DI) = 0002
004C	MOV BP, SP	8BEC	(ES) = 1A07 (BX) = 0003 (DI) = 0002 (AX) = FFCE DS:0005 =CE DS: 0006 = FF	(ES) = 1A07 (BX) = 0003 (DI) = 0002 (AX) = FFCE DS: 0005 = CE DS: 0006 = FF
004E	PUSH [0000]	FF360000	(BP) = 0014 (SP) = 0014	(BP) = 0014 (SP) = 0012
0052	PUSH [0002]	FF360000	DS:0000 = F4 DS:0001 = 01 (SP) = 0012 Stack: +0 01F4 Stack: +2 0000 Stack: +4 19F5	DS:0000 = F4 DS:0001 = 01 (SP) = 0010 Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000

0056	MOV BP, SP	8BEC	DS:0002 = CE DS:0003 = FF (SP) = 0010 Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5	DS:0002 = CE DS:0003 = FF (SP) = 0010 Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5
0058	MOV DX, [BP+02]	8B5602	(BP) = 0010 (SP) = 0010	(BP) = 0010 (SP) = 0010
005B	RET FAR 0002	CA0200	(DX) = 01F4 (BP) = 0010 Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5	(DX) = 01F4 (BP) = 0010 Stack: +0 19F5 Stack: +2 0000 Stack: +4 0000 Stack: +6 0000

Выводы.

БЫЛИ ОСВОЕНЫ ПРИНЦИПЫ ОТЛАДКИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ АССЕМБЛЕР ПРОЦЕССОРА INTEL X86, усвоены знания в области регистровой адресации.

ПРИЛОЖЕНИЕ А

ИСПРАВЛЕННЫЙ КОД

Название файла: main.asm

; Программа изучения режимов адресации процессора IntelX86

;EOL EQU '\$' ; warning A4031: Operand type must match

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 11,12,13,14,18,17,16,15

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

matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-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 ; error A2009: Symbol not defiend: EOL
mov bh,n2
; Прямая адресация
mov mem2,n2
mov bx,OFFSET vec1
mov mem1,ax
; Косвенная адресация
mov al,[bx]
;mov mem3,[bx] ; (1) error A2052: Improper operand type
; Базированная адресация
mov al,[bx]+3
mov cx,3[bx]
; Индексная адресация
mov di,ind
mov al,vec2[di]
;mov cx,vec2[di] ; (2) warning A4031: Operand types must match
; Адресация с базированием и индексированием
mov bx,3
mov al,matr[bx][di]
;mov cx,matr[bx][di] ; (3) warning A4031: Operand types must match
;mov ax,matr[bx*4][di] ; (4) error A2055: Illegal register value
; 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] ; (5) error A2046: Multiple base register
;mov ax,matr[bp+di+si] ; (6) error A2047: Multiple index register
; Использование сегмента стека
push mem1
push mem2
mov bp,sp
mov dx,[bp]+2
ret 2
Main ENDP ; (7) error A2006: Phase error between passes
CODE ENDS
END Main

```