

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 20:36:5

Page 1-1

```

; Программа изучения режи
; ов адресации процессора I
ntelX86

= 0024          EOL EQU '$' ; warning A4031: Operand type must
match

= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50

; Стек программы
0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
            ???
            ]

0018          AStack ENDS

; Данные программы
0000          DATA SEGMENT

; Директивы описания данн
; x

0000 0000      mem1 DW 0
0002 0000      mem2 DW 0
0004 0000      mem3 DW 0
0006 0B 0C 0D 0E 12 11  vec1 DB 11,12,13,14,18,17,16,15
            10 0F
000E 0A 14 F6 EC 1E 28  vec2 DB 10,20,-10,-20,30,40,-30,-40
            E2 D8
0016 01 02 FC FD 03 04  matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-6,-5
            FE FF 05 06 F8 F9
            07 08 FA FB
```



```

0026                                DATA ENDS

                                ; Код программы
0000                                CODE SEGMENT
                                ASSUME CS:CODE, DS:DATA, SS:AStack
                                ; Головная процедура
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
                                ; ПРОВЕРКА РЕЖИМОВ АДРЕСА □
                                □ ИИ НА УРОВНЕ СМЕЩЕНИИ
                                ; Регистровая адресация
0009 B8 01F4                            mov ax,n1
000C 8B C8                            mov cx,ax
000E B3 24                            mov bl,EOL ; error A2009: Symbol not defiend: E
                                OL
0010 B7 CE                            mov bh,n2
                                ; Прямая адресация
0012 C7 06 0002 R FFCE                mov mem2,n2
0018 BB 0006 R                        mov bx,OFFSET vec1
001B A3 0000 R                        mov mem1,ax
Microsoft (R) Macro Assembler Version 5.10                10/21/20 20:36:5
                                Page 1-2

                                ; Косвенная адресация
001E 8A 07                            mov al,[bx]
                                mov mem3,[bx] ; (1) error A2052: Improper opera
                                nd type
main_err.asm(44): error A2052: Improper operand type
                                ; Базирующая адресация
0020 8A 47 03                        mov al,[bx]+3
0023 8B 4F 03                        mov cx,3[bx]

```

```

; Индексная адресация
0026 BF 0002          mov di,ind
0029 8A 85 000E R     mov al,vec2[di]
002D 8B 8D 000E R     mov cx,vec2[di] ; (2) warning A4031: Operand ty
                        pes must match
main_err.asm(51): warning A4031: Operand types must match
; Адресация с базирование□
□ и индексированием
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] ; (3) warning A4031: Operan
                        d types must match
main_err.asm(55): warning A4031: Operand types must match
003C 8B 85 0022 R     mov ax,matr[bx*4][di] ; (4) error A2055: Illega
                        l register value
main_err.asm(56): error A2055: Illegal register value
; ПРОВЕРКА РЕЖИМОВ АДРЕСА□
□ИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмент
a
; ----- вариант 1
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
; ----- вариант 2
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
; ----- вариант 3
0054 BF 0002          mov di,ind
0057 26: 89 01        mov es:[bx+di],ax
; ----- вариант 4
005A 8B EC            mov bp,sp

```

005C 3E: 8B 86 0016 R mov ax,matr[bp+bx] ; (5) error A2046: Multiple
base register

main_err.asm(75): error A2046: Multiple base registers

0061 3E: 8B 83 0016 R mov ax,matr[bp+di+si] ; (6) error A2047: Multip
le index register

main_err.asm(76): error A2047: Multiple index registers

; Использование сегмента □

□тека

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 ; (7) error A2006: Phase error betwee
n passes

main_err.asm(83): error A2006: Phase error between passes

0076 CODE ENDS

Microsoft (R) Macro Assembler Version 5.10 10/21/20 20:36:5

Page 1-3

END Main

Microsoft (R) Macro Assembler Version 5.10 10/21/20 20:36:5

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	
MAIN	F PROC	0000	CODE Length = 0076
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_err	
@VERSION	TEXT	510	

85 Source Lines

85 Total Lines

19 Symbols

47784 + 459473 Bytes symbol space free

2 Warning Errors

5 Severe Errors

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

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

```