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. 9382 _____ Кодуков А.В.

Преподаватель _____ Ефремов М.А .

Санкт-Петербург
2020

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:

Объяснение ошибок

1) mov mem3,[bx]

adr.asm(42): error A2052: Improper operand type

Невозможно читать и писать память одной командой, необходимо воспользоаться присваиванием через регистр

2) mov cx,vec2[di]

adr.asm(49): warning A4031: Operand types must match

Размер cx - 2 байта, vec2 - 1 байт

3) mov cx,matr[bx][di]

adr.asm(53): warning A4031: Operand types must match

Размер cx - 2 байта, matr1 - 1 байт

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

adr.asm(54): error A2055: Illegal register value

Нельзя умножать регистры

5) mov ax,matr[bp+bx]

adr.asm(73): error A2046: Multiple base registers

2 обращения к base регистрам

6) mov ax,matr[bp+di+si]

adr.asm(74): error A2047: Multiple index registers

2 обращения к index регистрам

7) push mem1 push mem2

adr.asm(81): error A2006: Phase error between passes

Вершина стека должна содержать смещение и сегмент начала PSP

Отладка

CS : 1A0A, DS : 19F5, ES : 19F5, SS : 1A05

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(DS) = 19F5 (SP) = 0018 (IP) = 0000 Stack: +0 0000 +2 0000	(DS) = 19F5 (SP) = 0016 (IP) = 0001 Stack: +0 19F5 +2 0000
0001	SUB AX, AX	2BC0	(AX) = 0000 (IP) = 0001	(AX) = 0000 (IP) = 0003
0003	PUSH AX	50	(AX) = 0000 (SP) = 0016 (IP) = 0003 Stack: +0 19F5 +2 0000	(AX) = 0000 (SP) = 0014 (IP) = 0004 Stack: +0 0000 +2 19F5
0004	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0004	(AX) = 1A07 (IP) = 0007
0007	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 0007	(AX) = 1A07 (DS) = 1A07 (IP) = 0009

0009	MOV AX, 01F4	B8F401	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	MOV CX, AX	8BC8	(AX) = 01F4 (CX) = 00A8 (IP) = 000C	(AX) = 01F4 (CX) = 01F4 (IP) = 000E
000E	MOV BL, 24	B324	(BX) = 0000 (IP) = 000E	(BX) = 0024 (IP) = 0010
0010	MOV BH, CE	B7CE	(BX) = 0024 (IP) = 0010	(BX) = CE24 (IP) = 0012
0012	MOV [0002], FFCE	C7060200CEFF	(IP) = 0012 DS 0000: 2 00 3 00	(IP) = 0018 DS 0000: 2 CE 3 FF
0018	MOV BX, 0006	BB0600	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	MOV [0000], AX	A30000	(IP) = 001B (AX) = 01F4 DS 0000: 0 00 1 00	(IP) = 001E (AX) = 01F4 DS 0000: 0 F4 1 01
001E	MOV AL, [BX]	8A07	(AX) = 01F4 (BX) = 0006 (IP) = 001E DS 0000: 6 26	(AX) = 0126 (BX) = 0006 (IP) = 0020 DS 0000: 6 26
0020	MOV AL, [BX + 03]	8A4703	(AX) = 0126 (BX) = 0006 (IP) = 0020 DS 0000: 9 23	(AX) = 0123 (BX) = 0006 (IP) = 0023 DS 0000: 9 23
0023	MOV CX, [BX + 03]	8B4F03	(CX) = 01F4 (BX) = 0006 (IP) = 0023 DS 0000: 9 23 A 1F	(CX) = 1F23 (BX) = 0006 (IP) = 0026 DS 0000: 9 23 A 1F
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E + DI]	8A850E00	(AX) = 0123 (DI) = 0002 (IP) = 0029	(AX) = 01BA (DI) = 0002 (IP) = 002D

			DS 0010: 0 BA	DS 0010: 0 BA
002D	MOV BX, 0003	BB0300	(BX) = 0006 (IP) = 002D	(BX) = 0003 (IP) = 0030
0030	MOV AL, [0016 + BX + DI]	8A811600	(AX) = 01BA (BX) = 0003 (DI) = 0002 (IP) = 0030 DS 0010 B F9	(AX) = 01F9 (BX) = 0003 (DI) = 0002 (IP) = 0034 DS 0010 B F9
0034	MOV AX, 1A07	B8071A	(AX) = 01F9 (IP) = 0034	(AX) = 1A07 (IP) = 0037
0037	MOV ES, AX	8EC0	(ES) = 19F5 (AX) = 1A07 (IP) = 0037	(ES) = 1A07 (AX) = 1A07 (IP) = 0039
0039	MOV AX, ES : [BX]	268B07	(AX) = 1A07 (ES) = 1A07 (BX) = 0003 DS 0000 3 FF 4 00 (IP) = 0039	(AX) = 00FF (ES) = 1A07 (BX) = 0003 DS 0000 3 FF 4 00 (IP) = 003C
003C	MOV AX, 0000	B80000	(AX) = 00FF (IP) = 003C	(AX) = 0000 (IP) = 003F
003F	MOV ES, AX	8EC0	(ES) = 1A07 (AX) = 0000 (IP) = 003F	(ES) = 0000 (AX) = 0000 (IP) = 0041
0041	PUSH DS	1E	(SP) = 0014 (DS) = 1A07 (IP) = 0041 Stack: +0 0000 +2 19F5	(SP) = 0012 (DS) = 1A07 (IP) = 0042 Stack: +0 1A07 +2 0000
0042	POP ES	07	(SP) = 0012 (ES) = 0000 (IP) = 0042 Stack: +0 1A07 +2 0000	(SP) = 0014 (ES) = 1A07 (IP) = 0042 Stack: +0 0000 +2 19F5
0043	MOV CX, ES : [BX – 01]	268B4FFF	(CX) = 1F23 (ES) = 1A07	(CX) = FFCE (ES) = 1A07

			(BX) = 0003 (IP) = 0043 DS 0000 2 CE 3 FF	(BX) = 0003 (IP) = 0047 DS 0000 2 CE 3 FF
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) = 1A07 (BX) = 0003 (DI) = 0002 (AX) = FFCE (IP) = 004B DS 5 00 6 26	(ES) = 1A07 (BX) = 0003 (DI) = 0002 (AX) = FFCE (IP) = 004E DS 5 CE 6 FF
004E	MOV BP, SP	8BEC	(BP) = 0000 (SP) = 0014 (IP) = 004E	(BP) = 0014 (SP) = 0014 (IP) = 0050
0050	MOV DX, [BP + 02]	8B5602	(DX) = 0000 (BP) = 0014 (IP) = 0050 Stack: +2 19F5	(DX) = 19F5 (BP) = 0014 (IP) = 0050 Stack: +2 19F5
0053	RET Far 0002	CA0200	(CS) = 1A01 (IP) = 0053 Stack: +0 0000 +2 19F5	(CS) = 19F5 (IP) = 0000 Stack: +0 0000 +2 0000
0000	INT 20	CD20		

Вывод:

В ходе выполнения работы были изучены режимы адресации в intel 8086, объяснены ошибки, возникшие на этапе компиляции, а также получены результаты отладки программы с различными способами адресации на уровне смещений и с учетом регистров.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

adr.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 38,37,36,35,31,32,33,34
vec2 DB 70,80,-70,-80,50,60,-50,-60
matr DB -2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-6,-5,1,2
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] Невозможно читать и писать память одной командой,
необходимо воспользоаться присваиванием через регистр
; Базированная адресация
      mov al,[bx]+3
      mov cx,3[bx]
; Индексная адресация
      mov di,ind
      mov al,vec2[di]
      ;mov cx,vec2[di] Размер cx - 2 байта, vec2 - 1 байт
; Адресация с базированием и индексированием
      mov bx,3
      mov al,matr[bx][di]
      ;mov cx,matr[bx][di] Размер cx - 2 байта, matr - 1 байт
      ;mov ax,matr[bx*4][di] Нельзя умножать регистры
; 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] 2 обращения к base регистрам
      ;mov ax,matr[bp+di+si] 2 обращения к index регистрам
; Использование сегмента стека
      ;push mem1
      ;push mem2 Вершина стека должна содержать смещение и сегмент начала PSP
      mov bp,sp
      mov dx,[bp]+2
```

```
      ret 2
Main ENDP
CODE ENDS
      END Main
```

ПРИЛОЖЕНИЕ В
ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

```
  Microsoft (R) Macro Assembler Version 5.10                  9/30/20 24:48:09
                                                              Page     1-1


                        ; Программа изучения режимов адресации процессо
                        pa IntelX86
 = 0024                       EOL EQU '$'
 = 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 данных
 0000  0000             mem1 DW 0
 0002  0000             mem2 DW 0
 0004  0000             mem3 DW 0
 0006  26 25 24 23 1F 20      vec1 DB 38,37,36,35,31,32,33,34
       21 22
 000E  46 50 BA B0 32 3C      vec2 DB 70,80,-70,-80,50,60,-50,-60
       CE C4
 0016  FE FF 05 06 F8 F9      matr DB -2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-6,-5,1,2
       03 04 FC FD 07 08
       FA FB 01 02
 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 АДРЕСАЦИИ НА УРОВНЕ СМЕЩЕНИЙ
                        ; Регистровая адресация
0009  B8 01F4                        mov ax,n1
000C  8B C8                          mov cx,ax
000E  B3 24                          mov bl,EOL
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
                        ; Косвенная адресация
001E  8A 07                          mov al,[bx]
                              ;mov mem3,[bx] Невозможно читать и писа
                        ть память одной командой, необходимо воспользоа
                        ться присваиванием через регистр
                        ; Базированная адресация
0020  8A 47 03                       mov al,[bx]+3
Microsoft (R) Macro Assembler Version 5.10                  9/30/20 24:48:09
                                                             Page     1-2


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]
                              ;mov cx,vec2[di] Размер cx - 2 байта, v
                        ec2 - 1 байт
                        ; Адресация с базированием и индексированием
002D  BB 0003                        mov bx,3
0030  8A 81 0016 R                   mov al,matr[bx][di]
                              ;mov cx,matr[bx][di] Размер cx - 2 байт
                        а, matr - 1 байт
                              ;mov ax,matr[bx*4][di] Нельзя умножать
                        регистры
                        ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
                        ; Переопределение сегмента
                        ; ------ вариант 1
```

```
0034  B8 ---- R             mov ax, SEG vec2
0037  8E C0                      mov es, ax
0039  26: 8B 07             mov ax, es:[bx]
003C  B8 0000                      mov ax, 0
                       ; ------ вариант 2
003F  8E C0                      mov es, ax
0041  1E                   push ds
0042  07                   pop es
0043  26: 8B 4F FF               mov cx, es:[bx-1]
0047  91                   xchg cx,ax
                       ; ------ вариант 3
0048  BF 0002                    mov di,ind
004B  26: 89 01            mov es:[bx+di],ax
                       ; ------ вариант 4
                             ;mov bp,sp
                             ;mov ax,matr[bp+bx] 2 обращения к base
                       регистрам
                             ;mov ax,matr[bp+di+si] 2 обращения к in
                       dex регистрам
                       ; Использование сегмента стека
                             ;push mem1
                             ;push mem2 Вершина стека должна содержа
                       ть смещение и сегмент начала PSP
004E  8B EC                      mov bp,sp
0050  8B 56 02                   mov dx,[bp]+2
0053  CA 0002                    ret 2
0056                   Main ENDP
0056                   CODE ENDS
                             END Main
```


Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0056	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 = 0056
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  adr
@VERSION . . . . . .               TEXT  510


     83 Source  Lines
     83 Total   Lines
     19 Symbols

  47832 + 459428 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```