**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.**

**Вариант №9**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8381 |  | Киреев К.А. |
| Преподаватель |  | Кирьянчиков В.А. |

Санкт-Петербург

2019

**Цель работы.**

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера. Разобраться в используемых режимах адресации и получаемых результатах.

**Задание.**

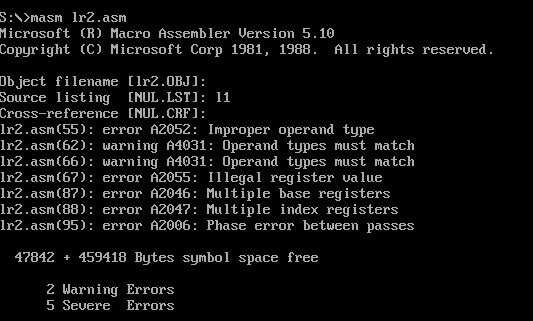
Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу lr2\_comp.asm на Ассемблере, которая в автоматическом режиме выполняться не должна, так как не имеет самостоятельного функционального назначения, а только тестирует режимы адресации. Поэтому ее выполнение 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.

**Ход работы.**

1. Изменение набора значений исходных данных (массивов) vec1, vec2 и matr из файла lr2.dat, согласно своему варианту.
2. Трансляция программы с созданием файла диагностических сообщений. Объяснение обнаруженных ошибок и предупреждений и закомментирование операторов с ошибками в тексте программы.



* Ошибка lr2.asm(55): error A2052: Improper operand type (Неверный тип операнда)

**Строка 55**: mov mem3, [bx]

Тип операнда, нельзя читать из памяти и писать в память одной командой. В данном случае необходимо перевести информацию из памяти в регистр, а затем уже перевести информацию из регистра в необходимый сегмент.

* Предупреждение lr2.asm(62): warning A4031: Operand types must match (Несоответствие типов операндов)

**Строка 62**: mov сx, vec2[di]

Несоответствие типов операндов, cx – 1 слово, элемент vec2 – 1 байт.

* Предупреждение lr2.asm(66): warning A4031: Operand types must match (Несоответствие типов операндов)

**Строка 66:** mov cx, matr[bx][di]

Несоответствие типов операндов, cx – 1 слово, элемент matr – 1 байт.

* Ошибка lr2.asm(67): error A2055: Illegal register value (Незаконное использование регистра)

**Строка 67**: mov ax,matr[bx\*4][di]

Здесь используется базово-индексная адресация. Такая форма адресации используется в тех случаях, когда в регистре находится адрес начала структуры данных, а доступ надо осуществить к какому-нибудь элементу этой структуры. При данном типе адресации надо сначала изменить значение регистра, затем уже переводить информацию.

* Ошибка lr2.asm(87): error A2046: Multiple base registers (несколько базовых регистров)

**Строка 87**: mov ax,matr[bp+bx]

Нельзя складывать регистры bp и bx. Так как здесь оба регистра базовые, надо сначала сложить значения регистров, и затем уже передавать информацию указателю из одного регистра. Необходимо сначала в регистр bp занести общую сумму, затем уже производить смещение.

* Ошибка lr2.asm(88): error A2047: Multiple index registers (несколько индексных регистров)

**Строка 88**: mov ax,matr[bp+di+si]

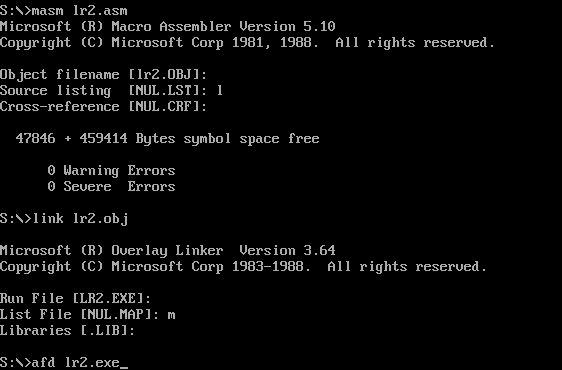
Нельзя складывать регистры di и si. Так как здесь два индексных регистра, надо сначала сложить значения регистров, и затем уже передавать информацию указателю из одного регистра. Необходимо сначала в регистр di занести общую сумму, затем уже производить смещение.

* Ошибка lr2.asm(95): error A2006: Phase error between passes

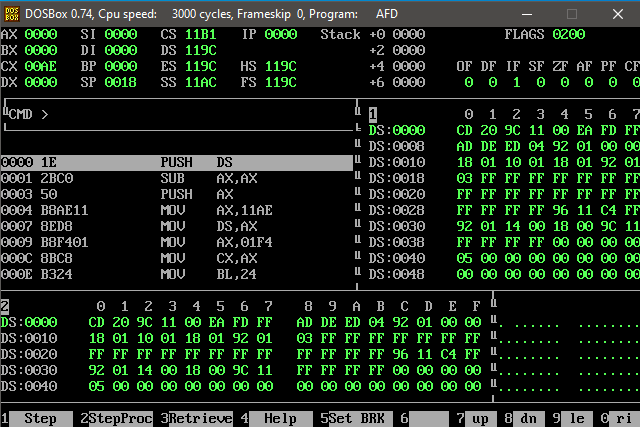
**Строка 95:** Main ENDP

Данная ошибка свидетельствует о том, что в функции main содержатся ошибки.

3. Повторная трансляция программы и компоновка загрузочного модуля.



4. Выполнение программы в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символьный код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
| До выполнения | После выполнения |
| 0000 | PUSH DS | 1E | (SP)=0018  (IP)=0000 | (SP)=0016  (IP)=0001 |
| 0001 | SUB AX. AX | 2BC0 | (IP)=0001 | (IP)=0003 |
| 0003 | PUSH AX | 50 | (SP)= 0016  (IP)= 0003 | (SP)= 0014  (IP)= 0004 |
| 0004 | MOV AX, 11 AE | B8AE11 | (AX)=0000  (IP)=0004 | (AX)= 11AE  (IP)=0007 |
| 0007 | MOV DS,AX | 8ED8 | (DS)=119С  (IP)= 0007 | (DS)=11AE  (IP)= 0009 |
| 0009 | MOV AX, 01F4 | B8F401 | (AX)=11AE  (IP)=0009 | (AX)= 01F4  (IP)= 000C |
| 000C | MOV CX,AX | 8BC8 | (IP)=000C  (CX)=00AE | (IP)=000E  (CX)=01F4 |
| 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 | (IP)=0018 |
| 0018 | MOV BX,0006 | BB0600 | (BX)=CE24  (IP)=0018 | (BX)=0006  (IP)=001B |
| 001B | MOV [0000],AX | A30000 | (IP)=001B | (IP)=001E |
| 001E | MOV AL,[BX] | 8A07 | (AX)=01F4  (IP)=001E | (AX)=011F  (IP)=0020 |
| 0020 | MOV AL,[BX+03] | 8A4703 | (IP) = 0020  (AX) = 011F | (IP)= 0023  (AX) = 0122 |
| 0023 | MOV CX, [BX+03] | 8B4F03 | (CX) = 01F4  (IP) = 0023 | (CX) = 2622  (IP) = 0026 |
| 0026 | MOV DI, 0002 | BF0200 | (DI) = 0000  (IP) = 0026 | (DI) = 0002  (IP) = 0029 |
| 0029 | MOV AL, [DI+ 000E] | 8A850E00 | (AX) = 0122 (IP) = 0029 | (AX)= 01CE  (IP)= 002D |
| 002D | MOV BX, 0003 | BB03000 | (IP) = 002D  (BX) = 0006 | (IP) = 0030  (BX) = 0003 |
| 0030 | MOV AL, [BX+DI+0016] | 8A811600 | (IP) = 0030  (AX) = 01CE | (IP) = 0034  (AX) = 01FF |
| 0034 | MOV AX, 11 AE | B8AE11 | (AX) = 0104  (IP)= 0034 | (AX) = 11AE  (IP)= 0037 |
| 0037 | MOV ES, AX | 8EC0 | (ES) = 119C  (IP)= 0037 | (ES) = 11AE  (IP)= 0039 |
| 0039 | MOV AX, ES:[BX] | 268B07 | (AX) = 11AE  (IP) = 0039 | (AX)= 00FF  (IP) = 003C |
| 003C | MOV AX, 0000 | B80000 | (AX)= 00FF  (IP)= 003C | (AX)=0000  (IP) = 003F |
| 003F | MOV ES, AX | 8EC0 | (ES) = 11AE  (IP)= 003F | (ES)= 0000  (IP)= 0041 |
| 0041 | PUSH DS | 1E | (IP)= 0041  (SP)= 0014 | (IP)= 0042  (SP)= 0012 |
| 0042 | POP ES | 07 | (SP)= 0012  (ES)=0000  (IP)= 0042 | (SP) = 0014  (ES)=11AE  (IP)= 0043 |
| 0043 | MOV CX, ES:[BX—01] | 268B4FFF | (CX) = 2622  (IP) = 0043 | (CX)= FFCE  (IP)= 0047 |
| 0047 | XCHG AX, CX | 91 | (AX) = 0000  (CX) = FFCE  (IP)=0047 | (AX) = FFCE  (CX) = 0000  (IP)=0048 |
| 0048 | MOV DI, 0002 | BF0200 | (IP) = 0048 | (IP) = 004B |
| 004B | MOV ES:[BX+DI], AX | 268901 | (IP) = 004B | (IP) = 004E |
| 004E | MOV BP, SP | 8BEC | (IP) = 004E  (BP) = 0010 | (IP) = 0050  (BP) = 0014 |
| 0050 | PUSH [0000] | FF360000 | (IP) = 0050  (SP)=0014 | (IP) = 0054  (SP)=0012 |
| 0054 | PUSH [0002] | FF360200 | (IP) = 0054  (SP) = 0012 | (IP) = 0058  (SP) = 0010 |
| 0058 | MOV BP, SP | 8BEC | (IP) = 0058  (BP) = 0014 | (IP) = 005A  (BP) = 0010 |
| 005A | MOX DX, [BP+02] | 8B5602 | (IP) = 005A  (DX) = 0000 | (IP) = 005D  (DX) = 01F4 |
| 005D | RET Far | CB | (IP) = 005D  (SP) = 0010  (CS)=11B1 | (IP) = FFCE  (SP)= 0014  (CS)=01F4 |

Изменение стека

|  |  |
| --- | --- |
| **Адрес команды** | **Стек** |
| Начальное положение | +0 0000  +2 0000  +4 0000  +6 0000 |
| 0000 | +0 119С  +2 0000  +4 0000  +6 0000 |
| 0003 | +0 0000  +2 119С  +4 0000  +6 0000 |
| 0041 | +0 11AE  +2 0000  +4 119C  +6 0000 |
| 0042 | +0 0000  +2 119C  +4 0000  +6 0000 |
| 0050 | +0 01F4  +2 0000  +4 119C  +6 0000 |
| 0054 | +0 FFCE  +2 01F4  +4 0000  +6 119C |
| 005D | +0 0000  +2 119С  +4 0000  +6 0000 |

**Выводы.**

В ходе выполнения лабораторной работы были получены основные навыки программирования на ассемблере, изучены основные режимы адресации памяти.

**Приложение А. Код программы lr2.asm**

; Учебная программа лабораторной работы №2 по дисциплине "Архитектура компьютера"

;

;

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 31,32,33,34,38,37,36,35

vec2 DB 50,60,-50,-60,70,80,-70,-80

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

; Базированная адресация

mov al,[bx]+3

mov cx,3[bx]

; Индексированная 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 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

CODE ENDS

Main ENDP

END Main

**Приложение Б. Листинг успешной трансляции**

**программы с закомментированными ошибочными**

**операторами**

Microsoft (R) Macro Assembler Version 5.10 9/26/19 23:32:55

Page 1-1

; Учебная программа лабораторной работы №2 по

дисциплине "Архитектура компьютера"

;

;

= 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 1F 20 21 22 26 25 vec1 DB 31,32,33,34,38,37,36,35

24 23

000E 32 3C CE C4 46 50 vec2 DB 50,60,-50,-60,70,80,-70,-80

BA B0

0016 FC FD 07 08 FE FF matr DB -4,-3,7,8,-2,-1,5,6,-8,-7,3,4,-

6,-5,1,2

05 06 F8 F9 03 04

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

Microsoft (R) Macro Assembler Version 5.10 9/26/19 23:32:55

Page 1-2

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]

; Базированная адресация

0020 8A 47 03 mov al,[bx]+3

0023 8B 4F 03 mov cx,3[bx]

; Индексированная адресация

0026 BF 0002 mov di,ind

0029 8A 85 000E R mov al,vec2[di]

;mov cx,vec2[di]

; 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]

;mov ax,matr[bx\*4][di]

; ПРОВЕРКА 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

004E 8B EC mov bp,sp

;mov ax,matr[bp+bx]

;mov ax,matr[bp+di+si]

; Использование сегмента стека

0050 FF 36 0000 R push mem1

0054 FF 36 0002 R push mem2

0058 8B EC mov bp,sp

005A 8B 56 02 mov dx,[bp]+2

005D CB ret

005E CODE ENDS

0000 Main ENDP

END Main

Microsoft (R) Macro Assembler Version 5.10 9/26/19 23:32:55

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

EOL . . . . . . . . . . . . . . NUMBER 0024

IND . . . . . . . . . . . . . . NUMBER 0002

MAIN . . . . . . . . . . . . . . F PROC 0000 CODE Length = 0000

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 lr2

@VERSION . . . . . . . . . . . . TEXT 510

97 Source Lines

97 Total Lines

19 Symbols

47846 + 459414 Bytes symbol space free

0 Warning Errors

0 Severe Errors