**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 основной памяти**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8382 |  | Мирончик П.Д. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2019

**Цель работы.**

Изучение различных видов адресации на языке ассемблер.

**Ход работы**

Данные массивов *vec1*, *vec2* и *matr* были заменены данными, полученными соответственно варианту.

В процессе первой трансляции программы были обнаружены следующие ошибки и предупреждения:

55: mov mem3,[bx]

main.asm(55): error A2052: Improper operand type– неверный тип операндов: оператор mov может выполнять операцию либо на запись, либо на чтение, но писать из памяти в память не может.

62: mov cx,vec2[di]

main.asm(62): warning A4031: Operand types must match– размер *ax*  2 байта, а размер элемента *vec2* – 1 байт

66: mov cx,matr[bx][di]

main.asm(66): warning A4031: Operand types must match - размер cx 2 байта, а размер элемента vec2 – 1 байт

67: mov ax,matr[bx\*4][di]

main.asm(67): error A2055: Illegal register value - При данном типе адресации надо сначала изменить значение регистра, затем уже переводить информацию.

87: mov ax,matr[bp+bx]

main.asm(87): error A2046: Multiple base registers - Нельзя складывать регистры bp и bx . В базированной адресации необходимо указывать базовый регистр, затем производить смещение с помощью индексного. Так как здесь оба базовые, надо сначала сложить значения регистров, затем уже передавать информацию указателю из одного регистра.

88: mov ax,matr[bp+di+si]

main.asm(88): error A2047: Multiple index registers - Нельзя складывать регистры bp и bx, необходимо сначала а регистр di занести общую сумму, затем уже производить смещение.

main.asm(95): error A2006: Phase error between passes

Табл.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес команды** | **Символический код команды** | **16-ричный код команды** | **Содержимое регистров и ячеек памяти до|после выполнения** | |
| 0000 | push DS | 1E | DS=119C  IP=0000  SP=0018  stack:  0000 | DS=119C  IP=0001  SP=0016  stack:  119C |
| 0001 | sub AX,AX | 2BC0 | IP=0001  AX=0000 | IP=0003  AX=0000 |
| 0003 | push AX | 50 | IP=0003  SP=0016  stack:  119C | IP=0004  SP=0014  stack:  0000  119C |
| 0004 | mov AX,11AE | B8AE11 | IP=0004  AX=0000 | IP=0007  AX=11AE |
| 0007 | mov DS,AX | 8ED8 | IP=0007  DS=119C  AX=11AE | IP=0009  DS=11AE  AX=11AE |
| 0009 | mov AX,01F4 | B8F401 | IP=0009  AX=11AE | IP=000C  AX=01F4 |
| 000C | mov CX,AX | 8BC8 | IP=000C  AX=01F4  CX=00B8 | IP=000E  AX=01F4  CX=01F4 |
| 000E | mov BL,24 | B409 | IP=000E  BX=0000 | IP=0010  BX=0024 |
| 0010 | mov BH,CE | B7CE | IP=0010  BX=0024 | IP=0012  BX=CE24 |
| 0012 | mov [0002],FFCE | C7060200CEFF | IP=0012 | IP=0018 |
| 0018 | mov BX,0006 | BB0600 | IP=0018  BX=CE24 | IP=001B  BX=0006 |
| 001B | mov [0000],AX | A30000 | IP=001B  AX=01F4 | IP=000E  AX=01F4 |
| 001E | mov AL,[BX] | 8A07 | IP=001E  AX=01F4  BX=0006 | IP=0020  AX=0101  BX=0006 |
| 0020 | mov AL,[BX+03] | 8A4703 | IP=0020  AX=0101  BX=0006 | IP=0023  AX=0104  BX=0006 |
| 0023 | mov CX,[BX+03] | 8B4F03 | IP=0023  CX=01F4  BX=0006 | IP=0026  CX=0804  BX=0006 |
| 0026 | mov DI,0002 | BF0200 | IP=0026  DI=0000 | IP=0029  DI=0002 |
| 0029 | mov AL,[DI+000E] | 8A850E00 | IP=0029  AX=0104  DI=0002 | IP=002D  AX=010A  DI=0002 |
| 002D | mov CX,[DI+000E] | 8B8D0E00 | IP=002D  CX=0804  DI=0002 | IP=0031  CX=140A  DI=0002 |
| 0031 | mov BX,0003 | BB0300 | IP=0031  BX=0006 | IP=0034  BX=0003 |
| 0034 | mov AL,[BX+DI+0016] | 8A811600 | IP=0034  AX=010A  BX=0003  DI=0002 | IP=0038  AX=01FD  BX=0003  DI=0002 |
| 0038 | mov CX,[BX+DI+0016] | 8B891600 | IP=0038  CX=140A  BX=0003  DI=0002 | IP=003C  CX=FEFD  BX=0003  DI=0002 |
| 003C | mov AX,11AE | B8AE11 | IP=003C  AX=01FD | IP=003F  AX=11AE |
| 003F | mov ES,AX | 8EC0 | IP=003F  ES=119C  AX=11AE | IP=0041  ES=11AE  AX=11AE |
| 0041 | mov AX,ES:[BX] | 268B07 | IP=0041  AX=11AE  ES=11AE  BX=0003 | IP=0044  AX=00FF  ES=11AE  BX=0003 |
| 0044 | mov AX,0000 | B80000 | IP=0044  AX=00FF | IP=0047  AX=0000 |
| 0047 | mov ES,AX | 8EC0 | IP=0047  ES=11AE  AX=0000 | IP=0049  ES=0000  AX=0000 |
| 0049 | push DS | 1E | IP=0049  DS=11AE  SP=0014  stack:  0000  119C | IP=004A  DS=11AE  SP=0012  stack:  11AE  0000  119C |
| 004A | pop ES | 07 | IP=004A  ES=0000  SP=0012  Stack;  11AE  0000  119C | IP=004B  ES=11AE  SP=0014  Stack:  0000  119C |
| 004B | mov CX,ES:[BX-01] | 268B4FFF | IP=004B  CX=FEFD  ES=11AE  BX=0003 | IP=004F  CX=FFCE  ES=11AE  BX=0003 |
| 004F | xchg AX,CX | 91 | IP=004F  AX=0000  CX=FFCE | IP=0050  AX=FFCE  CX=0000 |
| 0500 | mov DI,0002 | BF0200 | IP=0050  DI=0002 | IP=0053  DI=0002 |
| 0053 | mov ES:[BX+DI],AX | 268901 | IP=0053  ES=11AE  BX=0003  DI=0002  AX=FFCE | IP=0056  ES=11AE  BX=0003  DI=0002  AX=FFCE |
| 0056 | mov BP,SP | 8BEC | IP=0056  BP=0000  SP=0014 | IP=0058  BP=0014  SP=0014 |
| 0058 | push [0000] | FF360000 | IP=0058  SP=0014  Stack:  0000  119C | IP=005C  SP=0012  Stack:  01F4  0000  119C |
| 005C | push [0002] | FF360200 | IP=005C  SP=0012  Stack:  01F4  0000  119C | IP=0060  SP=0010  Stack:  FFCE  01F4  000  119C |
| 0060 | mov BP,SP | 8BEC | IP=0060  BP=0014  SP=0010 | IP=0062  BP=0010  SP=0010 |
| 0062 | mov DX,[BP+02] | 8B5602 | IP=0062  DX=0000  BP=0010 | IP=0065  DX=01F4  BP=0010 |
| 0065 | ret Far 0002 | CA0200 | IP=0065  SP=0010  Stack:  FFCE  01F4  0000  119C | IP=FFCE  SP=0016  Stack:  119C |

**Выводы**

В ходе выполнения данной работы был освоен навык работы с программами на языке Ассемблер. Изучены возможности адресации языка, выявлены основные ошибки при адресации и отработаны пути из устранения.

**Вложение А**

; 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 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]

; 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 2

Main ENDP

CODE ENDS

END Main

**Вложение B**

Microsoft (R) Macro Assembler Version 5.10 9/25/19 22:12:10

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 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

; 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 СМЕЩЕНИ

Й

; Регистровая адресация

Microsoft (R) Macro Assembler Version 5.10 9/25/19 22:12:10

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]

main.asm(55): error A2052: Improper operand type

; 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]

002D 8B 8D 000E R mov cx,vec2[di]

main.asm(62): warning A4031: Operand types must match

; Адресация с базированием 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]

main.asm(66): warning A4031: Operand types must match

003C 8B 85 0022 R mov ax,matr[bx\*4][di]

main.asm(67): error A2055: Illegal register value

; ПРОВЕРКА АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ

; Переопределение сегмента

; ------ 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]

main.asm(87): error A2046: Multiple base registers

0061 3E: 8B 83 0016 R mov ax,matr[bp+di+si]

main.asm(88): error A2047: Multiple index registers

; Использование сегмента стека

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(95): error A2006: Phase error between passes

0076 CODE ENDS

END Main

Microsoft (R) Macro Assembler Version 5.10 9/25/19 22:12:10

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

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

97 Source Lines

97 Total Lines

19 Symbols

47828 + 459432 Bytes symbol space free

2 Warning Errors

5 Severe Errors 0 Severe Errors