**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №2**

**по дисциплине «Организация ЭВМ и систем»**

**Тема: Изучение режимов адресации основной памяти**

|  |  |
| --- | --- |
| Студент гр. 3344 | Тукалкин В.А. |
| Преподаватель | Фирсов М.А. |

Санкт-Петербург

2024

**Цель работы**

Познакомиться со средой DOS и изучить базовые понятия в ассеблере.

**Задание**

1. Получить у преподавателя вариант выбора значений исходных данных (массивов) vec1, vec2 и matr из файла lr2+.dat и занести свои данные вместо значений, указанных в приведённой для образца программе.

2. Протранслировать программу с созданием файла диагностических сообщений и объяснить обнаруженные ошибки (error) и предупреждения (warning). Листинг привести в приложении к отчёту. Закомментировать операторы с ошибками в тексте программы, а операторы с предупреждениями оставить без изменения. Объяснения ошибок и предупреждений должны быть приведены в отчёте по лабораторной работе.

3. Снова протранслировать программу и скомпоновать загрузочный модуль. Учесть, что программа учебная и может выполняться только под отладчиком. В автоматическом режиме она выполняться не должна.

4. Выполнить программу в пошаговом режиме под управлением отладчика с фиксацией содержимого **используемых** регистров и ячеек памяти до и после выполнения каждой команды. Разобраться в используемых режимах адресации и получаемых результатах. Результаты прогона программы под управлением отладчика должны быть представлены в отчёте по лабораторной работе в табличном виде, аналогичном указанному в лаб.работе №1.

Вариант 20:

i vec1 i 39,38,37,36,32,33,34,35

20 i vec2 i 70,90,-70,-90,20,60,-20,-60

i matr i 2,1,-5,-6,8,7,-3,-4,4,3,-7,-8,6,5,-1,-2

**Основные теоретические положения**

Изучение режимов адресации основной памяти на языке ассемблера включает следующие основные теоретические положения:

Адресное пространство:

В процессорах с архитектурой x86 адресное пространство делится на 2 уровня: линейное и физическое. Линейный адрес преобразуется в физический при помощи механизма сегментации и трансляции страниц.

Сегментная организация памяти:

Процессоры x86 используют систему сегментов для организации адресного пространства. Сегменты содержат код, данные и стек. Каждый сегмент имеет свой базовый адрес и размер.

Режимы адресации:

Существует несколько режимов адресации:

* Относительная адресация (relative addressing): Адресата передаются как смещение относительно текущего регистра указателя команд (IP).
* Базовая адресация (base addressing): Использует базовый регистр для образования абсолютного адреса.
* Стековая адресация (stack addressing): Загрузка и сохранение данных происходит через стек.

Разновидности стековых операций:

* Погружение (push): Данные записываются на вершину стека.
* Подъем (pop): Данные извлекаются из вершины стека.

Управление страницами:

Современные процессоры поддерживают управление памятью на уровне страниц. Этот механизм используется для виртуализации памяти и оптимизации работы оперативной памяти.

Механизм защиты:

При использовании сегментов и дескрипторов можно задавать права доступа, что обеспечивает безопасность и изоляцию процессов.

Эмуляция сегментной архитектуры:

В современных версиях Windows и Linux сегментная архитектура эмулируется с помощью других механизмов управления памятью, таких как виртуальные адреса и таблицы страниц.

**Выполнение работы**

1) Согласно вариантам, распределёнными преподавателем, взяты из файла lr2+.dat и заменены на них в файле программы.

Данные согласно варианту 20:

vec1 39, 38, 37, 36, 32, 33, 34, 35

vec2 70, 90, -70, -90, 20, 60, -20, -60

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

2) Протранслирована программа и получены ошибки с предупреждениями, создать файл листинга (lr2\_comp.lst).



Объяснение ошибок:

LR2\_comp.ASM(54): error A2052: Improper operand type

Неподходящий тип операндов

Нельзя читать из памяти и писать в память одной командой

LR2\_comp.ASM(66): error A2055: Illegal register value

Незаконное использование регистра

В данном случае нельзя напрямую масштабировать bx

Несоответствие типов операндов

Размер элементов матрицы ‘matr' 1 байт, а ‘ах' - 2 байта

LR2\_comp.ASM(86): error A2046: Multiple base registers

Слишком много базовых регистров

Нельзя использовать более одного базового регистра

Несоответствие типов операндов

Размер элементов матрицы ‘matr' 1 байт, а ‘ах' - 2 байта

LR2\_comp.ASM(87): error A2047: Multiple index registers

Слишком много индексных регистров

Нельзя использовать более одного индексного регистра

Слишком много регистров

Нельзя использовать более двух регистров

Несоответствие типов операндов

Размер элементов матрицы ‘matr' 1 байт, а ‘ах' - 2 байта

LR2\_comp.ASM(94): error A2006: Phase error between passes

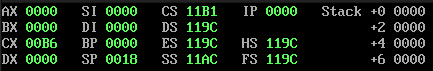
Фазовая ошибка между проходами

Эта ошибка возникает в паре с ошибкой с 54 строки

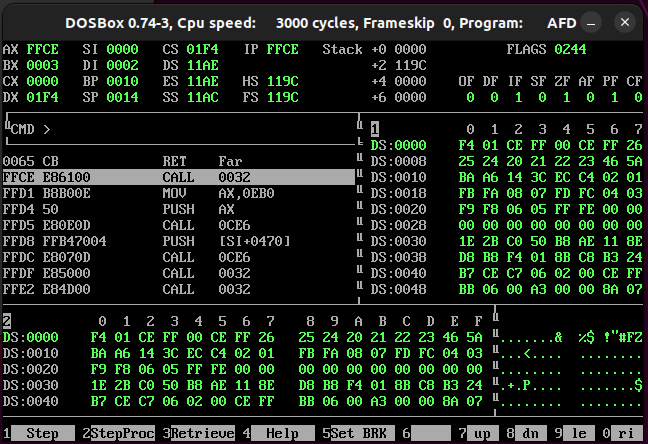
3) Закомментированы строки с ошибками, снова протранслирована программа и скомпонован загрузочный модуль, создан файл листинга (lr2.lst) и карта (lr2.map). Осталось только 2 предупреждения.



4) Выполнение программы под отладчиком:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Cимволический код команды | 16-ричный код команды | Содержание регистров и ячеек памяти | |
| До вып. команды | После вып. команды |
| 0000 | PUSH DS | 1E | SP = 0018  IP = 0000  Stack:  +0 0000  +2 0000  +4 0000  +6 0000 | SP = 0016  IP = 0001  Stack:  +0 119C  +2 0000  +4 0000  +6 0000 |
| 0001 | SUD AX,AX | 2BC0 | IP=0001  AX=0000 | IP=0003  AX=0000 |
| 0003 | PUSH AX | 50 | IP=0003  SP=0016  Stack  +0 119C  +2 0000  +4 0000  +6 0000 | IP=0004  SP=0014  Stack  +0 0000  +2 119C  +4 0000  +6 0000 |
| 0004 | MOV AX,11AE | B8AE11 | IP=0004  AX=0000 | IP=0007  AX=11AE |
| 0007 | MOV DS,AX | 8ED8 | IP=0007  DS=119C | IP=0009  DS=11AE |
| 0009 | MOV AX,01F4 | B8F401 | IP=0007  AX=11AE | IP=000C  AX=01F4 |
| 000C | MOV CX,AX | 8BC8 | IP=000C  CX=00B6 | IP=000E  CX=01F4 |
| 000E | MOV BL,24 | B324 | 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  [0002] = 0000 | IP=0018  [0002] =CEFF |
| 0018 | MOV BX,0006 | BB0600 | IP=0018  BX=CE24 | IP=001B  BX=0006 |
| 001B | MOV [0000],AX | A30000 | IP=001B  [0000] = 0000 | IP=001E  [0000] = CEFF |
| 001E | MOV AL,[BX] | 8A07 | IP=001E  AX=01F4 | IP=0020  AX=0127 |
| 0020 | MOV AL,[BX+03] | 8A4703 | IP=0020  AX=0127 | IP=0023  AX=0124 |
| 0023 | MOV CX,[BX+03] | 8B4F03 | IP=0023  CX=01F4 | IP=0026  CX=2024 |
| 0026 | MOV DI,0002 | BF0200 | IP=0026  DI=0000 | IP=0029  DI=0002 |
| 0029 | MOV AL,[DI+000E] | 8A850E00 | IP=0029  AX=0124 | IP=002D  AX=01BA |
| 002D | MOV CX,[DI+000E] | 8B8D0E00 | IP=002D  CX=2024 | IP=0031  CX=A6BA |
| 0031 | MOV BX,0003 | BB0300 | IP=0031  BX=0006 | IP=0034  BX=0003 |
| 0034 | MOV AL,[BX+DI+0016] | 8A811600 | IP=0034  AX=01BA | IP=0038  AX=0107 |
| 0038 | MOV CX,[BX+DI+0016] | 8B891600 | IP=0038  CX=A6BA | IP=003C  CX=FD07 |
| 003C | MOV AX,11AE | B8AE11 | IP=003C  AX=0107 | IP=003F  AX=11AE |
| 003F | MOV ES,AX | 8EC0 | IP=003F  ES=119C | IP=0041  ES=11AE |
| 0041 | MOV AX,ES:[BX] | 268B07 | IP=0041  AX=11AE | IP=0044  AX=00FF |
| 0044 | MOV AX,0000 | B80000 | IP=0044  AX=00FF | IP=0047  AX=0000 |
| 0047 | MOV ES,AX | 8EC0 | IP=0047  ES=11AE | IP=0049  ES=0000 |
| 0049 | PUSH DS | 1E | IP=0049  SP=0014  Stack  +0 0000  +2 119C  +4 0000  +6 0000 | IP=004A  SP=0012  Stack  +0 11AE  +2 0000  +4 119C  +6 0000 |
| 004A | POP ES | 07 | IP=004A  SP=0012  Stack  +0 11AE  +2 0000  +4 119C  +6 0000 | IP=004B  SP=0014  Stack  +0 0000  +2 119C  +4 0000  +6 0000 |
| 004B | MOV CX,ES:[BX-01] | 268B4FFF | IP=004B  CX=FD07 | IP=004F  CX=FFCE |
| 004F | XCHG AX,CX | 91 | IP=004F  AX=0000  CX=FFCE | IP=0050  AX=FFCE  CX=0000 |
| 0050 | MOV DI,0002 | BF0200 | IP=0050  DI=0002 | IP=0053  DI=0002 |
| 0053 | MOV ES:[BX+DI],AX | 268901 | IP=0053  ES:[BX+DI]=0027  AX=FFCE  ES=11AE  BX=0003  DI=0002 | IP=0056  ES:[BX+DI]=CEFF  AX=FFCE  ES=11AE  BX=0003  DI=0002 |
| 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  +0 0000  +2 119C  +4 0000  +6 0000 | IP=005C  SP=0012  Stack  +0 01F4  +2 0000  +4 119C  +6 0000 |
| 005C | PUSH [0002] | FF360200 | IP=005C  SP=0012  Stack  +0 01F4  +2 0000  +4 119C  +6 0000 | IP=0060  SP=0010  Stack  +0 FFCE  +2 01F4  +4 0000  +6 119C |
| 0060 | MOV BP,SP | 8BEC | IP=0060  BP=0014 | IP=0062  BP=0010 |
| 0062 | MOV DX,[BP+02] | 8B5602 | IP=0062  DX=0000 | IP=0065  DX=01F4 |
| 0065 | RET Far | CB | IP=0065  SP=0010  CS=11B1  Stack  +0 FFCE  +2 01F4  +4 0000  +6 119C | IP=FFCE  SP=0014  CS=01F4  Stack  +0 0000  +2 119C  +4 0000  +6 0000 |



**Вывод**

В результате выполнения лабораторной работы были изучены различные виды адресации (регистровая, прямая, косвенная, базированная, индексированная адресации и адресация с базированием и индексированием).

**Приложение**

Листинг lr2\_comp.lst (до исправления):

Microsoft (R) Macro Assembler Version 5.10 10/7/23 20:40:54

Page 1-1

; Учебная программа лабораторной работы №2 по

дисциплине "Организация ЭВМ и С";

;

= 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 СМЕЩЕНИ

Й

; Регистровая адресация

0009 B8 01F4 mov ax,n1

Microsoft (R) Macro Assembler Version 5.10 10/7/23 20:40:54

Page 1-2

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]

lr2\_comp.asm(54): 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]

lr2\_comp.asm(61): 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]

lr2\_comp.asm(65): warning A4031: Operand types must match

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

;;;

lr2\_comp.asm(66): 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]

lr2\_comp.asm(86): error A2046: Multiple base registers

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

;;;

lr2\_comp.asm(87): 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 CB ret

0074 Main ENDP

lr2\_comp.asm(94): error A2006: Phase error between passes

0074 CODE ENDS

END Main

Microsoft (R) Macro Assembler Version 5.10 10/7/23 20:40:54

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0018 PARA STACK

CODE . . . . . . . . . . . . . . 0074 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 = 0074

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\_comp

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

96 Source Lines

96 Total Lines

19 Symbols

47814 + 459446 Bytes symbol space free

2 Warning Errors

5 Severe Errors

Код программы lr2.asm (после исправления):

; Учебная программа лабораторной работы №2 по дисциплине "Организация ЭВМ и С";

;

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 39,38,37,36,32,33,34,35

vec2 DB 70,90,-70,-90,20,60,-20,-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]

; 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

Main ENDP

CODE ENDS

END Main

Листинг lr2.lst (после исправления):

#Microsoft (R) Macro Assembler Version 5.10 9/23/24 00:25:25

Page 1-1

; Учебная программа лабораторной работы №2 по

дисциплине "Организация ЭВМ и С";

;

= 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 27 26 25 24 20 21 vec1 DB 39,38,37,36,32,33,34,35

22 23

000E 46 5A BA A6 14 3C vec2 DB 70,90,-70,-90,20,60,-20,-60

EC C4

0016 02 01 FB FA 08 07 matr DB 2,1,-5,-6,8,7,-3,-4,4,3,-7,-8,6

,5,-1,-2

FD FC 04 03 F9 F8

06 05 FF FE

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

0009 B8 01F4 mov ax,n1

#Microsoft (R) Macro Assembler Version 5.10 9/23/24 00:25:25

Page 1-2

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]

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

lr2.ASM(61): 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]

lr2.ASM(65): warning A4031: Operand types must match

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

; ПРОВЕРКА АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ

; Переопределение сегмента

; ------ вариант 1

003C B8 ---- R mov ax, SEG vec2

003F 8E C0 mov es, ax

0041 26: 8B 07 mov ax, es:[bx]

0044 B8 0000 mov ax, 0

; ------ вариант 2

0047 8E C0 mov es, ax

0049 1E push ds

004A 07 pop es

004B 26: 8B 4F FF mov cx, es:[bx-1]

004F 91 xchg cx,ax

; ------ вариант 3

0050 BF 0002 mov di,ind

0053 26: 89 01 mov es:[bx+di],ax

; ------ вариант 4

0056 8B EC mov bp,sp

; mov ax,matr[bp+bx]

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

; Использование сегмента стека

0058 FF 36 0000 R push mem1

005C FF 36 0002 R push mem2

0060 8B EC mov bp,sp

0062 8B 56 02 mov dx,[bp]+2

0065 CB ret

0066 Main ENDP

0066 CODE ENDS

END Main

#Microsoft (R) Macro Assembler Version 5.10 9/23/24 00:25:25

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0018 PARA STACK

CODE . . . . . . . . . . . . . . 0066 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 = 0066

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

96 Source Lines

96 Total Lines

19 Symbols

47842 + 459418 Bytes symbol space free

2 Warning Errors

0 Severe Errors

Карта lr2.map (после исправления):

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 00045H 00026H DATA

00050H 000B5H 00066H CODE

Program entry point at 0005:0000