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. 3384 _____ Рудаков А.Л.

Преподаватель _____ Ковалев А.Д.

Санкт-Петербург

2024

Цель работы.

Изучить режимы адресации и формирования исполнительного адреса на языке Ассемблер.

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 приведенной для образца программе.

2. Протранслировать программу с созданием файла диагностических сообщений и объяснить обнаруженные ошибки (error) и предупреждения (warning). Закомментировать операторы с ошибками в тексте программы, а операторы с предупреждениями оставить без изменения. Объяснения ошибок и предупреждений должны быть приведены в отчете 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.

5. Результаты прогона программы под управлением отладчика должны быть подписаны преподавателем и представлены в отчете по лабораторной работе в виде, аналогичном указанному в лаб.работе №1.

Вариант 19

Основные теоретические положения.

1. Регистровая адресация:

Описание: Операнд - это значение, хранящееся в регистре процессора.

Пример: mov ax, bx - переносит содержимое регистра bx в регистр ax.

Особенности: Быстрый доступ, так как операнд находится в быстром регистре процессора.

2. Прямая адресация:

Описание: Операнд - это значение, хранящееся в ячейке памяти по фиксированному адресу.

Пример: mov al, [0x1234] - переносит байт из ячейки памяти по адресу 0x1234 в регистр al.

Особенности: Адрес ячейки памяти задан непосредственно в команде.

3. Косвенная адресация:

Описание: Операнд - это значение, хранящееся в ячейке памяти, адрес которой находится в другом регистре.

Пример: mov al, [bx] - переносит байт из ячейки памяти, адрес которой хранится в регистре bx, в регистр al.

Особенности: Позволяет гибко выбирать ячейку памяти для доступа, адрес которой можно менять в регистре.

4. Базированная адресация:

Описание: Операнд - это значение, хранящееся в ячейке памяти, адрес которой вычисляется как сумма смещения и значения базового регистра.

Пример: mov ax, [bx + 5] - переносит слово из ячейки памяти по адресу bx + 5 в регистр ax.

Особенности: Позволяет получить доступ к ячейке памяти, смещение которой от базового адреса задано в команде.

5. Индексированная адресация:

Описание: Операнд - это значение, хранящееся в ячейке памяти, адрес которой вычисляется как сумма смещения и значения индексного регистра.

Пример: mov al, [si + 2] - переносит байт из ячейки памяти по адресу si + 2 в регистр al.

Особенности: Позволяет получить доступ к элементам массива, используя индексный регистр для выбора нужного элемента.

6. Адресация с базированием и индексированием:

Описание: Операнд - это значение, хранящееся в ячейке памяти, адрес которой вычисляется как сумма смещения, значения базового регистра и значения индексного регистра.

Пример: mov al, [bx + si + 3] - переносит байт из ячейки памяти по адресу bx + si + 3 в регистр al.

Особенности: Позволяет получить доступ к сложным структурам данных, используя базовый и индексный регистры для вычисления адреса.

Выполнение работы.

Запуск DosBox, монтирование диска.

Подключение русификатора (переключение на кодировку CP866 для работы с буквами русского алфавита).

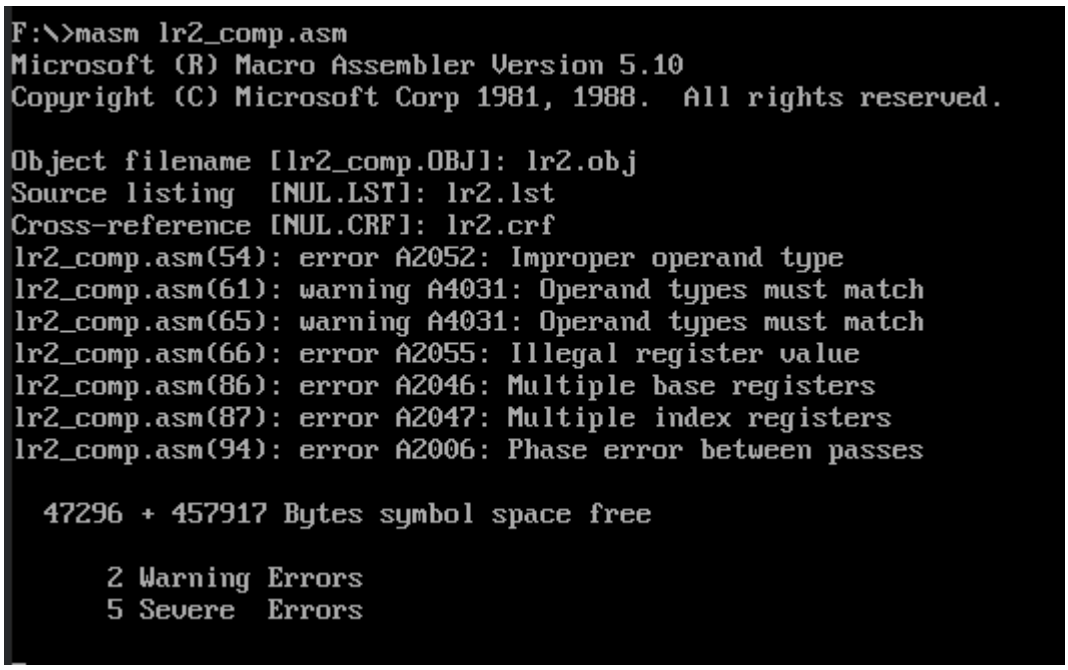
Переход к директории диска

Трансляция программы LR2_comp.asm:

Команда для запуска трансляции: masm lr2_comp.asm.

По команде порождаются объектный файл и файл листинга:

- lr2.obj – не сгенерирован
- lr2.lst.



Обнаружилось 5 ошибок и 2 предупреждения.

Ошибки:

- Ошибка A2052: "Improper operand type" (Строка 54) mov ax, es:[bx]

Проблема: В этой строке производиться попытка записать значение из памяти по адресу es:[bx] в регистр ax. ax - регистр слова (16 бит), который хранит 2 байта. Однако es:[bx] указывает на место в памяти, которое может содержать только 1 байт (потому что es указывает на сегмент vec2, где все данные типа DB - байты).

- Ошибка A2055: "Illegal register value" (Строка 66)

Проблема: производиться попытка использовать es в качестве базового регистра в адресации es:[bx+di], в то время, как es не может использоваться в качестве базового регистра. Базовые регистры - это bx, bp, di, si.

- Ошибка A2046: "Multiple base registers" (Строка 86)

Проблема: производиться попытка использовать два базовых регистра bp и bx одновременно в адресации matr[bp+bx], в то время, как можно использовать только один базовый регистр за раз.

Предупреждения:

- Предупреждение A4031: "Operand types must match" (Строки 61, 65)

Проблема: В этих строках производиться попытка записать значение, которое может быть одним байтом, в регистр cx (16 бит). Компилятор предупреждает о несовместимости типов.

После комментирования ошибок повторяем

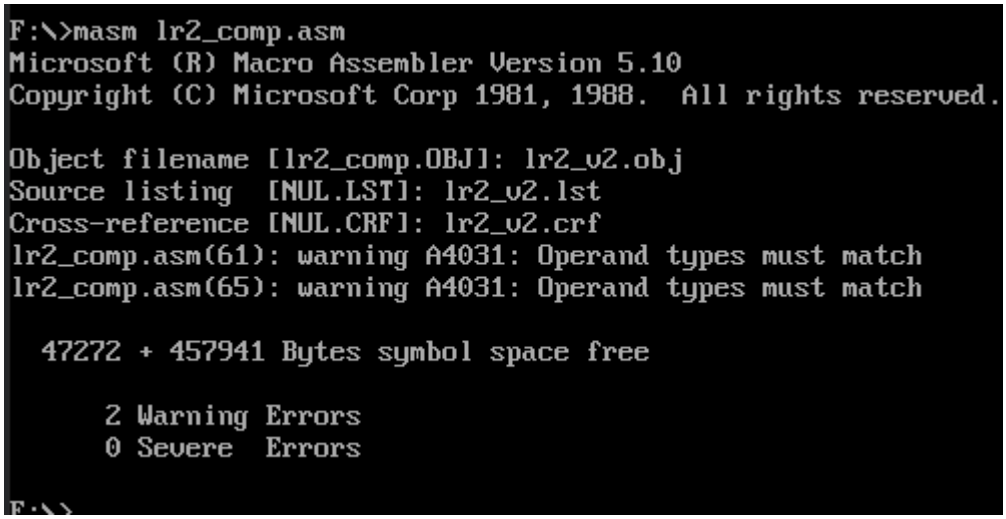
Трансляция программы LR2_comp.asm:

Команда для запуска трансляции: masm lr2_comp.asm.

По команде порождаются объектный файл и файл листинга:

- lr2_v2.obj

- lr2_v2.lst.



Компоновка загрузочного модуля:

Команда для запуска компоновки: link lr2_v2.obj.

По команде порождаются исполняемый файл и карта памяти:

- lr2_v2.exe

- lr2_v2.map

```
F:\>link lr2_v2.obj

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

Run File [LR2_V2.EXE]: lr2_v2.exe
List File [NUL.MAP]: lr2_v2.map
Libraries [.LIB]:

F:\>
```

Выполнение программы в отладчике:

Команда для запуска программы в режиме отладки: afd hello1.exe

```
AX 0000   SI 0000   CS 11B1   IP 0000   Stack +0 0000
BX 0000   DI 0000   DS 119C               +2 0000
CX 00B6   BP 0000   ES 119C   HS 119C      +4 0000
DX 0000   SP 0018   SS 11AC   FS 119C      +6 0000
```

Адрес команды	Код команды	16-ый код команды	Содержание регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	IP = 0000 SP = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	IP = 0001 SP = 0016 Stack +0 119C +2 0000 +4 0000 +6 0000
0001	SUB AX, AX	2BC0	FLAGS = 0200 AX = 0000 IP = 0001	FLAGS = 0244 AX = 0000 IP = 0003
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	AX = 0000 IP = 0004	AX = 11AE IP = 0007
0007	MOV DS, AX	8ED8	IP = 0007	IP = 0009

			DS = 119C	DS = 11AE
0009	MOV AX, 01F4	B8F401	AX = 11AE IP = 0009	AX = 01F4 IP = 000C
000C	MOV CX, AX	8DC8	IP = 000C CX = 00B6	IP = 000E CX = 01F4
000E	MOV BL, 24	B324	IP = 000E BX = 0000	IP = 0010 BX = 0024
0010	MOV DH, CE	B7CE	IP = 0010 BX = 0024	IP = 0012 BX = CE24
0012	MOV [0002], FFCE	C7060200CE FF	IP = 0012 DS:0002 = 00 DS:0003 = 00	IP = 0018 DS:0002 = CE DS:0003 = FF
0018	MOV BX, 0006	BB0600	IP = 0018 BX = CF24	IP = 001B BX = 0006
001B	MOV [0000], AX	A30000	IP = 001B DS:0000 = 00 DS:0001 = 00	IP = 001E DS:0000 = F4 DS:0001 = 01
001E	MOV AL, [BX]	8A07	IP = 001E AX = 01F4	IP = 0020 AX = 0120
0020	MOV AL, [BX+03]	8A4703	IP = 0020 AX = 0120	IP = 0023 AX = 0123
0023	MOV CX, [BX+03]	8B4F03	IP = 0023 CX = 01F4	IP = 0026 CX = 2723
0026	MOV DI, 0002	BF0200	IP = 0026 DI = 0000	IP = 0029 DI = 0002
0029	MOV AL, [DI+000E]	8A850E00	IP = 0029 AX = 0123	IP = 002D AX = 01CE
002D	MOV CX, [DI+000E]	8B8D0E00	IP = 002D CX = 2723	IP = 0031 CX = BACE
0031	MOV BX, 0003	BB0300	IP = 0031 BX = 0006	IP = 0034 BX = 0003
0034	MOV AL, [BX+DI+0016]	8A811600	IP = 0034 AX = 01CE	IP = 0038 AX = 0101

0038	MOV CX, [BX+DI+0016]	8B891600	IP = 0038 CX = BACE	IP = 003C CX = 0501
003C	MOV AX, 11AE	B8AE11	IP = 003C AX = 0101	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 ES = 0000 SP = 0012 Stack +0 11AE +2 0000 +4 119C +6 0000	IP = 004B ES = 11AE SP = 0014 Stack +0 0000 +2 119C +4 0000 +6 0000
004B	MOV CX, ES: [BX-01]	268B4FFF	IP = 004B CX = 0501	IP = 004F CX = FFCE
004F	XCHG AX, CX	91	IP = 004F AX = 0000 CX = FFCE	IP = 0050 AX = FFCE AX = 0000
0050	MOV DI, 0002	BF0200	IP = 0050 DI = 0002	IP = 0053 DI = 0002

0053	MOV ES: [BX+DI], AX	268901	IP = 0053 DS:0005 = 00 DS:0006 = 20	IP = 0056 DS:0005 = CE DS:0006 = FF
0056	MOV BP, SP	8BEC	IP = 0056 BP = 0000	IP = 0058 BP = 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]	FF3602000	IP = 005C SP = 0012 Stack +0 01F4 +2 0000 +4 119C +6 0000	IP = 0060 SP = 0100 Stack +0 FFCE +2 01F4 +4 0000 +6 119C
0060	MOV BP, SP	8BEC	IP = 0060 BP = 0014	IP = 0062 BX = 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 = 0066 SP = 0014 CS = 01F4 Stack +0 0000 +2 119C +4 0000 +6 0000

Сообщение отладчика после выполнения программы:

Программный код см. в приложении А.

Выводы.

В ходе лабораторной работы были исследованы разные методы адресации: регистровая, прямая, косвенная, базированная, индексированная, а также комбинированная адресация с использованием базирования и индексации.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr2_comp.asm

```
      ; 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  32,33,34,35,39,38,37,36
      vec2   DB  50,70,-50,-70,60,80,-60,-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
Main   ENDP
CODE   ENDS
END Main
```

Название файла: lr2.lst

```
Microsoft (R) Macro Assembler Version 5.10         9/29/24 19:21:03
                                Page   1-1


    1                   ; Учебная программа лабораторной работ
                              ы №2 по дисциплине "Организация ЭВМ и С
                              ";
    2                   ;
    3 = 0024                        EOL EQU '$'
    4 = 0002                        ind EQU  2
    5 = 01F4                        n1  EQU 500
    6 =-0032                        n2  EQU -50
    7
    8                   ; Стек программы
    9
   10 0000                    AStack  SEGMENT STACK
   11 0000 000C[                   DW 12 DUP(?)
   12   ????
   13             ]
```

```
   14
   15 0018                    AStack  ENDS
   16
   17                    ; Данные программы
   18
   19 0000                    DATA   SEGMENT
   20
   21                    ; Директивы описания данных
   22
   23 0000 0000               mem1   DW  0
   24 0002 0000               mem2   DW  0
   25 0004 0000               mem3   DW  0
   26 0006 20 21 22 23 27 26  vec1   DB  32,33,34,35,39,38,37,36
   27    25 24
   28 000E 32 46 CE BA 3C 50  vec2   DB  50,70,-50,-70,60,80,-60
                              ,-80
   29    C4 B0
   30 0016 04 03 F9 F8 02 01  matr   DB  4,3,-7,-8,2,1,5,-6,8,7,
                              -3,-4,-6,5,-1,-2
   31    05 FA 08 07 FD FC
   32    FA 05 FF FE
   33
   34 0026                    DATA   ENDS
   35
   36                    ; Код программы
   37
   38 0000                    CODE   SEGMENT
   39                        ASSUME CS:CODE, DS:DATA, SS:A
                              Stack
   40
   41                    ; Головная процедура
   42 0000                    Main   PROC FAR
   43 0000 1E                      push DS
   44 0001 2B C0                   sub  AX,AX
   45 0003 50                      push AX
   46 0004 B8 ---- R               mov  AX,DATA
   47 0007 8E D8                   mov  DS,AX
   48
   49                    ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ
Microsoft (R) Macro Assembler Version 5.10         9/29/24 19:21:03
                              Page   1-2


                              СМЕЩЕНИЙ
   50                    ; Регистровая адресация
   51 0009 B8 01F4                      mov ax,n1
   52 000C 8B C8                   mov cx,ax
   53 000E B3 24                   mov bl,EOL
   54 0010 B7 CE                   mov bh,n2
   55                    ; Прямая  адресация
   56 0012 C7 06 0002 R FFCE       mov mem2,n2
   57 0018 BB 0006 R               mov bx,OFFSET vec1
   58 001B A3 0000 R               mov mem1,ax
   59                    ; Косвенная адресация
   60 001E 8A 07                   mov al,[bx]
   61                         mov mem3,[bx]
lr2_comp.asm(54): error A2052: Improper operand type
   62                    ; Базированная адресация
   63 0020 8A 47 03                     mov al,[bx]+3
   64 0023 8B 4F 03                     mov cx,3[bx]
   65                    ; Индексированная адресация
   66 0026 BF 0002                      mov di,ind
   67 0029 8A 85 000E R            mov al,vec2[di]
```

```
   68 002D 8B 8D 000E R           mov cx,vec2[di]
lr2_comp.asm(61): warning A4031: Operand types must match
   69                   ; Адресация с базированием и индексиро
                              ванием
   70 0031 BB 0003                     mov bx,3
   71 0034 8A 81 0016 R           mov al,matr[bx][di]
   72 0038 8B 89 0016 R           mov cx,matr[bx][di]
lr2_comp.asm(65): warning A4031: Operand types must match
   73 003C 8B 85 0022 R           mov ax,matr[bx*4][di]
lr2_comp.asm(66): error A2055: Illegal register value
   74
   75                   ; ПРОВЕРКА АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТО
                              В
   76                   ; Переопределение сегмента
   77                   ; ------ вариант 1
   78 0040 B8 ---- R              mov ax, SEG vec2
   79 0043 8E C0                  mov es, ax
   80 0045 26: 8B 07              mov ax, es:[bx]
   81 0048 B8 0000                      mov ax, 0
   82                   ; ------ вариант 2
   83 004B 8E C0                  mov es, ax
   84 004D 1E                     push ds
   85 004E 07                     pop es
   86 004F 26: 8B 4F FF           mov cx, es:[bx-1]
   87 0053 91                     xchg cx,ax
   88                   ; ------ вариант 3
   89 0054 BF 0002                     mov di,ind
   90 0057 26: 89 01              mov es:[bx+di],ax
   91                   ; ------ вариант 4
   92 005A 8B EC                  mov bp,sp
   93 005C 3E: 8B 86 0016 R            mov ax,matr[bp+bx]
lr2_comp.asm(86): error A2046: Multiple base registers
   94 0061 3E: 8B 83 0016 R            mov ax,matr[bp+di+si]
lr2_comp.asm(87): error A2047: Multiple index registers
   95                   ; Использование сегмента стека
   96 0066 FF 36 0000 R           push mem1
   97 006A FF 36 0002 R           push mem2
   98 006E 8B EC                  mov  bp,sp
   99 0070 8B 56 02                    mov  dx,[bp]+2
   100 0073 CB                    ret
Microsoft (R) Macro Assembler Version 5.10         9/29/24 19:21:03
                               Page   1-3


   101 0074                   Main   ENDP
lr2_comp.asm(94): error A2006: Phase error between passes
   102 0074                   CODE   ENDS
   103                             END Main
Microsoft (R) Macro Assembler Version 5.10         9/29/24 19:21:03
                               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

  47296 + 457917 Bytes symbol space free

    2 Warning Errors
5 Severe Errors
```

Название файла: lr2_v2.map

```
 Start Stop  Length Name          Class
 00000H 00017H 00018H ASTACK
 00020H 00045H 00026H DATA
 00050H 000B5H 00066H CODE

Program entry point at 0005:0000
```

Название файла: lr2_v2.lst

```
Microsoft (R) Macro Assembler Version 5.10       9/29/24 23:30:19
                                                 Page 1-1


 1                       ; Учебная программа лабораторной работ
                         ы №2 по дисциплине "Организация ЭВМ и С
                         ";
 2                       ;
```

```
 3 = 0024                        EOL EQU '$'
 4 = 0002                        ind EQU  2
 5 = 01F4                        n1  EQU 500
 6 =-0032                        n2  EQU -50
 7
 8                          ; Стек программы
 9
10 0000                          AStack  SEGMENT STACK
11 0000 000C[                        DW 12 DUP(?)
12      ????
13                          ]
14
15 0018                          AStack  ENDS
16
17                          ; Данные программы
18
19 0000                          DATA   SEGMENT
20
21                          ; Директивы описания данных
22
23 0000 0000                     mem1   DW  0
24 0002 0000                     mem2   DW  0
25 0004 0000                     mem3   DW  0
26 0006 20 21 22 23 27 26 vec1   DB  32,33,34,35,39,38,37,36
27    25 24
28 000E 32 46 CE BA 3C 50        vec2   DB  50,70,-50,-70,60,80,-60
                                 ,-80
29    C4 B0
30 0016 04 03 F9 F8 02 01 matr   DB  4,3,-7,-8,2,1,5,-6,8,7,
                                 -3,-4,-6,5,-1,-2
31    05 FA 08 07 FD FC
32    FA 05 FF FE
33
34 0026                          DATA   ENDS
35
```

```
 36                             ; Код программы
 37
 38 0000                                CODE   SEGMENT
 39                                   ASSUME CS:CODE, DS:DATA, SS:A
                                        Stack
 40
 41                             ; Головная процедура
 42 0000                                Main   PROC FAR
 43 0000 1E                           push DS
 44 0001 2B C0                              sub  AX,AX
 45 0003 50                           push AX
 46 0004 B8 ---- R                    mov  AX,DATA
 47 0007 8E D8                              mov  DS,AX
 48
 49                             ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ
Microsoft (R) Macro Assembler Version 5.10        9/29/24 23:30:19
                    Page   1-2


                                        СМЕЩЕНИЙ
 50                             ; Регистровая адресация
 51 0009 B8 01F4                            mov ax,n1
 52 000C 8B C8                              mov cx,ax
 53 000E B3 24                              mov bl,EOL
 54 0010 B7 CE                              mov bh,n2
 55                             ; Прямая  адресация
 56 0012 C7 06 0002 R FFCE                  mov mem2,n2
 57 0018 BB 0006 R                    mov bx,OFFSET vec1
 58 001B A3 0000 R                    mov mem1,ax
 59                             ; Косвенная адресация
 60 001E 8A 07                              mov al,[bx]
 61                             ;     mov mem3,[bx]
 62                             ; Базированная адресация
 63 0020 8A 47 03                           mov al,[bx]+3
 64 0023 8B 4F 03                           mov cx,3[bx]
```

```
  65                            ; Индексированная адресация
  66 0026 BF 0002                       mov di,ind
  67 0029 8A 85 000E R                  mov al,vec2[di]
  68 002D 8B 8D 000E R                  mov cx,vec2[di]
lr2_comp.asm(61): warning A4031: Operand types must match
  69                            ; Адресация с базированием и индексиро
                                        ванием
  70 0031 BB 0003                       mov bx,3
  71 0034 8A 81 0016 R                  mov al,matr[bx][di]
  72 0038 8B 89 0016 R                  mov cx,matr[bx][di]
lr2_comp.asm(65): warning A4031: Operand types must match
  73                            ;     mov ax,matr[bx*4][di]
  74
  75                            ; ПРОВЕРКА АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТО
                                        В
  76                            ; Переопределение сегмента
  77                            ; ------ вариант 1
  78 003C B8 ---- R                 mov ax, SEG vec2
  79 003F 8E C0                         mov es, ax
  80 0041 26: 8B 07                 mov ax, es:[bx]
  81 0044 B8 0000                           mov ax, 0
  82                            ; ------ вариант 2
  83 0047 8E C0                         mov es, ax
  84 0049 1E                        push ds
  85 004A 07                        pop es
  86 004B 26: 8B 4F FF                  mov cx, es:[bx-1]
  87 004F 91                        xchg cx,ax
  88                            ; ------ вариант 3
  89 0050 BF 0002                       mov di,ind
  90 0053 26: 89 01                 mov es:[bx+di],ax
  91                            ; ------ вариант 4
  92 0056 8B EC                         mov bp,sp
  93                            ;     mov ax,matr[bp+bx]
  94                            ;     mov ax,matr[bp+di+si]
  95                            ; Использование сегмента стека
```

```
  96 0058 FF 36 0000 R                  push mem1
  97 005C FF 36 0002 R                  push mem2
  98 0060 8B EC                         mov  bp,sp
  99 0062 8B 56 02                      mov  dx,[bp]+2
 100 0065 CB                            ret
Microsoft (R) Macro Assembler Version 5.10        9/29/24 23:30:19
                                                  Page   1-3


 101 0066                               Main   ENDP
 102 0066                               CODE   ENDS
 103                                 END Main
Microsoft (R) Macro Assembler Version 5.10        9/29/24 23:30:19
                                                  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_comp
@VERSION . . . . . .  TEXT 510


  96 Source Lines
  96 Total  Lines
  19 Symbols

 47272 + 457941 Bytes symbol space free

  2 Warning Errors
  0 Severe Errors
```