

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
КАФЕДРА МО ЭВМ

ОТЧЕТ

по лабораторной работе №2
по дисциплине «ОргЭВМиС»

Тема: Изучение режимов адресации и формирования исполнительного
адреса.

Студент гр. 0382

Афанасьев Н. С.

Преподаватели

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить режимы адресации и формирование исполнительного адреса на языке Ассемблера.

Задание.

1. Получить у преподавателя вариант набора значений исходных данных (массивов) `vec1`, `vec2` и `matr` из файла `lr2.dat`, приведенного в каталоге Задания и занести свои данные вместо значений, указанных в приведенной ниже программе.

2. Протранслировать программу с созданием файла диагностических сообщений; объяснить обнаруженные ошибки и закомментировать соответствующие операторы в тексте программы.

3. Снова протранслировать программу и скомпоновать загрузочный модуль.

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

Выполнение работы.

Значения `vec1`, `vec2` и `matr` из варианта были подставлены в программу.

Программа протранслирована с созданием файла диагностических сообщений. Были обнаружены ошибки и предупреждения:

- `mov mem3,[bx]` - error A2052: Improper operand type.

Перемещать данные можно только между регистром и памятью или между регистрами.

- `mov cx,vec2[di]` - warning A4031: Operand types must match.
- `mov cx,matr[bx][di]` - warning A4031: Operand types must match.

Левый операнд (`cx`) имеет размер 2 байта, а правый операнд – 1 байт.

- `mov ax,matr[bx*4][di]` - error A2055: Illegal register value.

Масштабирование доступно только для расширенных регистров (32 бит).

- `mov ax,matr[bp+bx]` - error A2046: Multiple base registers.

Нельзя использовать более 1-го базового регистра.

- **mov ax,matr[bp+di+si]** - error A2047: Multiple index registers.

Нельзя использовать более 1-го индексного регистра.

Ошибки были закомментированы, и программа заново протранслирована и затем скомпилирована.

Программа была выполнена в пошаговом режиме в отладчике – результаты представлены в таблице 1.

Таблица 1 – Результаты пошагового выполнения программы hello2.exe

Начальные значения: (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	(SP) = 0018 Stack: +0 0000	(SP) = 0016 Stack: +0 19F5
0001	sub AX,AX	2BC0	(AX) = 0000	(AX) = 0000
0003	push AX	50	(SP) = 0016 (AX) = 0000 Stack: +0 19F5	(SP) = 0014 (AX) = 0000 Stack: +0 0000 +2 19F5
0004	mov AX,DATA	B8071A	(AX) = 0000	(AX) = 1A07
0007	mov DS,AX	8ED8	(DS) = 19F5	(DS) = 1A07
0009	mov ax,n1	B8F401	(AX) = 1A07	(AX) = 01F4
000C	mov cx,ax	8BC8	(CX) = 00B0	(CX) = 01F4
000E	mov bl,EOL	B324	(BX) = 0000	(BX) = 0024
0010	mov bh,n2	B7CE	(BX) = 0024	(BX) = CE24
0012	mov mem2,n2	C7060200CE FF	DS[2] = 00 DS[3] = 00	DS[2] = CE DS[3] = FF
0018	mov bx,OFFSET vec1	BB0600	(BX) = CE24	(BX) = 0006

001B	mov mem1,ax	A30000	DS[0] = 00 DS[1] = 00	DS[0] = F4 DS[1] = 01
001E	mov al,[bx]	8A07	(AX) = 01F4	(AX) = 0105
0020	mov al,[bx]+3	8A4703	(AX) = 0105	(AX) = 0108
0023	mov cx,3[bx]	8B4F03	(CX) = 01F4	(CX) = 0C08
0026	mov di,ind	BF0200	(DI) = 0000	(DI) = 0002
0029	mov al,vec2[di]	8A850E00	(AX) = 0108	(AX) = 0114
002D	mov bx,3	BB0300	(BX) = 0006	(BX) = 0003
0030	mov al,matr[bx][di]	8A811600	(AX) = 0114	(AX) = 0103
0034	mov ax, SEG vec2	B8071A	(AX) = 0103	(AX) = 1A07
0037	mov es, ax	8EC0	(ES) = 19F5	(ES) = 1A07
0039	mov ax, es:[bx]	268B07	(AX) = 1A07	(AX) = 00FF
003C	mov ax, 0	B80000	(AX) = 00FF	(AX) = 0000
003F	mov es, ax	8EC0	(ES) = 1A07	(ES) = 0000
0041	push ds	1E	(DS) = 1A07 (SP) = 0014 Stack: +0 0000 +2 19F5	(DS) = 1A07 (SP) = 0012 Stack: +0 1A07 +2 0000 +4 19F5
0042	pop es	07	(ES) = 0000 (SP) = 0012 Stack: +0 1A07 +2 0000 +4 19F5	(ES) = 1A07 (SP) = 0014 Stack: +0 0000 +2 19F5
0043	mov cx, es:[bx- 1]	268B4FFF	(CX) = 0C08	(CX) = FFCE
0047	xchg cx,ax	91	(AX) = 0000 (CX) = FFCE	(AX) = FFCE (CX) = 0000
0048	mov di,ind	BF0200	(DI) = 0002	(DI) = 0002

004B	mov es:[bx+di],ax	268901	DS[5] = 00 DS[6] = 05	DS[5] = CE DS[6] = FF
004E	mov bp,sp	8BEC	(BP) = 0000	(BP) = 0014
0050	push mem1	FF360000	(SP) = 0014 Stack: +0 0000 +2 19F5	(SP) = 0012 Stack: +0 01F4 +2 0000 +4 19F5
0054	push mem2	FF360200	(SP) = 0012 Stack: +0 01F4 +2 0000 +4 19F5	(SP) = 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	mov bp,sp	8BEC	(BP) = 0014	(BP) = 0010
005A	mov dx,[bp]+2	8B5602	(DX) = 0000	(DX) = 01F4
005D	ret 2	CA0200	(CS) = 1A0A (SP) = 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5	(CS) = 01F4 (SP) = 0016 Stack: +0 19F5

Выводы.

Были изучены режимы адресации и формирование исполнительного адреса. Также был изучен процесс поиска ошибок.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЕ КОДЫ ПРОГРАММ

Название файла: lr2.asm

```
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50

AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

DATA SEGMENT
    mem1 DW 0
    mem2 DW 0
    mem3 DW 0
    vec1 DB 5,6,7,8,12,11,10,9
    vec2 DB -20,-30,20,30,-40,-50,40,50
    matr DB -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4,8,7,6,5
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
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]
    ; Индексная 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 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]
;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

```

ПРИЛОЖЕНИЕ В

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr2-err.lst

Microsoft (R) Macro Assembler Version 5.10

9/27/21 16:43:58

```
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50

0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
      ????
    ]

0018          AStack ENDS

0000          DATA SEGMENT
0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 05 06 07 08 0C 0B      vec1 DB 5,6,7,8,12,11,10,9
      0A 09
000E EC E2 14 1E D8 CE      vec2 DB -20,-30,20,30,-40,-50,40,50
      28 32
0016 FB FA F9 F8 04 03      matr DB -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4
,8,7,6,5
      02 01 FF FE FD FC
      08 07 06 05
0026          DATA ENDS

0000          CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack
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
0009 B8 01F4          mov ax,n1
000C 8B C8          mov cx,ax
000E B3 24          mov bl,EOL
0010 B7 CE          mov bh,n2
0012 C7 06 0002 R FFCE      mov mem2,n2
0018 BB 0006 R      mov bx,OFFSET vec1
001B A3 0000 R      mov mem1,ax
001E 8A 07          mov al,[bx]
mov mem3,[bx]
lr2.ASM(35): error A2052: Improper operand type
0020 8A 47 03          mov al,[bx]+3
0023 8B 4F 03          mov cx,3[bx]
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(40): warning A4031: Operand types must match
```



```

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(43): warning A4031: Operand types must match
003C 8B 85 0022 R mov ax,matr[bx*4][di]
lr2.ASM(44): error A2055: Illegal register value
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
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
0054 BF 0002 mov di,ind
0057 26: 89 01 mov es:[bx+di],ax
005A 8B EC mov bp,sp
005C 3E: 8B 86 0016 R mov ax,matr[bp+bx]
lr2.ASM(57): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R mov ax,matr[bp+di+si]
lr2.ASM(58): error A2047: Multiple index registers
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
lr2.ASM(64): error A2006: Phase error between passes
0076 CODE ENDS
END Main

```

Microsoft (R) Macro Assembler Version 5.10

9/27/21 16:43:58

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 lr2
@VERSION . . . . . TEXT 510

```

```

66 Source Lines
66 Total Lines
19 Symbols

```

47834 + 461473 Bytes symbol space free

```

2 Warning Errors
5 Severe Errors

```

Название файла: lr2.lst

Microsoft (R) Macro Assembler Version 5.10

9/27/21 16:52:04

```

= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50

0000            AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
    ???
]

0018            AStack ENDS

0000            DATA SEGMENT
0000 0000      mem1 DW 0
0002 0000      mem2 DW 0
0004 0000      mem3 DW 0
0006 05 06 07 08 0C 0B      vec1 DB 5,6,7,8,12,11,10,9
    0A 09
000E EC E2 14 1E D8 CE      vec2 DB -20,-30,20,30,-40,-50,40,50
    28 32
0016 FB FA F9 F8 04 03      matr DB -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4
,8,7,6,5
    02 01 FF FE FD FC
    08 07 06 05
0026            DATA ENDS

0000            CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack
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
0009 B8 01F4     mov ax,n1
000C 8B C8      mov cx,ax

```

000E	B3 24	mov bl,EOL
0010	B7 CE	mov bh,n2
0012	C7 06 0002 R FFCE	mov mem2,n2
0018	BB 0006 R	mov bx,OFFSET vec1
001B	A3 0000 R	mov mem1,ax
001E	8A 07	mov al,[bx]
	;mov mem3,[bx]	
0020	8A 47 03	mov al,[bx]+3
0023	8B 4F 03	mov cx,3[bx]
0026	BF 0002	mov di,ind
0029	8A 85 000E R	mov al,vec2[di]
	;mov cx,vec2[di]	
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]	
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
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
0048	BF 0002	mov di,ind
004B	26: 89 01	mov es:[bx+di],ax
004E	8B EC	mov bp,sp
	;mov ax,matr[bp+bx]	
	;mov ax,matr[bp+di+si]	
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	CA 0002	ret 2
0060		Main ENDP
0060		CODE ENDS
	END Main	

Microsoft (R) Macro Assembler Version 5.10

9/27/21 16:52:04
Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0060	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 = 0060
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	1r2		
@VERSION	TEXT	510		

66 Source Lines
 66 Total Lines
 19 Symbols

47842 + 461465 Bytes symbol space free

0 Warning Errors
 0 Severe Errors