

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»
Тема: Изучение режимов адресации и формирования
исполнительного адреса.

Студент(ка) гр. 1381

Биктагирова Д.С.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

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.

Выполнение 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

```
Z:\>mount c d:/tools
Drive C is mounted as local directory d:/tools\

Z:\>c:

C:\>masm lr2_comp.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lr2_comp.OBJ]:
Source listing [NUL.LST]: lr2_comp
Cross-reference [NUL.CRF]:
lr2_comp.asm(41): error A2052: Improper operand type
lr2_comp.asm(48): warning A4031: Operand types must match
lr2_comp.asm(52): warning A4031: Operand types must match
lr2_comp.asm(53): error A2055: Illegal register value
lr2_comp.asm(67): error A2006: Phase error between passes

47800 + 461507 Bytes symbol space free

      2 Warning Errors
      3 Severe Errors
```

1)

```
54      L          mov mem3,[bx]
55      □ lr2_comp.asm(41): error A2052: Improper operand type
```

Нельзя читать из памяти и писать в память одной командой.

2)

```
66      002D  8B 8D 000E R      mov cx,vec2[di]
67      □ lr2_comp.asm(48): warning A4031: Operand types must match
```

Размер элементов массива `vec2` - 1 байт, а регистра `cx` - 2 байта.

3)

```
72      0038  8B 89 0016 R      mov cx,matr[bx][di]
73      □ lr2_comp.asm(52): warning A4031: Operand types must match
```

Размер элементов матрицы `matr` - 1 байт, а регистра `cx` - 2 байта.

4)

```
74      003C 8B 85 0022 R      mov ax,matr[bx*4][di]
75      lr2_comp.asm(53): error A2055: Illegal register value
```

В режиме реальных адресов нельзя использовать адресацию с масштабированием (нельзя умножать 16-битные регистры).

5)

```
90      0058 8B 85 02      mov dx,[bp]
91      0058 CA 0002      ret 2
92      005B      Main ENDP
93      lr2_comp.asm(67): error A2006: Phase error between passes
94      005B      CODE ENDS
```

Данная ошибка свидетельствует о том, что в процедуре Main содержатся ошибки.

3. Повторная трансляция исправленной программы и линковка.

```
C:\>masm lr2_comp.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lr2_comp.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49950 + 461407 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link lr2_comp.obj

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

Run File [LR2_COMP.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>
```

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

Начальные значения: CS = 1A0A, DS = 19F5, ES = 19F5, SS = 1A05, SP = 0018, IP = 0000.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	Stack: +0 0000 +2 0000 +4 0000 +6 0000 DS =19F5 SP = 0018 IP = 0000	Stack: +0 19F5 +2 0000 +4 0000 +6 0000 DS =19F5 SP = 0016 IP = 0001
0001	SUB AX, AX	2BC0	AX = 0000 IP = 0001	AX = 0000 IP = 0003
0003	PUSH AX	50	Stack: +0 19F5 +2 0000 +4 0000 +6 0000 AX = 0000 SP = 0016 IP = 0003	Stack: +0 0000 +2 19F5 +4 0000 +6 0000 AX = 0000 SP = 0014 IP = 0004
0004	B8071A	MOV AX, 1A07	AX = 0000 IP = 0004	AX = 1A07 IP = 0007
0007	8ED8	MOV DS, AX	AX = 1A07 DS =19F5 IP = 0007	AX = 1A07 DS =1A07 IP = 0009
0009	B8F401	MOV AX, 01F4	AX = 1A07 IP = 0009	AX = 01F4 IP = 000C
000C	8BC8	MOV CX, AX	CX = 009F AX = 01F4 IP = 000C	CX = 01F4 AX = 01F4 IP = 000E
000E	B324	MOV BL, 24	BX = 0000 IP = 000E	BX = 0024 IP = 0010
0010	B7CE	MOV BH, CE	BX = 0024 IP = 0010	BX = CE24 IP = 0012
0012	C7060200CEFF	MOV [0002], FFCE	IP = 0012	IP = 0018

0018	BB0600	MOV BX, 0006	BX = CE24 IP = 0018	BX = 0006 IP = 001B
001B	A30000	MOV [0000], AX	AX = 01F4 IP = 001B	AX = 01F4 IP = 001E
001E	8A07	MOV AL, [BX]	AX = 01F4 BX = 0006 IP = 001E	AX = 0101 BX = 0006 IP = 0020
0020	8A4703	MOV AL, [BX + 03]	AX = 0101 BX = 0006 IP = 0020	AX = 0104 BX = 0006 IP = 0023
0023	8B4F03	MOV CX, [BX + 03]	CX = 01F4 BX = 0006 IP = 0023	CX = 0804 BX = 0006 IP = 0026
0026	BF0200	MOV DI, 0002	DI = 0000 IP = 0026	DI = 0002 IP = 0029
0029	8A850E00	MOV AL, [000E + DI]	AX = 0104 DI = 0002 IP = 0029	AX = 010A DI = 0002 IP = 002D
002D	BB0300	MOV BX, 0003	BX = 0006 IP = 002D	BX = 0003 IP = 0030
0030	8A811600	MOV AL, [0016 + BX + DI]	AX = 010A BX = 0003 DI = 0002 IP = 0030	AX = 01FD BX = 0003 DI = 0002 IP = 0034
0034	B8071A	MOV AX, 1A07	AX = 01FD IP = 0034	AX = 1A07 IP = 0037
0037	8EC0	MOV ES, AX	AX = 1A07 ES = 19F5 IP = 0037	AX = 1A07 ES = 1A07 IP = 0039
0039	268B07	MOV AX, ES:[BX]	AX = 1A07 ES = 1A07 BX = 0003 IP = 0039	AX = 00FF ES = 1A07 BX = 0003 IP = 003C
003C	B80000	MOV AX, 0000	AX = 00FF IP = 003C	AX = 0000 IP = 003F
003F	FF360000	PUSH [0000]	Stack: +0 0000 +2 19F5	Stack: +0 01F4 +2 0000

			+4 0000 +6 0000 SP = 0014 IP = 003F	+4 19F5 +6 0000 SP = 0012 IP = 0043
0043	FF360200	PUSH [0002]	Stack: +0 01F4 +2 0000 +4 19F5 +6 0000 SP = 0012 IP = 0043	Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 SP = 0010 IP = 0047
0047	8BEC	MOV BP, SP	BP = 0000 SP = 0010 IP = 0047	BP = 0010 SP = 0010 IP = 0049
0049	8B5602	MOV DX, [BP + 02]	DX = 0000 BP = 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 SP = 0010 IP = 0049	DX = 01F4 BP = 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 SP = 0010 IP = 004C
004C	CA0200	RET Far 0002	Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 SP = 0010 IP = 004C CS = 1A0A	Stack: +0 19F5 +2 0000 +4 0000 +6 0000 SP = 0016 IP = FFCE CS = 01F4

Выводы.

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

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr2_comp.asm

```
; 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]
; 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
; Использование сегмента стека
push mem1
push mem2
mov bp, sp
mov dx, [bp]+2
ret 2
Main ENDP
CODE ENDS
END

```