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

ОТЧЁТ

по лабораторной работе №2
по дисциплине «Организация ЭВМ и систем»
Тема: Изучение режимов адресации основной памяти.

Студент гр. 0381 _____ Ковалев П. А.

Преподаватель _____ Ефремов М. А.

Санкт-Петербург
2022

Цель работы

Изучить режимы адресации основной памяти.

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.

Выполнение работы

1. Был скопирован файл `LR2_comp.ASM` в каталог с компилятором ASM
2. Был занесен набор значений данных в соответствии с вариантом 9.
3. С помощью эмулятора `dosbox` и компилятора `MASM` программа была оттранслирована. Были получены следующие ошибки:
   - `error A2052: Improper operand type`

     Ошибка в строке: `mov mem3,[bx]`. Нельзя копировать данные из памяти в память, только из памяти в регистр и наоборот.
   - `warning A4031: Operand types must match`

     Ошибка в строке: `mov cx,vec2[di]`. Размерности операнд инструкции не совпадают: регистр cx – 2 байта, vec2 – 1 байт.
   - `error A2055: Illegal register value`

     Ошибка в строке: `mov ax,matr[bx*4][di]`. Умножение адреса на число не поддерживается, только смещение(+).
   - `error A2046: Multiple base registers`

     Ошибка в строке: `mov ax,matr[bp+bx]`. В “[ ]” указано два регистра, второй регистр должен быть индексным(di, si).
   - `error A2046: Multiple index registers`

     Ошибка в строке: `mov ax,matr[bp+di+si]`. В “[ ]” указано 2 индексных регистра.
4. После исправления ошибок и успешной линковки программа была выполнена пошагово с помощью утилиты `AFD.COM`.

   Результаты выполнения представлены в Таблице 1

Вывод

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

Приложение А

Результаты пошагового выполнения программы

Таблица 1: Результаты выполнения программы в пошаговом режиме

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров до выполнения команды	Содержимое регистров после выполнения команды	Содержимое стека до выполнения команды	Содержимое стека после выполнения команды
0000	PUSH DS	1E	SP=0018 IP=0000	SP=0016 IP=0001	+0 0000 +2 0000 +4 0000 +6 0000	+0 119C +2 0000 +4 0000 +6 0000
0001	SUB AX,AX	2BC0	AX=0000 IP=0001	AX=0000 IP=0003		
0003	PUSH AX	50	SP=0016 IP=0003	SP=0014 IP=0004	+0 119C +2 0000 +4 0000 +6 0000	+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	AX=11AE DS=119C IP=0007	AX=11AE DS=11AE IP=0009		
0009	MOV AX,01F4	B8F401	AX=11AE IP=0009	AX=01F4 IP=000C		
000C	MOV CX,AX	8BC8	CX=00D2 IP=000C	CX=01F4 IP=000E		
000E	MOV BL,24	B324	BX=0000 IP=000E	BX=0024 IP=0010		
0010	MOV BH,CE	B7CE	BX=0024 IP=0010	BX=CE24 IP=0012		
0012	MOV [0002], FFCE	C70602 00CEFF	IP=0012	IP=0018		
0018	MOV BX,0006	BB0600	BX=CE24 IP=0018	BX=0006 IP=001B		
001B	MOV [0000],AX	A30000	IP=001B	IP=001E		
001E	MOV AL,[BX]	8A07	AX=01F4 IP=001E	AX=011F IP=0020		
0020	MOV AX,[BX]	8B07	AX=011F IP=0020	AX=201F IP=0022		

Таблица 2: Продолжение таблицы 1

0022	MOV [0004],AX	A30400	IP=0022	IP=0025		
0025	MOV AL,[BX+03]	8A4703	AX=201F IP=0025	AX=2022 IP=0028		
0028	MOV CX,[BX+03]	8B4F03	CX=01F4 IP=0028	CX=2622 IP=002B		
002B	MOV DI,0002	BF0200	DI=0000 IP=002B	DI=0002 IP=002E		
002E	MOV AL,[DI+000E]	8A85 0E00	AX=2022 IP=002E	AX=20CE IP=0032		
0032	MOV CL,[DI+000E]	8A8D 0E00	CX=2622 IP=0032	CX=26CE IP=0036		
0036	MOV BX,0003	BB0300	BX=0006 IP=0036	BX=0003 IP=0039		
0039	MOV AL,[BX+DI+0016]	8A81 1600	AX=20CE IP=0039	AX=20FF IP=003D		
003D	MOV CL,[BX+DI+0016]	8A89 1600	CX=26CE IP=003D	CX=26FF IP=0041		
0041	SHL BX,1	D1E3	BX=0003 IP=0041	BX=0006 IP=0043		
0043	SHL BX,1	D1E3	BX=0006 IP=0043	BX=0012 IP=0045		
0045	MOV AL,[BX+DI+0016]	8A81 1600	AX=20FF IP=0045	AX=2001 IP=0049		
0049	MOV AX,11AE	B8AE11	AX=2001 IP=0049	AX=11AE IP=004C		
004C	MOV ES,AX	8EC0	ES=119C IP=004C	ES=11AE IP=004E		
004E	MOV AX,ES:[BX]	268B07	AX=11AE IP=004E	AX=2324 IP=0051		
0051	MOV AX,0000	B80000	AX=2324 IP=0051	AX=0000 IP=0054		
0054	MOV ES,AX	8EC0	ES=11AE IP=0054	ES=0000 IP=0056		
0056	PUSH DS	1E	SP=0014 IP=0056	SP=0012 IP=0057	+0 0000 +2 119C +4 0000 +6 0000	+0 11AE +2 0000 +4 119C +6 0000

Таблица 3: Продолжение таблицы 1

0057	POP ES	07	SP=0012 IP=0057 ES=0000	SP=0014 IP=0058 ES=11AE	+0 11AE +2 0000 +4 119C +6 0000	+0 0000 +2 119C +4 0000 +6 0000
0058	MOV CX,ES:[BX-01]	268B 4FFF	CX=26FF IP=0058	CX=2425 IP=005C		
005C	XCHG AX,CX	91	AX=0000 CX=2425 IP=005C	AX=2425 CX=0000 IP=005D		
005D	MOV DI,0002	BF0200	DI=0002 IP=005D	DI=0002 IP=0060		
0060	MOV ES:[BX+DI],AX	268901	ES=11AE AX=2425 IP=0060	ES=11AE AX=2425 IP=0063		
0063	MOV BP,SP	8BEC	BP=0000 IP=0063	BP=0014 IP=0065		
0065	ADD BX,BP	03DD	BX=0006 IP=0065	BP=0020 IP=0067		
0067	MOV AL,[BX+0016]	8A87 1600	AX=2425 IP=0067	AX=2411 IP=006B		
006B	ADD DI,SI	03FE	DI=0002 IP=006B	DI=0002 IP=006D		
006D	ADD BP,DI	03EF	BP=0014 IP=006D	BP=0016 IP=006F		
006F	MOV AL,DS:[BP+0016]	3E8A 861600	AX=2411 IP=006F	AX=2400 IP=0074		
0074	PUSH [0000]	FF36 0000	SP=0014 IP=0074	SP=0012 IP=0078	+0 0000 +2 119C +4 0000 +6 0000	+0 01F4 +2 0000 +4 119C +6 0000
0078	PUSH [0002]	FF36 0200	SP=0012 IP=0078	SP=0010 IP=007C	+0 01F4 +2 0000 +4 119C +6 0000	+0 FFCE +2 01F4 +4 0000 +6 119C
007C	MOV BP,SP	8BEC	BP=0016 IP=007C	BP=0010 IP=007E		
007E	MOV DX,[BP+02]	8B5602	DX=20FF IP=007E	DX=01F4 IP=0081		
0081	RET Far	CB	IP=0081	IP=FFCE		

Приложение Б

Исходный код программ

Название файла: LR2_comp.ASM

```
; 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
Main      ENDP
CODE      ENDS
          END Main
```

Название файла: LR2_comp_after.ASM

```
; 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    31,32,33,34,38,37,36,35
vec2      DB    50,60,-50,-60,70,80,-70,-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  ax,[bx]
          mov  mem3,ax
;  Базированная адресация
          mov  al,[bx]+3
          mov  cx,3[bx]
;  Индексированная адресация
          mov  di,ind
          mov  al,vec2[di]
          mov  cl,vec2[di]
;  Адресация с базированием и индексированием
          mov  bx,3
          mov  al,matr[bx][di]
          mov  cl,matr[bx][di]
          shl  bx,1
          shl  bx,1
          mov  al,matr[bx][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
          add  bx,bp
```

```
          mov  al,matr[bx]
          add  di,si
          add  bp,di
          mov  al,matr[bp]
;  Использование сегмента стека
          push  mem1
          push  mem2
          mov   bp,sp
          mov   dx,[bp]+2
          ret
Main      ENDP
CODE      ENDS
          END Main
```