Министерство образования Российской Федерации

**Санкт-Петербургский государственный**

**электротехнический университет "ЛЭТИ"**

**кафедра МОЭВМ**

факультет КТИ

Лабораторная работа №2

**" Изучение режимов адресации основной памяти "**

Вариант 10

Выполнил: Сорокин Е.В.

Гр 0303

Проверил:

Санкт-Петербург

2012 год

1. **Теоретическая справка**

# Режимы адресации памяти в процессоре Intel 8086

Большинство команд процессора выполняются с аргументами, которые принято называть операндами. Операнды в программе могут задаваться следующим образом:

* в регистрах общего назначения;
* непосредственно в коде команды;
* в ячейках памяти, задаваемых в команде прямо или косвенно;
* в портах ввода-вывода.

Для указания места расположения операнда используются 7 режимов адресации.

1. Регистровая адресация

Операнды могут располагаться в любых регистрах общего назначения и сегментных регистрах. В этом случае в операторе программы (на языке ассемблера) указывается название соответствующего регистра.

2. Непосредственная адресация

Некоторые команды (пересылки, все арифметические команды, кроме деления) по­зволяют указывать один из операндов непосредственно в операторе про­граммы.

3. Прямая адресация

Если известен адрес операнда, располагающегося в памяти, можно ис­пользовать этот адрес. В реальных программах обычно для задания статических переменных используют директивы определения данных, которые позволяют ссылаться на статические пере­менные не по адресу, а по имени.

Если селектор сегмента данных находится в DS, имя сегментного регистра при прямой адресации можно не указывать, так как DS используется по умолчанию. Прямая адресация иногда называется ад­ресацией по смещению.

4. Косвенная адресация

Адрес операнда в памяти также можно не указывать непосредственно, а хранить в любом регистре. До 80386 для этого можно было использовать только ВХ, SI, DI и ВР, но потом эти ограничения были сняты и адрес операнда разрешили считывать также и из ЕАХ, ЕВХ, ЕСХ, EDX, ESI, EDI, ЕВР и ESP (но не из AX, CX, DX или SP напрямую – надо исполь­зовать ЕАХ, ЕСХ, EDX, ESP соответственно или предварительно скопи­ровать смещение в ВХ, S1, DI или ВР). Как и в случае прямой адресации, DS используется по умолчанию, но не во всех случаях: если смещение берут из регистров ESP, ЕВР или ВР, то в качестве сегментного регистра используется SS. В реальном режиме можно свободно пользоваться всеми 32-битными регистрами, надо толь­ко следить, чтобы их содержимое не превышало границ 16-битного слова.

5. Адресация по базе

Та­кая форма адресации используется в тех случаях, когда в регистре находится адрес начала структуры данных, а доступ надо осуществить к какому-нибудь элементу этой структуры. Другое важное применение адресации по базе со сдвигом – доступ из подпрограммы к параметрам, переданным в стеке, используя регистр ВР (ЕВР) в качестве базы и но­мер параметра в качестве смещения.

До 80386 в качестве базового регистра можно было использовать только ВХ, ВР, SI или DI и сдвиг мог быть только байтом или словом (со знаком). Начиная с 80386 и старше, процессоры Intel позволяют допол­нительно использовать ЕАХ, ЕВХ, ЕСХ, EDX, ЕВР, ESP, ESI и EDI, так же как и для обычной косвенной адресации. С помощью этого метода можно организовывать доступ к одномерным массивам байт: смещение соответствует адресу начала массива, а число в регистре – индексу эле­мента массива, который надо использовать. Очевидно, что если массив состоит не из байт, а из слов, придется умножать базовый регистр на два, а если из двойных слов – на четыре. Для этого предусмотрен следующий специальный метод адресации.

6. Косвенная адресация с масштабированием

Этот метод адресации полностью идентичен предыдущему, за исключе­нием того, что с его помощью можно прочитать элемент массива слов, двойных слов или учетверенных слов, просто поместив номер элемента в регистр

mov ax, [esi\*2]+2

Множитель, который может быть равен 1, 2, 4 или 8, соответствует размеру элемента массива – байту, слову, двойному слову, учетверенно­му слову соответственно. Из регистров в этом варианте адресации мож­но использовать только ЕАХ, ЕВХ, ЕСХ, EDX, ESI, EDI, ЕВР, ESP, но не SI, DI, ВР или SP, которые можно было использовать в предыдущих ва­риантах.

7. Адресация по базе с индексированием и масштабированием

Это самая полная возможная схема адресации, в которую входят все случаи, рассмотренные ранее, как частные. Смещение может быть байтом, словом или двойным словом. Если ESP или ЕВР используются в роли базового регистра, селектор сегмента операн­да берется по умолчанию из регистра SS, во всех остальных случаях – из DS.

1. **Задание**
2. Протранслировать программу с созданием файла диагностических сообщений;

объяснить обнаруженные ошибки.

2. Снова протранслировать программу и скомпоновать загрузочный модуль.

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

**3. Входные данные**

vec1 38,37,36,35,31,32,33,34

vec2 70,80,-70,-80,50,60,-50,-60

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

**4.Ход выполнения программы**

Изначальный код программы с ошибками:

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

;

;

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 38,37,36,35,31,32,33,34

vec2 DB 70,80,-70,-80,50,60,-50,-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 2

Main ENDP

CODE ENDS

END Main

При сборке объектного файла были выявлены следующие ошибки и предупреждения:

LR2\_COMP.ASM(55): error A2052: Improper operand type

LR2\_COMP.ASM(62): warning A4031: Operand types must match

LR2\_COMP.ASM(66): warning A4031: Operand types must match

LR2\_COMP.ASM(67): error A2055: Illegal register value

LR2\_COMP.ASM(87): error A2046: Multiple base registers

LR2\_COMP.ASM(88): error A2047: Multiple index registers

LR2\_COMP.ASM(95): error A2006: Phase error between passes

48630 + 407942 Bytes symbol space free

2 Warning Errors

5 Severe Errors

1. Ошибка LR2\_COMP.ASM(55): error A2052: Improper operand type (Неверный тип операнда)

**Строка 55**: mov mem3,[bx]

Эта команда переводит информацию из сегмента памяти в другой, что невозможно в языке Assembler. В данном случае необходимо перевести информацию из памяти в регистр, а затем уже в необходимый сегмент информацию перевести из регистра. Код будет выглядеть так:

push ax

mov ax,[bx]

mov mem3, ax

pop ax

1. LR2\_COMP.ASM(62): warning A4031: Operand types must match

**Строка 62**: mov ax,vec2[di]

Объем регистра ax составляет 2 байта, в то время, как вес элемента в массиве составляет 1 байт, в данном случае надо вместо регистра ax использовать, например al:

mov al,vec2[di]

1. LR2\_COMP.ASM(67): error A2055: Illegal register value

**Строка 67**: mov ax,matr[bx\*4][di]

Здесь используется базово-индексная адресация. При данном типе адресации надо сначала изменить значение регистра, затем уже переводить информацию. Исправление будет выглядеть так:

mov cl,2

shl bx,cl

mov al,matr[bx][di]

1. LR2\_COMP.ASM(87): error A2046: Multiple base registers

**Строка 87**: mov ax,matr[bp+bx]

В базированной адресации необходимо указывать базовый регистр, затем производить смещение с помощью индексного. Так как здесь оба базовые, надо сначала сложить значения регистров, затем уже передавать информацию указателю из одного регистра.

add bp, bx

mov al,matr[bp]

1. LR2\_COMP.ASM(88): error A2047: Multiple index registers

**Строка 88**: mov ax,matr[bp+di+si]

Здесь ошибка, похожая на ошибку в п.4, за одним исключением- повторяются индексные регистры. В таком случае необходимо сначала а регистр di занести общую сумму, затем уже производить смещение.

add di,si

mov al,matr[bp+di]

Код программы с исправлениями:

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 38,37,36,35,31,32,33,34

vec2 DB 70,80,-70,-80,50,60,-50,-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]

push ax

mov ax,[bx]

mov mem3, ax

pop 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]

mov cl,2

shl bx,cl

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 bp, bx

mov al,matr[bp]

add di,si

mov al,matr[bp+di]

; Использование сегмента стека

push mem1

push mem2

mov bp,sp

mov dx,[bp]+2

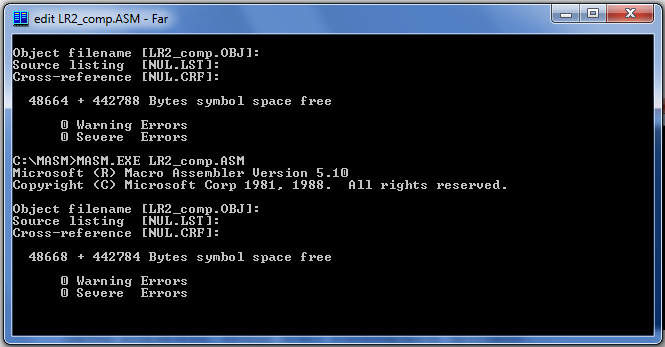
ret 2

Main ENDP

CODE ENDS

END Main

Сборка объектного файла:



Ошибки исправлены, теперь можно компилировать программу.

**5. Выполнение программы в режиме отладчика**

AX 0000 SI 0000 CS 15D3 IP 0000 Stack +0 0000

BX 0000 DI 0000 DS 15BE +2 0000

CX 00D5 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0018 SS 15CE FS 15BE +6 0000

0000 1E PUSH DS

AX 0000 SI 0000 CS 15D3 IP 0001 Stack +0 15BE

BX 0000 DI 0000 DS 15BE +2 0000

CX 00D5 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0016 SS 15CE FS 15BE +6 0000

0001 2BC0 SUB AX,AX

AX 0000 SI 0000 CS 15D3 IP 0003 Stack +0 15BE

BX 0000 DI 0000 DS 15BE +2 0000

CX 00D5 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0016 SS 15CE FS 15BE +6 0000

0003 50 PUSH AX

AX 0000 SI 0000 CS 15D3 IP 0004 Stack +0 0000

BX 0000 DI 0000 DS 15BE +2 15BE

CX 00D5 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0004 B8D015 MOV AX,15D0

AX 15D0 SI 0000 CS 15D3 IP 0007 Stack +0 0000

BX 0000 DI 0000 DS 15BE +2 15BE

CX 00D5 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0007 8ED8 MOV DS,AX

AX 15D0 SI 0000 CS 15D3 IP 0009 Stack +0 0000

BX 0000 DI 0000 DS 15D0 +2 15BE

CX 00D5 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0009 B8F401 MOV AX,01F4

AX 01F4 SI 0000 CS 15D3 IP 000C Stack +0 0000

BX 0000 DI 0000 DS 15D0 +2 15BE

CX 00D5 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

000C 8BC8 MOV CX,AX

AX 01F4 SI 0000 CS 15D3 IP 000E Stack +0 0000

BX 0000 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

000E B324 MOV BL,24

AX 01F4 SI 0000 CS 15D3 IP 000E Stack +0 0000

BX 0000 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

000E B324 MOV BL,24

AX 01F4 SI 0000 CS 15D3 IP 0010 Stack +0 0000

BX 0024 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0010 B7CE MOV BH,CE

AX 01F4 SI 0000 CS 15D3 IP 0012 Stack +0 0000

BX CE24 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0012 C7060200CEFF MOV [0002],FFCE

AX 01F4 SI 0000 CS 15D3 IP 0018 Stack +0 0000

BX CE24 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0018 BB0600 MOV BX,0006

AX 01F4 SI 0000 CS 15D3 IP 001B Stack +0 0000

BX 0006 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

001B A30000 MOV [0000],AX

AX 01F4 SI 0000 CS 15D3 IP 001E Stack +0 0000

BX 0006 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

001E 8A07 MOV AL,[BX]

AX 0126 SI 0000 CS 15D3 IP 0020 Stack +0 0000

BX 0006 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0020 50 PUSH AX

AX 0126 SI 0000 CS 15D3 IP 0021 Stack +0 0126

BX 0006 DI 0000 DS 15D0 +2 0000

CX 01F4 BP 0000 ES 15BE HS 15BE +4 15BE

DX 0000 SP 0012 SS 15CE FS 15BE +6 0000

0021 8B07 MOV AX,[BX]

AX 2526 SI 0000 CS 15D3 IP 0023 Stack +0 0126

BX 0006 DI 0000 DS 15D0 +2 0000

CX 01F4 BP 0000 ES 15BE HS 15BE +4 15BE

DX 0000 SP 0012 SS 15CE FS 15BE +6 0000

0023 A30400 MOV [0004],AX

AX 2526 SI 0000 CS 15D3 IP 0026 Stack +0 0126

BX 0006 DI 0000 DS 15D0 +2 0000

CX 01F4 BP 0000 ES 15BE HS 15BE +4 15BE

DX 0000 SP 0012 SS 15CE FS 15BE +6 0000

0026 58 POP AX

AX 0126 SI 0000 CS 15D3 IP 0027 Stack +0 0000

BX 0006 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0027 8A4703 MOV AL,[BX+03]

AX 0123 SI 0000 CS 15D3 IP 002A Stack +0 0000

BX 0006 DI 0000 DS 15D0 +2 15BE

CX 01F4 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

002A 8B4F03 MOV CX,[BX+03]

AX 0123 SI 0000 CS 15D3 IP 002D Stack +0 0000

BX 0006 DI 0000 DS 15D0 +2 15BE

CX 1F23 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

002D BF0200 MOV DI,0002

AX 0123 SI 0000 CS 15D3 IP 0030 Stack +0 0000

BX 0006 DI 0002 DS 15D0 +2 15BE

CX 1F23 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0030 8A850E00 MOV AL,[DI+000E]

AX 01BA SI 0000 CS 15D3 IP 0034 Stack +0 0000

BX 0006 DI 0002 DS 15D0 +2 15BE

CX 1F23 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0034 8A8D0E00 MOV CL,[DI+000E]

AX 01BA SI 0000 CS 15D3 IP 0038 Stack +0 0000

BX 0006 DI 0002 DS 15D0 +2 15BE

CX 1FBA BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0038 BB0300 MOV BX,0003

AX 01BA SI 0000 CS 15D3 IP 003B Stack +0 0000

BX 0003 DI 0002 DS 15D0 +2 15BE

CX 1FBA BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

003B 8A811600 MOV AL,[BX+DI+0016]

AX 01F9 SI 0000 CS 15D3 IP 003F Stack +0 0000

BX 0003 DI 0002 DS 15D0 +2 15BE

CX 1FBA BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

003F 8A891600 MOV CL,[BX+DI+0016]

AX 01F9 SI 0000 CS 15D3 IP 0043 Stack +0 0000

BX 0003 DI 0002 DS 15D0 +2 15BE

CX 1FF9 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0043 B102 MOV CL,02

AX 01F9 SI 0000 CS 15D3 IP 0045 Stack +0 0000

BX 0003 DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0045 D3E3 SHL BX,CL

AX 01F9 SI 0000 CS 15D3 IP 0047 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0047 8A811600 MOV AL,[BX+DI+0016]

AX 0101 SI 0000 CS 15D3 IP 004B Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

004B B8D015 MOV AX,15D0

AX 15D0 SI 0000 CS 15D3 IP 004E Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15BE HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

004E 8EC0 MOV ES,AX

AX 15D0 SI 0000 CS 15D3 IP 0050 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0050 268B07 MOV AX,ES:[BX]

AX 2221 SI 0000 CS 15D3 IP 0053 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0053 B80000 MOV AX,0000

AX 0000 SI 0000 CS 15D3 IP 0056 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0056 8EC0 MOV ES,AX

AX 0000 SI 0000 CS 15D3 IP 0058 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 0000 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0058 1E PUSH DS

AX 0000 SI 0000 CS 15D3 IP 0059 Stack +0 15D0

BX 000C DI 0002 DS 15D0 +2 0000

CX 1F02 BP 0000 ES 0000 HS 15BE +4 15BE

DX 0000 SP 0012 SS 15CE FS 15BE +6 0000

0059 07 POP ES

AX 0000 SI 0000 CS 15D3 IP 005A Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 1F02 BP 0000 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

**005A 268B4FFF MOV CX,ES:[BX-01]**

**AX 0000 SI 0000 CS 15D3 IP 005E Stack +0 0000**

**BX 000C DI 0002 DS 15D0 +2 15BE**

**CX 2120 BP 0000 ES 15D0 HS 15BE +4 0000**

**DX 0000 SP 0014 SS 15CE FS 15BE +6 0000**

**005E 91 XCHG AX,CX**

**AX 2120 SI 0000 CS 15D3 IP 005F Stack +0 0000**

**BX 000C DI 0002 DS 15D0 +2 15BE**

**CX 0000 BP 0000 ES 15D0 HS 15BE +4 0000**

**DX 0000 SP 0014 SS 15CE FS 15BE +6 0000**

**005F BF0200 MOV DI,0002**

**AX 2120 SI 0000 CS 15D3 IP 0062 Stack +0 0000**

**BX 000C DI 0002 DS 15D0 +2 15BE**

**CX 0000 BP 0000 ES 15D0 HS 15BE +4 0000**

**DX 0000 SP 0014 SS 15CE FS 15BE +6 0000**

0062 268901 MOV ES:[BX+DI],AX

AX 2120 SI 0000 CS 15D3 IP 0065 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0000 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0065 8BEC MOV BP,SP

AX 2120 SI 0000 CS 15D3 IP 0067 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0014 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0067 03EB ADD BP,BX

AX 2120 SI 0000 CS 15D3 IP 0069 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0069 3E8A861600 MOV AL,DS:[BP+0016]

AX 2115 SI 0000 CS 15D3 IP 006E Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

006E 03FE ADD DI,SI

AX 2115 SI 0000 CS 15D3 IP 0070 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0070 3E8A831600 MOV AL,DS:[BP+DI+0016]╙

AX 21D8 SI 0000 CS 15D3 IP 0075 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0075 FF360000 PUSH [0000]

AX 21D8 SI 0000 CS 15D3 IP 0079 Stack +0 01F4

BX 000C DI 0002 DS 15D0 +2 0000

CX 0000 BP 0020 ES 15D0 HS 15BE +4 15BE

DX 0000 SP 0012 SS 15CE FS 15BE +6 0000

0079 FF360200 PUSH [0002]

AX 21D8 SI 0000 CS 15D3 IP 007D Stack +0 FFCE

BX 000C DI 0002 DS 15D0 +2 01F4

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0010 SS 15CE FS 15BE +6 15BE

007D 8BEC MOV BP,SP

AX 21D8 SI 0000 CS 15D3 IP 007F Stack +0 FFCE

BX 000C DI 0002 DS 15D0 +2 01F4

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0010 SS 15CE FS 15BE +6 15BE

007F 8B5602 MOV DX,[BP+02]

AX 21D8 SI 0000 CS 15D3 IP 0082 Stack +0 FFCE

BX 000C DI 0002 DS 15D0 +2 01F4

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 01F4 SP 0010 SS 15CE FS 15BE +6 15BE

0082 CA0200 RET Far 0002

AX 21D8 SI 0000 CS 01F4 IP FFCE Stack +0 15BE

BX 000C DI 0002 DS 15D0 +2 0000

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 01F4 SP 0016 SS 15CE FS 15BE +6 0000

0065 8BEC MOV BP,SP

AX 2120 SI 0000 CS 15D3 IP 0067 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0014 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0067 03EB ADD BP,BX

AX 2120 SI 0000 CS 15D3 IP 0069 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0069 3E8A861600 MOV AL,DS:[BP+0016]

AX 2115 SI 0000 CS 15D3 IP 006E Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

006E 03FE ADD DI,SI

AX 2115 SI 0000 CS 15D3 IP 0070 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0070 3E8A831600 MOV AL,DS:[BP+DI+0016]

AX 21D8 SI 0000 CS 15D3 IP 0075 Stack +0 0000

BX 000C DI 0002 DS 15D0 +2 15BE

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0014 SS 15CE FS 15BE +6 0000

0075 FF360000 PUSH [0000]

AX 21D8 SI 0000 CS 15D3 IP 0079 Stack +0 01F4

BX 000C DI 0002 DS 15D0 +2 0000

CX 0000 BP 0020 ES 15D0 HS 15BE +4 15BE

DX 0000 SP 0012 SS 15CE FS 15BE +6 0000

0079 FF360200 PUSH [0002]

AX 21D8 SI 0000 CS 15D3 IP 007D Stack +0 FFCE

BX 000C DI 0002 DS 15D0 +2 01F4

CX 0000 BP 0020 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0010 SS 15CE FS 15BE +6 15BE

007D 8BEC MOV BP,SP

AX 21D8 SI 0000 CS 15D3 IP 007F Stack +0 FFCE

BX 000C DI 0002 DS 15D0 +2 01F4

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 0000 SP 0010 SS 15CE FS 15BE +6 15BE

007F 8B5602 MOV DX,[BP+02]

AX 21D8 SI 0000 CS 15D3 IP 0082 Stack +0 FFCE

BX 000C DI 0002 DS 15D0 +2 01F4

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 01F4 SP 0010 SS 15CE FS 15BE +6 15BE

0082 CA0200 RET Far 0002

AX 21D8 SI 0000 CS 01F4 IP FFCE Stack +0 15BE

BX 000C DI 0002 DS 15D0 +2 0000

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 01F4 SP 0016 SS 15CE FS 15BE +6 0000

FFCE 7406 JZ FFD6

AX 21D8 SI 0000 CS 01F4 IP FFD1 Stack +0 15BE

BX 000C DI 0002 DS 15D0 +2 0000

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 01F4 SP 0016 SS 15CE FS 15BE +6 0000

FFD0 B080 MOV AL,80

AX 2180 SI 0000 CS 01F4 IP FFD2 Stack +0 15BE

BX 000C DI 0002 DS 15D0 +2 0000

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 01F4 SP 0016 SS 15CE FS 15BE +6 0000

FFD2 E6A0 OUT [A0],AL

FFDA FB STI

AX 2180 SI 0000 CS 01F4 IP FFDB Stack +0 15BE

BX 000C DI 0002 DS 15D0 +2 0000

CX 0000 BP 0010 ES 15D0 HS 15BE +4 0000

DX 01F4 SP 0016 SS 15CE FS 15BE +6 0000

FFDB F606F99E02 TEST [9EF9],02

В итоге программа успешно скомпилировалась, и закончила свою работу

1. **Вывод:**

Я был ознакомлен с режимами адресации, нашел и исправил ошибки в исходнике, и, скомпилировав, получил корректно работающую программу.