**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Организация систем и ЭВМ»**

Тема «**Изучение режимов адресации и формирования исполнительного адреса**»

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1303 |  | Сырцева Д.Д |
| Преподаватель |  | Ефремов М.А |

Санкт-Петербург

2022

## Цель работы.

Изучить режимы адресации. Ознакомиться с тем, как происходит работа с ними на языке программирования ассемблер.

## 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. Необходимо составить протокол выполнения программы в пошаговом режиме отладчика по типу таблицы 1 предыдущей лабораторной работы и подписать его у преподавателя. На защите студенты должны уметь объяснить результат выполнения каждой команды с учетом используемого вида адресации. Результаты, полученные с помощью отладчика, не являются объяснением, а только должны подтверждать ваши объяснения.

## Выполнение работы. Протокол работы на компьютере. Протоколы пошагового исполнения каждой из программ под управлением отладчика.

**Ход работы:**

1. Получен вариант набора значений исходных данных из файла lr2.dat(2

вариант), приведенного в каталоге Задания и занесены свои данные вместо значений, указанных в приведенной ниже программе.

1. Программа протранслирована «см. рис. 1». Создан файл

диагностических сообщений, были обнаружены ошибки и закомментированы соответствующие операторы в тексте программы.

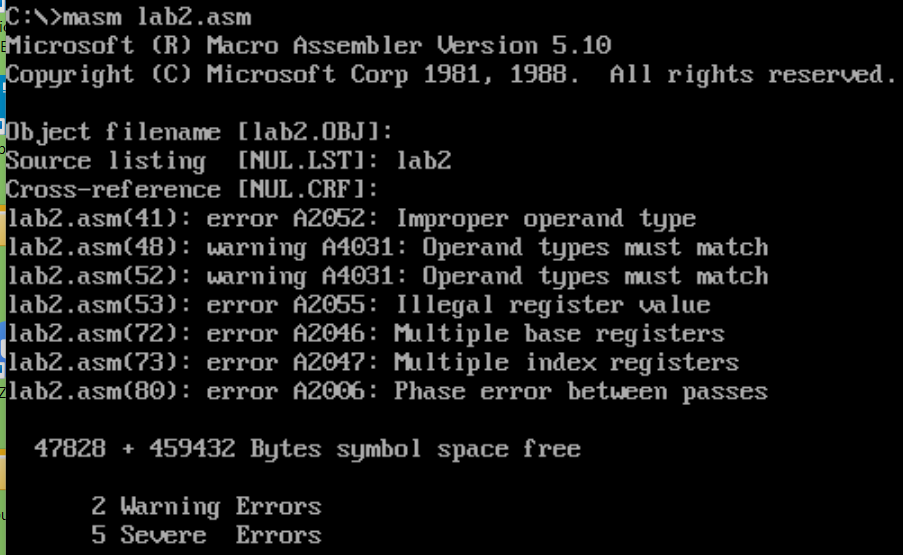


Рисунок1

Выявленные ошибки:

lab2.asm(41): error A2052: Improper operand type

mov mem3, [bx]

Причина ошибки: данная ошибка возникла вследствие того, что один из операндов должен быть либо регистром, либо значением, а другой операнд ячейкой в памяти, то есть машинные команды не могут работать одновременно с двумя ячейками в памяти.

lab2.asm(48): warning A4031: Operand types must match

mov cx, vec2[di]

lab2.asm(52): warning A4031: Operand types must match

mov cx, matr[bx][di]

Причина обеих ошибок: несоответствие типов операндов. Пытаемся поместить 1 байт(matr[bx][di]) в cx (в слово – 2 байта).

lab2.asm(53): error A2005: Illegal register value

mov ax, matr[bx\*4][di]

Причина ошибки: масштабирование регистра bx при индексации с базированием и индексированием. Нельзя масштабировать адрес на наборе инструкций.

lab2.asm(72): error A0246: Multiple base registers

mov ax, matr[bp+bx]

Причина ошибки: оба регистра являются базовыми. Исполняемый адрес при адресации с базированием и индексированием – сумма адресов находящихся в базовом и индексном регистрах.

lab2.asm(73): error A2047: Multiple base register

mov ax, matr[bp+di+si]

Причина ошибки: берется 2 индексных и 1 базовый регистр. В косвенной базовой индексной адресации со смещением эффективный адрес формируется как сумма базового и индексного регистра, к которым прибавляется значение поля смещения в команде.

lab2.asm(80): error A2006: Phase error between passes

Main ENDP

Причина ошибки: наличие ошибок в функции main

1. Снова протранслирована программа и скомпонован загрузочный модуль.
2. Программа выполнена в пошаговом режиме под управлением отладчика

с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

Запущено выполнение программы под управлением отладчика. Полученные результаты приведены в таблице 1.

Таблица 1 – Протокол пошагового исполнения lab2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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  (SP) =0018  (IP) = 0000 | STACK(+0)=19F5  (SP) = 0016  (IP) = 0001 |
| 0001 | SUB AX, AX | 2BCO | (AX) = 0000  (IP) = 0001 | (AX) = 0000  (IP) = 0003 |
| 0003 | PUSH AX | 50 | STACK(+0)=19F5  STACK(+2)=0000  (SP) =0016  (IP) = 0003 | STACK(+0)=0000  STACK(+2)=19F5  (SP) =0014  (IP) = 0004 |
| 0004 | MOV AX,1A07 | B8071A | (AX) = 0000  (IP) = 0004 | (AX) = 1A07  (IP) = 0007 |
| 0007 | MOV DS,AX | 8ED8 | DS=19F5  IP=0007 | (DS)=1A07  (IP) = 0009 |
| 0009 | MOV AX,01F4 | B8F401 | (AX) = 1A07  (IP) = 0009 | (AX) = 01F4  (IP) = 000C |
| 000C | MOV CX,AX | 8BC8 | (CX) = 00B0  (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 | C7060200CEFF | (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)=0105  (IP)=0020 |
| 0020 | MOV AL,[BX+03] | 8A4703 | (IP)=0020 (AX)=0105 | (IP)=0023  (AX)=0108 |
| 0023 | MOV CX, [BX+03] | 8B4F03 | (CX)= 01F4  (IP)= 0023 | (CX)= 0C08  (IP)= 0026 |
| 0026 | MOV DI, 0002 | BF0200 | (DI)= 0000  (IP)= 0026 | (DI)= 0002  (IP)= 0029 |
| 0029 | MOV AL,[000E+DI] | 8A850E00 | (AX)= 0108  (IP) = 0029 | (AX)= 0114  (IP) = 002D |
| 002D | MOV BX, 0003 | BB0300 | (IP) = 002D  (BX)= 0006 | (IP) = 0030  (BX) = 0003 |
| 0030 | MOV AL,[0016+BX+DI] | 8A811600 | (IP) = 0030  (AX) =0114 | (IP) = 0034  AX =0103 |
| 0034 | MOV AX, 1A07 | B8071A | (AX)=0103  (IP)=0034 | (AX)=1A07  (IP)=0037 |
| 0037 | MOV ES, AX | 8EC0 | (ES) =19F5  (IP)=0037 | (ES) =1A07  (IP)=0039 |
| 0039 | MOV AX, ES:[BX] | 268B07 | (AX) = 1A07  (IP) = 0039 | (AX) = 00FF  (IP) = 003C |
| 003C | MOV AX, 0000 | B80000 | (AX)= 00FF (IP)=003C | (AX)= 0000  (IP)=003F |
| 003F | MOV ES, AX | 8EC0 | (ES) =1A07  ( IP)= 003F | (ES) =0000  ( IP)= 0041 |
| 0041 | PUSH DS | 1E | (IP)=0041  (SP)=0014  STACK (+0) =  0000  STACK (+2) = 19F5  STACK (+4) =0000 | (IP)=0042  (SP)=0012  STACK (+0) =  1A07  STACK (+2) = 0000  STACK (+4) =19F5 |
| 0042 | POP ES | 07 | (IP)=0042  (SP)=0012  (ES)=0000  STACK (+0) =  1A07  STACK (+2) = 0000  STACK (+4) =19F5 | (IP)=0043  (SP)=0014  (ES)=1A07  STACK (+0) =  0000  STACK (+2) = 19F5  STACK (+4) =0000 |
| 0043 | MOV CX, ES:[BX-1] | 268B4FFF | (CX) = 0C08  (IP) = 0043 | (CX) = FFCE  (IP) = 0047 |
| 0047 | XCHG AX, CX | 91 | (AX) = 0000 (CX) = FFCE (IP)=0047 | (AX) = FFCE  (CX) = 0000 (IP)=0048 |
| 0048 | MOV DI, 0002 | BF0200 | (IP) = 0048 (DI)=0002 | (IP) = 004B  (DI)=0002 |
| 004B | MOV ES:[BX+DI], AX | 268901 | (IP) = 004B | (IP) = 004E |
| 004E | MOV BP, SP | 8BEC | (IP) = 004E  (BP) = 0000 | (IP) = 0050  (BP) = 0014 |
| 0050 | PUSH [0000] | FF360000 | (IP)=0050  (SP)=0014  STACK (+0) =  0000  STACK (+2) = 19F5  STACK (+4) =0000 | (IP)=0054  (SP)=0012  STACK (+0) =  01F4  STACK (+2) = 0000  STACK (+4) =19F5 |
| 0054 | PUSH [0002] | FF360200 | (IP)=0054  (SP)=0012  STACK (+0) =  01F4  STACK (+2) = 0000  STACK (+4) =19F5  STACK (+6) = 0000 | (IP)=0058  (SP)=0010  STACK (+0) =  FFCE  STACK (+2) = 01F4  STACK (+4) =0000  STACK (+6) = 19F5 |
| 0058 | MOV BP, SP | 8BEC | (IP) = 0058  (BP) = 0014 | (IP) = 005A  (BP) = 0010 |
| 005A | MOV DX, [BP+02] | 8B5602 | (IP) = 005A  ( DX) = 0000 | (IP) = 005D  ( DX) = 01F4 |
| 005D | RET Far 0002 | CA0200 | (IP)=005D  (SP)=0010  (CS)=1A0A  STACK (+0) =  FFCE  STACK (+2) = 01F4  STACK (+4) =0000  STACK (+6) = 19F5 | (IP)=FFCE  (SP)=0016  (CS)=01F4  STACK (+0) =  19F5  STACK (+2) = 0000  STACK (+4) =0000  STACK (+6) = 0000 |

## Вывод.

В ходе выполнения лабораторной работы были получены основные навыки по работе с режимами адресации на языке программирования ассемблер.

## ПРИЛОЖЕНИЕ А

## Текст исходного файла программы lab2.

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 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

; 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]

; 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

### Текст файла диагностических сообщений lab2.lst.

Microsoft (R) Macro Assembler Version 5.10 10/16/22 12:01:1

Page 1-1

= 0024 EOL EQU '$'

= 0002 ind EQU 2

= 01F4 n1 EQU 500

=-0032 n2 EQU -50

; Стек программы

0000 AStack SEGMENT STACK

0000 000C[ DW 12 DUP(?)

????

]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

; Директивы описания даннύ

х

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

; Код программы

0000 CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура

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

; ПРОВЕРКА РЕЖИМОВ АДРЕСА΍

ИИ НА УРОВНЕ СМЕЩЕНИЙ

; Регистровая адресация

0009 B8 01F4 mov ax,n1

000C 8B C8 mov cx,ax

000E B3 24 mov bl,EOL

0010 B7 CE mov bh,n2

; Прямая адресация

0012 C7 06 0002 R FFCE mov mem2,n2

0018 BB 0006 R mov bx,OFFSET vec1

001B A3 0000 R mov mem1,ax

; Косвенная адресация

001E 8A 07 mov al,[bx]

; mov mem3,[bx]

; Базированная адресация

0020 8A 47 03 mov al,[bx]+3

0023 8B 4F 03 mov cx,3[bx]

; Индексная адресация

Microsoft (R) Macro Assembler Version 5.10 10/16/22 12:01:1

Page 1-2

0026 BF 0002 mov di,ind

0029 8A 85 000E R mov al,vec2[di]

; mov cx,vec2[di]

; Адресация с базирование΍

и индексированием

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]

; ПРОВЕРКА РЕЖИМОВ АДРЕСА΍

ИИ С УЧЕТОМ СЕГМЕНТОВ

; Переопределение сегмент

а

; ------ вариант 1

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

; ------ вариант 2

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

; ------ вариант 3

0048 BF 0002 mov di,ind

004B 26: 89 01 mov es:[bx+di],ax

; ------ вариант 4

004E 8B EC mov bp,sp

; mov ax,matr[bp+bx]

; mov ax,matr[bp+di+si]

; Использование сегмента ύ

тека

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 10/16/22 12:01:1

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 lab2d

@VERSION . . . . . . . . . . . . TEXT 510

82 Source Lines

82 Total Lines

19 Symbols

47832 + 459428 Bytes symbol space free

0 Warning Errors

0 Severe Errors