

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 и систем»
Тема: Изучение режимов адресации и формирования
исполнительного адреса
Вариант 6

Студент гр. 0382

Злобин А. С.

Преподаватель

Евфремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучить режимы адресации процессора Intel X86 при помощи

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.

5. Результаты выполнения программы под управлением отладчика должны быть подписаны преподавателем и представлены в отчете.

Выполнение работы.

Данные для 6 варианта:

`vec1` 18,17,16,15,11,12,13,14

`vec2` 30,40,-30,-40,10,20,-10,-20

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

Были обнаружены следующие ошибки:

- `mov mem3, [bx]` - Невозможно перемещение из одной ячейки памяти в другую.
- `mov cx, vec2[di]` - Попытка перемещения DB в регистр размера DW.
- `mov cx, matr[bx][di]` - Ошибка, аналогичная предыдущему случаю.
- `mov ax, matr[bx*4][di]` - Умножение 2-х байтовых регистров запрещено.

- `mov ax, matr[bp+bx]` - Нельзя использовать несколько базовых регистров для адресации.
- `mov ax, matr[bp+di+si]` - Нельзя использовать несколько индексных регистров для адресации

Таблица 1. - Протокол отладки

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	Stack +0 19F5 SP 0016	Stack +0 0000 +2 19F5 SP 0014
0004	MOV AX, 1A07	B8071A	AX 0000	AX 1A07
0007	MOV DS, AX	8ED8	DS 19F5	DS 1A07
0009	MOV AX, 01F4	B8F401	AX 1A07	AX 01F4
000C	MOV CX, AX	8BC8	CX 00B0	CX 01F4
000E	MOV BL, 24	B324	BX 0000	BX 0024
0010	MOV BH, CE	B7CE	BX 0024	BX CE24
0012	MOV [0002], FFCE	C7060200CEFF	DS:0000 00 00 00 00 00 00	DS:0000 00 00 CE FF 00 00
0018	MOV BX, 0006	BB0600	BX CE24	BX 0006
001B	MOV [0000], AX	A30000	DS:0000 00 00 CE FF 00 00	DS:0000 F4 01 CE FF 00 00
001E	MOV AL, [BX]	8A07	AX 01F4	AX 011C
0020	MOV AL, [BX+03]	8A4703	AX 011C	AX 0119
0020	MOV AL, [BX+03]	8A4703	AX 0112	AX 010F
0023	MOV CX, [BX+03]	8B4F03	CX 01F4	CX 0B0F
0026	MOV DI, 0002	BF0200	DI 0000	DI 0002

0029	MOV AK, [000E+DI]	8A850E00	AX 010F	AX 01E2
002D	MOV BX, 0003	BB0300	BX 0006	BX 0003
0030	MOV AL, [0016+BX+DI]	8A811600	AX 01E2	AX 01FF
0034	MOV AX, 1A07	B8071A	AX 01FF	AX 1A07
0037	MOV ES, AX	8EC0	ES 19F5	ES 1A07
0039	MOV AX, ES:[BX]	268B07	AX 1A07	AX 00FF
003C	MOV AX, 0000	B80000	AX 00FF	AX 0000
003F	8EC0	MOV ES, AX	ES 1A07	ES 0000
0041	PUSH DS	1E	SP 0014 Stack +0 0000 +2 19F5	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-01]	268B4FFF	CX 0B0F	CX FFCE
0047	XCHG AX, CX	91	AX 0000 CX FFCE	AX FFCE CX 0000
0048	MOV DI, 0002	BF0200	DI 0002	DI 0002
004B	MOV ES:[BX+DI], AX	268901	DS:0000 F4 01 CE FF 00 00 00	DS:0000 F4 01 CE FF 00 CE FF
004E	MOV BP, SP	8BEC	BP 0000	BP 0014
0050	PUSH [0000]	FF360000	SP 0014 Stack +0 0000 +2 19F5	SP 0012 Stack +0 01F4 +2 0000 +4 19F5
0054	PUSH [0002]	FF360200	SP 0012	SP 0010

			Stack +0 01F4	Stack +0 FFCE
			+2 0000 +4 19F5	+2 01F4 +4 0000 +6 19F5
0058	MOV BP, SP	8BEC	BP 0014	BP 0010
005A	MOV DX, [BP+02]	8B5602	DX 0000	DX 01F4
005D	RET Far 0002	CA0200	CS 1A0A IP 005D SP 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	CS 01F4 IP FFCE SP 0016 Stack +0 19F5 +2 0000

Выводы.

В ходе работы были изучены режимы адресации процессоров Intel X86, исправлены ошибки в программе, составлен протокол отладки программы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab2.asm

```
; Программа изучения режимов адресации процессора IntelX86

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 18,17,16,15,11,12,13,14
vec2 DB 30,40,-30,-40,10,20,-10,-20
matr DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-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] ; - Перемещение из памяти в память запрещено на
архитектурном уровне.

; Базированная адресация
    mov al,[bx]+3
    mov cx,3[bx]

; Индексная адресация
    mov di,ind
    mov al,vec2[di]
    mov cx,vec2[di] ; - Попытка перемещения DB в регистр размера DW.

; Адресация с базированием и индексированием
    mov bx,3
    mov al,matr[bx][di]
    mov cx,matr[bx][di] ; - Не совпадают размеры операторов
    mov ax,matr[bx*4][di] ; - Умножение 2-х байтовых регистров
запрещено.

; 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] ; - Нельзя использовать несколько базовых
регистров для адресации.
    mov ax,matr[bp+di+si] ; - Нельзя использовать несколько индексных
регистров для адресации

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

```

ПРИЛОЖЕНИЕ А

Листинги

До исправления

Название файла: LAB2_ERR.LST

Microsoft (R) Macro Assembler Version 5.10
10/5/21 16:57:56

Page

1-1

```
; PµCБPsPiCБP°PjPjP° PëP·CíC†PµPSPëCЦ CБPµP¶PëP
jPsPI P°PrCБPµCíP°C†PëPë PiCБPsC†PµCíCíPsCБP° I
ntelX86

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

; PŸC,PµPe PiCБPsPiCБP°PjPjC<

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

0018                AStack ENDS

; P"P°PSPSC<Pµ PiCБPsPiCБP°PjPjC<

0000                DATA SEGMENT

; P"PëCБPµPeC,PëPIC< PsPiPëCíP°PSPëCЦ PrP°PSPSC
<C...

0000 0000                mem1 DW 0
0002 0000                mem2 DW 0
0004 0000                mem3 DW 0

0006 12 11 10 0F 0B 0C    vec1 DB 18,17,16,15,11,12,13,14
```



```

0D 0E
000E 1E 28 E2 D8 0A 14    vec2 DB 30,40,-30,-40,10,20,-10,-20
F6 EC
0016 FC FD 01 02 FE FF    matr DB
-4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5
03 04 05 06 07 08
F8 F9 FA FB

```

```
0026 DATA ENDS
```

```

; PŁPsPr PiCŁPsPiCŁP°PjPjC<
0000 CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack

```

```
; P"PsP»PsPIPsP°CŁ PiCŁPsC†PμPrC†CŁP°
```

```

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

```

```

; PŁP PhP'P•P PŁPŁ P P•P-P~PŁPhP' PŁP"P P•PŸPŁP
|P~P~ PŁPŁ PJP PhP'PŁP• PŸPŁP•P©P•PŁP~P™
; P PμPiPŁC†C, CŁPsPIPs°CŁ P°PrCŁPμC†P°C†PŁCŁ

```

```

0009 B8 01F4 mov ax,n1
000C 8B C8 mov cx,ax
Microsoft (R) Macro Assembler Version 5.10

```

10/5/21 16:57:56

Page

1-2

```

000E B3 24 mov bl,EOL
0010 B7 CE mov bh,n2

```

```

; PŁCŁCŁPjP°CŁ P°PrCŁPμC†P°C†PŁCŁ
0012 C7 06 0002 R FFCE mov mem2,n2
0018 BB 0006 R mov bx,OFFSET vec1
001B A3 0000 R mov mem1,ax

```

```

; PъPсCfPIPμPSPSP°CЦ P°PrCъPμCfP°C†PëCЦ
001E 8A 07 mov al,[bx]
mov mem3,[bx] ; - PμPμCъPμPjPμC%PμPSPëP
μ PëP· PìP°PjCЦC,Pë PI PìP°PjCЦC,Чъ P·P°PìCъPμC
%PμPSPs PSP° P°CъC...PëC,PμPëC,CfCъPSPsPj CfCъPsP
IPSPμ.

```

lab2.asm(54): error A2052: Improper operand type

```

; P`P°P·PëCъPsPIP°PSPSP°CЦ P°PrCъPμCfP°C†PëCЦ
0020 8A 47 03 mov al,[bx]+3
0023 8B 4F 03 mov cx,3[bx]

; P~PSPPrPμPëCfPSP°CЦ P°PrCъPμCfP°C†PëCЦ
0026 BF 0002 mov di,ind
0029 8A 85 000E R mov al,vec2[di]
002D 8B 8D 000E R mov cx,vec2[di] ; - PμPsPìC<C,PëP°

```

PìPμ

```

CъPμPjPμC%PμPSPëCЦ DB PI CъPμPìPëCfC,Чъ CъP°P·P
jPμCъP° DW.

```

lab2.asm(63): warning A4031: Operand types must match

```

; PђPrCъPμCfP°C†PëCЦ Cf P±P°P·PëCъPsPIP°PSPëPμP
j Pë PëPSPPrPμPëCfPëCъPsPIP°PSPëPμPj
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] ; -

```

lab2.asm(68): warning A4031: Operand types must match

```

003C 8B 85 0022 R mov ax,matr[bx*4][di] ; -

```

PJPjPSPsP¶PμP

```

SPëPμ 2-C... P±P°P№C,PsPIC<C... CъPμPìPëCfC,ЧъPsPI
P·P°PìCъPμC%PμPSPs.

```

lab2.asm(69): error A2055: Illegal register value

```

; PμP PђP'P·P PъPђ P P·P-P~PђPђP' PђP"P P·PŸPђP
|P~P~ PŸ PJPSP·PŸPђPђ PŸP·P"PђP·PкPŸPђP'
; PμPμCъPμPsPìCъPμPrPμP»PμPSPëPμ CfPμPìPjPμPSC,
P°
; ----- PIP°CъPëP°PSC, 1

```

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

; ----- PIP°CЪPӨP°PSC, 2

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

; ----- PIP°CЪPӨP°PSC, 3

0054 BF 0002 mov di,ind

0057 26: 89 01 mov es:[bx+di],ax

Microsoft (R) Macro Assembler Version 5.10

10/5/21 16:57:56

Page

1-3

; ----- PIP°CЪPӨP°PSC, 4

005A 8B EC mov bp,sp

005C 3E: 8B 86 0016 R mov ax,matr[bp+bx] ; -

РќРµР»CЪP·CЎ PөC

ЃРїРѕР»CЪP·PѕРІР°C, CЪ PSPµCЃРёРѕРѕP»CЪРёРѕ P±Р°Р·
PѕРІC<C... CЪРµРїРёPөCЃC, CЪPѕРІ РґР»»CЎ Р°РґCЪРµCЃР°
C†PөPө.

lab2.asm(90): error A2046: Multiple base registers

0061 3E: 8B 83 0016 R mov ax,matr[bp+di+si] ; -

РќРµР»CЪP·CЎ

PөCЃРїРѕРѕP»CЪP·PѕРІР°C, CЪ PSPµCЃРёРѕРѕP»CЪРёРѕ PөP
SPґРµРёCЃРѕP<C... CЪРµРїРёPөCЃC, CЪPѕРІ РґР»»CЎ Р°РґC
ЪРµCЃР°C†PөPө

lab2.asm(91): error A2047: Multiple index registers

; P~CЃРїРѕРѕP»CЪP·PѕРІР°PSPөРµ CЃРµРїРјРµPSC, P° C
ЃC, РµРёР°°

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
lab2.asm(99): error A2006: Phase error between passes
0076                      CODE ENDS
                      END Main

```

```

Microsoft      (R)      Macro      Assembler      Version      5.10
10/5/21 16:57:56

```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
	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 lab2
@VERSION . . . . . TEXT 510
```

```
101 Source Lines
101 Total Lines
19 Symbols
```

```
47814 + 459446 Bytes symbol space free
```

```
2 Warning Errors
5 Severe Errors
```

После исправления

После исправления: LAB2.LST

```
Microsoft (R) Macro Assembler Version 5.10
10/5/21 16:50:59
Page
1-1
```

```
; P4C7PsPiC7P°PjPjP° P6P·CfC+PμPSP6CЦ C7PμP¶P6P
jPsPI P°PrC7PμCfP°C+P6P6 PiC7PsC+PμCfCfPsC7P° I
ntelX86
```

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

```
; PŸC,PμP6 PiC7PsPiC7P°PjPjC<
```

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

```

0018                                AStack ENDS

                                ; P"P°PSPSC<Pµ PİCḂPsPİCḂP°PjPjC<

0000                                DATA SEGMENT

                                ; P"PḂCḂPµPeC,PḂPIC< PsPİPḂCÍP°PSPḂCḂ PrP°PSPSC
                                <C...

0000    0000                                mem1 DW 0
0002    0000                                mem2 DW 0
0004    0000                                mem3 DW 0

0006    12 11 10 0F 0B 0C    vec1 DB 18,17,16,15,11,12,13,14
                                0D 0E
000E    1E 28 E2 D8 0A 14    vec2 DB 30,40,-30,-40,10,20,-10,-20
                                F6 EC

0016    FC FD 01 02 FE FF    matr                                DB
-4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5
                                03 04 05 06 07 08
                                F8 F9 FA FB

0026                                DATA ENDS

                                ; PḂPsPr PİCḂPsPİCḂP°PjPjC<

0000                                CODE SEGMENT

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

                                ; P"PsP»PsPIPSP°CḂ PİCḂPsC†PµPrCÍCḂP°

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

                                ; PµP PhP'P•P PḂPḂ P P•P-P~PḂPhP' PhP"P P•PŸPhP
                                ¦P~P~ PÍPḂ PJP PhP'PÍP• PŸPḂP•P©P•PÍP~P™
                                ; P PµPİPḂCÍC,CḂPsPIP°CḂ P°PrCḂPµCÍP°C†PḂCḂ

0009    B8 01F4                                mov ax,n1
000C    8B C8                                mov cx,ax

```

1-2

```

000E B3 24 mov bl,EOL
0010 B7 CE mov bh,n2

; PµC7CµPjP°Cµ P°PrC7PµC7P°C†PëCµ
0012 C7 06 0002 R FFCE mov mem2,n2
0018 BB 0006 R mov bx,OFFSET vec1
001B A3 0000 R mov mem1,ax

; PµPSc7PIPµPSPSP°Cµ P°PrC7PµC7P°C†PëCµ
001E 8A 07 mov al,[bx]
; mov mem3,[bx] - PµPµC7PµPjPµC%PµPSPëP
µ PëP· PìP°PjCµC,Pë PI PìP°PjCµC,C7 P·P°PìC7PµC
%PµPSPs PSP° P°C7C...PëC,PµPeC,C7C7PSPsPj C7C7PsP
IPSPµ.

; P`P°P·PëC7PsPIP°PSPSP°Cµ P°PrC7PµC7P°C†PëCµ
0020 8A 47 03 mov al,[bx]+3
0023 8B 4F 03 mov cx,3[bx]

; P~PSPPrPµPeC7PSP°Cµ P°PrC7PµC7P°C†PëCµ
0026 BF 0002 mov di,ind
0029 8A 85 000E R mov al,vec2[di]
; mov cx,vec2[di] - PµPsPìC<C,PeP° PìPµ
C7PµPjPµC%PµPSPëCµ DB PI C7PµPìPëC7C,C7 C7P°P·P
jPµC7P° DW.

; PjPrC7PµC7P°C†PëCµ C7 P±P°P·PëC7PsPIP°PSPëPµP
j Pë PëPSPPrPµPeC7PëC7PsPIP°PSPëPµPj
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] - PJPjPSPsPjPµP
SPëPµ 2-C... P±P°PN°C,PsPIC<C... C7PµPìPëC7C,C7PsPI
P·P°PìC7PµC%PµPSPs.

```



```

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
10/5/21 16:50:59

```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
	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	lab2
@VERSION	TEXT	510

101 Source Lines

101 Total Lines

19 Symbols

47826 + 459434 Bytes symbol space free

0 Warning Errors

0 Severe Errors