

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №4
з дисципліни «Архітектура комп'ютера»
Тема: «Дослідження роботи стеку МПС
архітектури IA-32 (X86) у Real Address Mode»
Команда 3

Виконали: студенти групи ІТ-01

Дмитрієва І.І.

Гончаренко А.А.

Чорній В.І.

Викладач: Бердник Ю.М.

Захищено з балом _____

Дата здачі: 08.04.2021

Київ 2021

Мета лабораторної роботи

Мета лабораторної роботи полягає у набутті впевнених знань і навичок з розробки ПЗ на Асемблері у ході якої застосовуються знання архітектури комп'ютерів.

Хід лабораторної роботи:

- вивчити відеопам'ять архітектури IA5-32 у real address mode;
- виконати повний цикл розробки, тестування і налагодження програмного забезпечення;
- зберегти отриману програму, зробити висновки щодо необхідності знань архітектури комп'ютера у ході розробки ПЗ.

Завдання для лабораторної роботи №4:

1. Створити масив `arr_stack` розміром 16×16 , що має тип даних слово у сегменті даних (рис. 4.6). Заповнити масив випадковими числами. Варіанти реалізації у п.1.1, 1.2, 1.3 за вибором студента.

1.1 Перший варіант заповнення - взяти числа з Додатку 2, з таблиці випадкових чисел.

1.2 Другий варіант заповнення складається з двох кроків. Перший крок - написати програму на високорівневій мові і створити файл з випадковими числами. Другий крок - заповнити цими числами масив `arr_stack` розміром 16×16 .

1.3 Завдання підвищеної складності. Написати на Асемблері процедуру для створення псевдо-випадкових чисел. Користуючись статистичними методами дослідження визначити закон розподілу отриманих чисел. Заповнити цими числами масив `arr_stack` розміром 16×16 .

2. Написати процедуру, що робить кілька однакових копій даного масиву у сегмент даних. Кількість копій відповідає номеру варіанта. Для створення копій використати циклічні конструкції Асемблера або інші варіанти.

3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	1	5	0	7	1	9	9	9	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Рис. 4.6. Вигляд масиву

3. Розрахувати розмір стека для розміщення масиву. Користуючись відповідними стековими командами Асемблера (push, pop) перенести значення однієї з копій масиву до стека.
4. Заповнити у стеку рядок числами дня, місяця, року народження студентів робочої групи. Номер рядка має відповідати номеру варіанта. Скористатися базовою адресацією. Визначити фізичну адресу крайніх елементів рядка з числами народження користуючись TD та записати це до звіту.

Для вирішення завдання був написаний код на Асемблері.

Посилання на GitHub:

- [Гончаренко Андрій](#)
- [Дмитрієва Ірина](#)
- [Чорній Владислав](#)

У сегменті даних оголошується масив 16x16, ініціалізований випадковими числами типу слово. Для спрощення коду використовується макрос.

```

1  TITLE Vihidni kod 2.1
2  ;-----
3  ;LP №4, команда №3, IT-01
4  ;-----
5  ;-----I.ЗАГОЛОВОК ПРОГРАМИ-----
6  IDEAL
7  MODEL SMALL
8  STACK 16384
9  ;-----II.МАКРОСИ-----
10 ; Складний макрос для ініціалізації
11 MACRO M_Init ; Початок макросу
12     mov ax,@data ; @data ідентифікатор, що створюється директивою model
13     mov ds, ax ; Завантаження початку сегменту даних в регістр ds
14     mov es, ax ; Завантаження початку сегменту даних в регістр es
15 ENDM M_Init
16
17 ;-----III.ПОЧАТОК СЕГМЕНТУ ДАНИХ-----
18 DATASEG
19
20 ;Оголошення двовимірному експериментального масиву 16x16
21 array2Db DW '3', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
22           DW '3', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
23           DW '3', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
24           DW '3', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
25           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
26           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
27           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
28           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
29           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
30           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
31           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
32           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
33           DW '2', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
34           DW '3', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
35           DW '3', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
36           DW '3', '4', '8', '9', '7', '3', '0', '3', '9', '0', '7', '6', '1', '0', '8', '6'
37
38
39
40 exCode DB 0
41 CODESEG

```

Виконується код, відповідно до завдання. Використовується базова адресація, викликаються процедури:

- `copy`, для створення однакових копій масиву у сегмент даних. За варіантом команди масив копіюється три рази.
- `set_birthday`, перезаповнює треті рядки масиву і двох його копій (команда 3) датою і роком народження студента;
- `to_stack`, розміщує елементи заданого масиву з сегменту даних у сегмент стеку.

Викликаються функції переривання DOS int 21h: очікування вводу з клавіатури і завершення програми.

```
43 ;-----VI. ПОЧАТОК СЕГМЕНТУ КОДУ-----
44 Start:
45 M_Init
46 ;Способи адресації - Базова адресація. Призначена для роботи з масивами
47
48 xor si, si ; обнулюємо si, що використовується для адресації
49 call copy ; викликаємо процедуру copy для копіювання у сегмент даних
50 add si, 64 ; додаємо до si 64 для перенесення до початку наступного масиву
51 call copy ; повторюємо минулий пункт
52 add si, 64
53 call copy
54
55 call set_birthday ; додаємо до 3 ряду першого масиву день народження
56
57 call to_stack ; додаємо
58
59 ;-----очікування натискання клавіші-----
60 mov ah, 01h
61 int 21h
62 ;-----завершення роботи програми-----
63 mov ah, 4ch
64 mov al, [exCode]
65 int 21h
66 ;-----
```

Процедура `set_birthday` за допомогою базово-індексної адресації перезаписує елементи масивів. Команда `LEA` завантажує в `BX` початок масиву, `ADD` додає значення у `BX` для майбутнього переміщення на потрібний рядок.

```

69  PROC set_birthday
70
71      LEA BX, array2Db ; записуємо в bx посилання на початок масиву
72      add BX, 64 ; зміщуємося на 3 строку масиву
73      mov[bx], '1' ; додаємо цифри дня народження
74      mov[bx+2], '8'
75      mov[bx+4], '1'
76      mov[bx+6], '1'
77      mov[bx+8], '2'
78      mov[bx+10], '0'
79      mov[bx+12], '0'
80      mov[bx+14], '2'
81
82      add BX, 220h ; зміщуємося на 3 строку першої копії
83      mov[bx], '0' ; додаємо цифри дня народження
84      mov[bx+2], '5'
85      mov[bx+4], '0'
86      mov[bx+6], '7'
87      mov[bx+8], '2'
88      mov[bx+10], '0'
89      mov[bx+12], '0'
90      mov[bx+14], '3'
91
92      add BX, 240h ; зміщуємося на 3 строку другої копії
93      mov[bx], '1' ; додаємо цифри дня народження
94      mov[bx+2], '4'
95      mov[bx+4], '0'
96      mov[bx+6], '2'
97      mov[bx+8], '2'
98      mov[bx+10], '0'
99      mov[bx+12], '0'
100     mov[bx+14], '3'
101     ret
102  ENDP set_birthday

```

Процедура сору використовує цикл та базово-індексну адресацію для копіювання масиву. Задане раніше значення SI і додане нове перезаписують масив у нове місце, не накладаючи один на одиний.

Процедура to_stack також використовує команду LEA, цикл, індексну адресацію. Також використовується команда push, що забезпечує розміщення у сегменті стеку вихідного масиву.

```

104 PROC copy
105     mov cx, 256 ; 16*16 - розмір масиву
106     my_vaaptr:
107         mov bx, [ds:si] ; записуємо у bx значення
108         mov [ds:[si+220h]], bx; записуємо bx до зміщеного на 220h місця у сегменті даних
109         add si, 2 ; переходимо до наступного елемента масиву
110         loop my_vaaptr
111
112     ret
113 ENDP copy
114
115 PROC to_stack
116     lea si, [array2Db] ; записуємо у si початок масиву
117     mov ax, 16 ; записуємо до cx 16 - кількість повторів циклу
118     mov cx, ax
119     stack1:
120     mov ax, [si] ; записуємо до ax елемент масиву
121     add si, 2 ; додаємо 2 для обробки наступного елемента
122     push ax ; додаємо елемент до стеку
123     loop stack1
124
125     ret
126 ENDP to_stack
127
128 end Start

```

Для зручності при запуску був використаний бат-файл:

```

1 set argpath=%lab4
2 set arg1=%lab4test
3 cd %argpath%
4 f:\tasm %arg1%.asm
5 f:\tlink %arg1%.obj
6 %arg1%.exe
7 cd ..

```

Для відладки програми запускається Turbo Debugger.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[]-CPU 80486

Address	Instruction	Comment	Register/Value	Register/Value
cs:0000	B8FE48	mov	ax,48FE	ax 0000
cs:0003	8ED8	mov	ds,ax	bx 0000
cs:0005	8EC0	mov	es,ax	cx 0000
cs:0007	33F6	xor	si,si	dx 0000
cs:0009	E8A100	call	00AD	si 0000
cs:000C	83C640	add	si,0040	di 0000
cs:000F	E89B00	call	00AD	bp 0000
cs:0012	83C640	add	si,0040	sp 4000
cs:0015	E89500	call	00AD	ds 48E1
cs:0018	E80E00	call	0029	es 48E1
cs:001B	E89E00	call	00BC	ss 491F
cs:001E	B401	mov	ah,01	cs 48F1
cs:0020	CD21	int	21	ip 0000
cs:0022	B44C	mov	ah,4C	
cs:0024	A00002	mov	al,[0200]	

ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω

ds:0008 AD DE E4 01 CB 15 AE 01 i 20 15 <0

ds:0010 CB 15 80 02 26 10 94 01 15 80 02 26

ds:0018 01 01 01 00 02 FF FF FF 01 01 01 00

ds:0020 FF FF FF FF FF FF FF

ss:4002 0000

ss:4000 0000

ss:3FFE FFFF

ss:3FFC 0000

ss:3FFA 0000

Dump пам'яті до роботи:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[]-Dump

ds:0000 CD 20 FF 9F 00 EA FF FF AD DE E4 01 CB 15 AE 01 = f Ω i 20 15 <0

ds:0010 CB 15 80 02 26 10 94 01 01 01 01 00 02 FF FF FF 15 80 02 26

ds:0020 FF FF FF FF FF FF FF FF FF FF FF D5 48 94 F6 15 80 02 26

ds:0030 1B 21 14 00 18 00 E1 48 FF FF FF FF 00 00 00 00 < 1 1 1 1 BH

ds:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

ds:0050 CD 21 CB 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

ds:0060 00

ds:0070 00

ds:0080 00 0D 00 FF 4E F7 61 22 A4 01 CA F7 01 80 FF FF 1 1 1 1 1 1 1 1

ds:0090 CA F7 FF FF 00 00 64 F7 01 02 A7 32 CA F7 01 80 1 1 1 1 1 1 1 1

ds:00A0 72 F7 28 06 18 00 14 00 6E F7 1C 21 A4 01 8C AD r 1 1 1 1 1 1 1 1

ds:00B0 14 00 00 00 8C AD 7E F7 2A 00 B5 08 8C AD 00 00 1 1 1 1 1 1 1 1

ds:00C0 14 00 56 AB 70 AD 8C F7 8B 00 46 12 14 00 56 AB 1 1 1 1 1 1 1 1

ds:00D0 70 AD 01 00 9C F7 69 01 9D 0B 02 00 01 00 70 AD p 1 1 1 1 1 1 1 1

ds:00E0 29 54 56 AB 2A F8 C0 02 5F 38 56 AB 70 AD 29 54 1 1 1 1 1 1 1 1

ds:00F0 00 00 46 3A 5C 54 44 48 45 4C 50 2E 54 44 48 00 F:TDHELP.TDH

ds:0100 B8 FE 48 8E D8 8E C0 33 F6 E8 A1 00 83 C6 40 E8 1 1 1 1 1 1 1 1

ds:0110 9B 00 83 C6 40 E8 95 00 E8 0E 00 E8 9E 00 B4 01 1 1 1 1 1 1 1 1

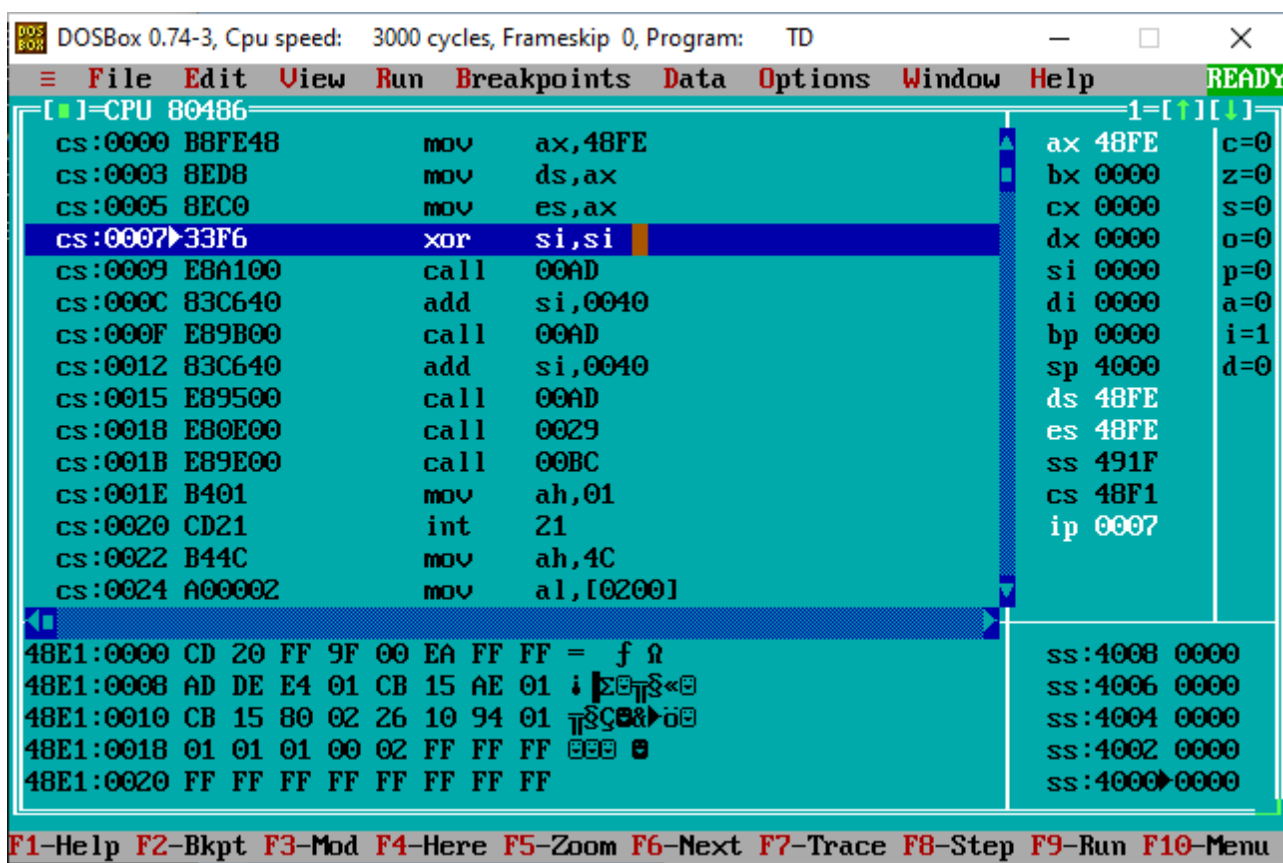
ds:0120 CD 21 B4 4C A0 00 02 CD 21 BB 00 00 83 C3 40 C7 = 1 1 1 1 1 1 1 1

ds:0130 07 31 00 C7 47 02 38 00 C7 47 04 31 00 C7 47 06 1 1 1 1 1 1 1 1

ds:0140 31 00 C7 47 08 32 00 C7 47 0A 30 00 C7 47 0C 30 1 1 1 1 1 1 1 1

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Після початку роботи можна побачити у DS:0000 початковий масив.



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[CPU 80486]

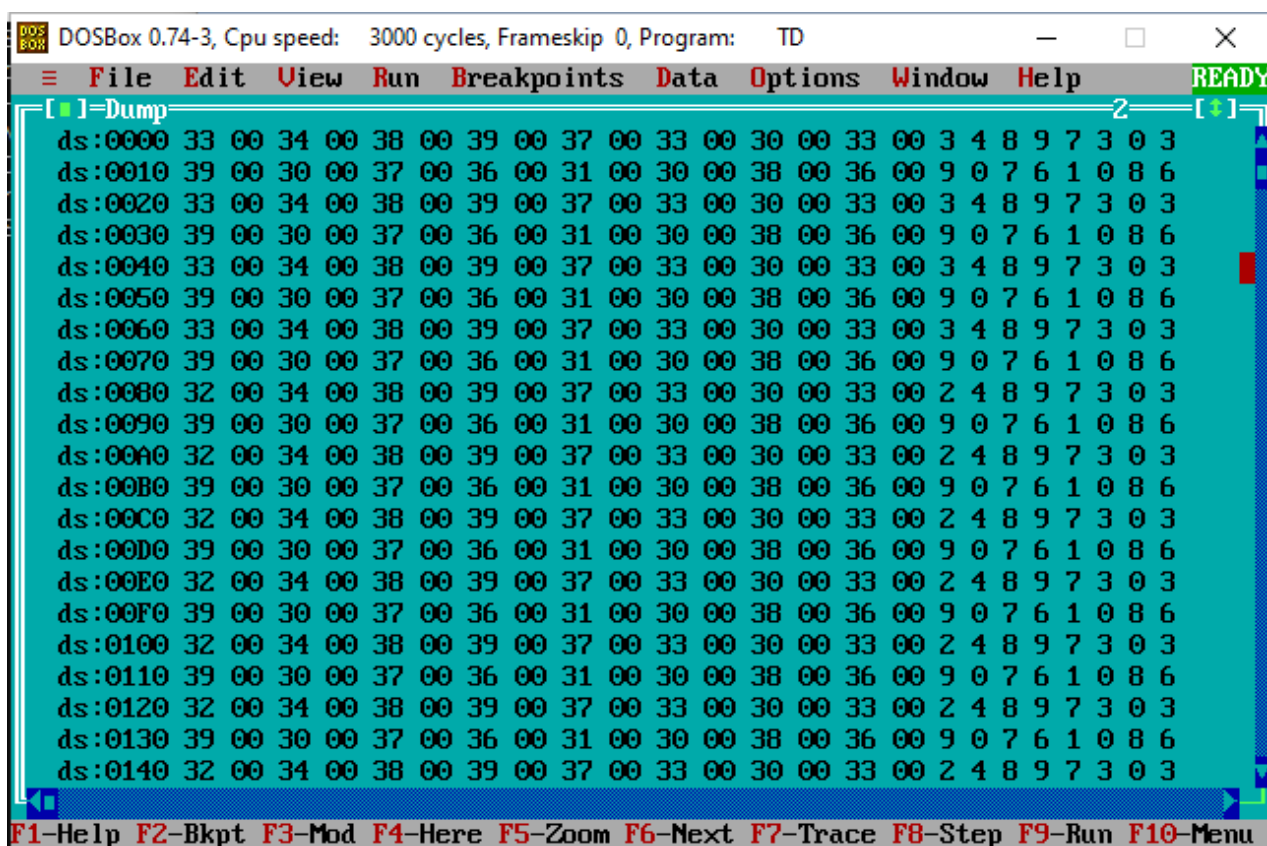
Address	Instruction	Register/Value
cs:0000	B8FE48	mov ax,48FE
cs:0003	8ED8	mov ds,ax
cs:0005	8EC0	mov es,ax
cs:0007	33F6	xor si,si
cs:0009	E8A100	call 00AD
cs:000C	83C640	add si,0040
cs:000F	E89B00	call 00AD
cs:0012	83C640	add si,0040
cs:0015	E89500	call 00AD
cs:0018	E80E00	call 0029
cs:001B	E89E00	call 00BC
cs:001E	B401	mov ah,01
cs:0020	CD21	int 21
cs:0022	B44C	mov ah,4C
cs:0024	A00002	mov al,[0200]

ax 48FE
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 4000
ds 48FE
es 48FE
ss 491F
cs 48F1
ip 0007

48E1:0000 CD 20 FF 9F 00 EA FF FF = f 9
48E1:0008 AD DE E4 01 CB 15 AE 01 : 2 3 4 5 6 7 8 9
48E1:0010 CB 15 80 02 26 10 94 01 : 2 3 4 5 6 7 8 9
48E1:0018 01 01 01 00 02 FF FF FF : 2 3 4 5 6 7 8 9
48E1:0020 FF FF FF FF FF FF FF FF

ss:4008 0000
ss:4006 0000
ss:4004 0000
ss:4002 0000
ss:4000 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[Dump]

Address	Hex Data
ds:0000	33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0010	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0020	33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0030	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0040	33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0050	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0060	33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0070	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0080	32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0090	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:00A0	32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:00B0	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:00C0	32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:00D0	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:00E0	32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:00F0	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0100	32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0110	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0120	32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0130	39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0140	32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Створені копії масивів за адресами, що впливають із написаних даних у коді.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[F2]=Dump 2 [F4]

ds:0220	33	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	3	4	8	9	7	3	0	3
ds:0230	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0240	33	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	3	4	8	9	7	3	0	3
ds:0250	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0260	30	00	35	00	30	00	37	00	32	00	30	00	30	00	33	00	0	5	0	7	2	0	0	3
ds:0270	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0280	33	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	3	4	8	9	7	3	0	3
ds:0290	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:02A0	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:02B0	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:02C0	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:02D0	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:02E0	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:02F0	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0300	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0310	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0320	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0330	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0340	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0350	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0360	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[F2]=Dump 2 [F4]

ds:0460	33	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	3	4	8	9	7	3	0	3
ds:0470	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0480	33	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	3	4	8	9	7	3	0	3
ds:0490	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:04A0	31	00	34	00	31	00	31	00	32	00	30	00	30	00	32	00	1	4	1	1	2	0	0	2
ds:04B0	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:04C0	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:04D0	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:04E0	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:04F0	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0500	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0510	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0520	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0530	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0540	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0550	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0560	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0570	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:0580	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3
ds:0590	39	00	30	00	37	00	36	00	31	00	30	00	38	00	36	00	9	0	7	6	1	0	8	6
ds:05A0	32	00	34	00	38	00	39	00	37	00	33	00	30	00	33	00	2	4	8	9	7	3	0	3

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[J]=Dump

```

ds:06A0 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:06B0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:06C0 30 00 35 00 30 00 37 00 32 00 30 00 30 00 33 00 0 5 0 7 2 0 0 3
ds:06D0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:06E0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:06F0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0700 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0710 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0720 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0730 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0740 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0750 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0760 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0770 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0780 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0790 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:07A0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:07B0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:07C0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:07D0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:07E0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3

```

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Виконання процедури set_birthday:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[J]=Dump

```

ds:0000 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0010 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0020 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0030 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0040 31 00 38 00 31 00 31 00 32 00 30 00 30 00 32 00 1 8 1 1 2 0 0 2
ds:0050 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0060 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0070 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0080 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0090 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:00A0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:00B0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:00C0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:00D0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:00E0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:00F0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0100 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0110 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0120 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0130 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0140 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3

```

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[J]=Dump

```
ds:0220 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0230 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0240 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0250 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0260 30 00 35 00 30 00 37 00 32 00 30 00 30 00 33 00 0 5 0 7 2 0 0 3
ds:0270 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0280 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0290 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:02A0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:02B0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:02C0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:02D0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:02E0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:02F0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0300 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0310 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0320 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0330 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0340 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0350 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0360 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
```

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[J]=Dump

```
ds:0460 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0470 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0480 33 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 3 4 8 9 7 3 0 3
ds:0490 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:04A0 31 00 34 00 30 00 32 00 32 00 30 00 30 00 33 00 1 4 0 2 2 0 0 3
ds:04B0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:04C0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:04D0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:04E0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:04F0 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0500 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0510 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0520 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0530 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0540 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0550 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0560 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0570 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:0580 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
ds:0590 39 00 30 00 37 00 36 00 31 00 30 00 38 00 36 00 9 0 7 6 1 0 8 6
ds:05A0 32 00 34 00 38 00 39 00 37 00 33 00 30 00 33 00 2 4 8 9 7 3 0 3
```

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Висновки: у цій лабораторній роботі були набуті впевнені знання і навички з розробки ПЗ на Асемблері, у ході якої застосовуються знання архітектури комп'ютерів.