

Міністерство освіти і науки України
Національний університет „Львівська політехніка”
Кафедра “Спеціалізованих комп’ютерних систем”



Звіт
до лабораторної роботи № 2
з дисципліни
Системне програмування

Дослідження способів представлення даних в пам’яті комп’ютера з
архітектурою x86.

Варіант: 19

Виконав:

ст. гр. КІ-207

Шаповал Віталій

Перевірив:

Асистент катедри ЕОМ

Максимів М. Р.

Мета: вивчити способи задання констант та змінних в Асемблері, набути навички інтерпретування даних в пам'яті комп'ютера з архітектурою x86.

Завдання:

1. Створити *.exe програму, яка розміщує в пам'яті даних комп'ютера, операнди, що задані варіантом. Вхідні операнди A, B, C, D, E, F з індексом u вважати без знаковими і довжиною в байтах, згідно з індексу, з індексом fs вважати з рухомою комою одинарної точності (32 біти), з індексом fd вважати з рухомою комою подвійної точності (64 біти), з індексом fe вважати з рухомою комою розширеної точності (80 біт); крім цього операнд A є масивом з 3-ох елементів. При оголошенні призначити операндам початкові значення використовуючи всі можливі системи лічби. K – константа, довжина якої визначається значенням(згідно варіанту), а значення задане в шістнадцятковому форматі. Для її опису слід використати директиву EQU. Задати одну мітку в довільному місці сегменту даних. Задати в сегменті даних змінну Message db 'Прізвище',13,10, , де 'Прізвище' – прізвище виконавця роботи, яке вивести на екран.
2. За допомогою меню Debug середовища Visual Studio 2019, дослідити представлення даних в пам'яті комп'ютера (продемонструвати розміщення даних та здійснити інтерпретацію).
3. Скласти звіт про виконану роботу з приведенням тексту програми з коментарями, дампу пам'яті та аналітично інтерпретувати дані для кожної з змінних.
4. Дати відповідь на контрольні запитання.

Мій варіант 19, отже я маю ініціалізувати наступні числа:

$A_4, B_{3u}, C_4, D_{fd}, E_{10}, F_8, K = 4019$

Виконання:

Код:

```
.686  
.model flat, stdcall  
option casemap:none  
include \masm32\include\windows.inc  
include \masm32\include\kernel32.inc  
includelib \masm32\lib\kernel32.lib  
  
.data
```

```

A dd 5d0h, 2720q, 10111010000b ;A4
B dd 00A5h, 0AB3Ch ; B3u
Cc dd 555h ;C4
LBL LABEL BYTE
D dq 3.14e8 ;D fd
E dt 123456789ABCDEh ;10
F dq 12345678h ;8
K equ 4019d
Message db 'Shapoval',13,10
NumberOfCharsToWrite dd $-Message
hConsoleOutput dd 0
NumberOfCharsWritten dd 0

```

```

.code
start:
push -11
call GetStdHandle
mov hConsoleOutput, eax
push 0
push offset NumberOfCharsWritten
push NumberOfCharsToWrite
push offset Message
push hConsoleOutput
call WriteConsoleA
push 0
call ExitProcess
end start

```

Скрини:



Скрин 1. Перегляд вмісту регістру EIP

0x003A3FFB	00 00 00 00 00 d0 05 00 00 d0 05 00 00 d0 05 00 00 a5 00 00P...P...P...Г...
0x003A400F	00 3c ab 00 00 55 05 00 00 00 00 00 80 42 b7 b2 41 de bc 9a	.<«..U.....ЂВ·ІАЮјљ
0x003A4023	78 56 34 12 00 00 00 78 56 34 12 00 00 00 53 68 61 70 6f	xV4....xV4.....Shapo
0x003A4037	76 61 6c 0d 0a 0a 00 00 00 00 00 00 00 00 00 00 00 00	val.....

Скрин 2. Перегляд вмісту пам'яті

Аналіз дампу пам'яті:

```

0x003A4000  d0 05 00 00 d0 05 00 00 d0 05 00 00 a5 00 00 00 3c ab 00 00 55 05 00 00 00 00 00
80 42 b7 b2 41 de bc 9a 78 56 34 12 00 00 00 78 56 34
P...P...P...Г...<«..U.....ЂВ·ІАЮјљxV4....xV4
0x003A402D  12 00 00 00 00 53 68 61 70 6f 76 61 6c 0d 0a 0a 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....Shapoval.....

```

.data

A dd 5d0h, 2720q, 10111010000b ;A4

B dd 00A5h, 0AB3Ch ; B3u

Cc dd 555h ;C4

LBL LABEL BYTE

D dq 3.14e8 ;D fd

E dt 123456789ABCDEh ;10

F dq 12345678h ;8

K equ 4019d

Message db 'Shapoval',13,10

Перевірка дробового числа:

$$3.14e8 = 3.14 * 10^8 = 314000000 = 1\ 0010\ 1011\ 0111\ 0100\ 0010\ 1000\ 0000_2 = 1.0010101101110100001010000000 * 2^{28}$$

Знак числа = 0

$$\text{Порядок} = 28 + 1023 = 1051 = 100\ 0001\ 1011_2$$

$$\text{Мантиса} = 0010\ 1011\ 0111\ 0100\ 0010\ 1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$$

$$\text{Внутрішнє представлення числа} = 0100\ 0001\ 1011\ 0010\ 1011\ 0111\ 0100\ 0010\ 1000\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 41\ B2\ B7\ 42\ 80\ 00\ 00\ 00_{16}$$

Висновок: виконавши цю лабораторну роботу я дізнався про способи задання констант та змінних в Асемблері, набув навички інтерпретувати та читати дані в пам'яті комп'ютера з архітектурою x86.