

Міністерство освіти і науки, молоді та спорту України  
Національний університет “Львівська політехніка”

Кафедра ЕОМ



## Звіт

з лабораторної роботи №1 з дисципліни:  
“Системного програмування”  
на тему: «Дослідження способів представлення  
даних в пам’яті комп’ютера з архітектурою x86.»  
Варіант: № 24.

Виконав:  
ст. групи КІ-203  
Ширий Б. І.  
Перевірила:  
стар. вик. кафедри ЕОМ  
Ногаль М. В.

## МЕТА РОБОТИ:

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

## ЗАВДАННЯ:

### УМОВА 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.

### ВИКОНАННЯ:

Ознайомившись із завданням почав працювати над його реалізацією відповідно до моїх даних, які зображені у таблиці 1.

Таблиця 1. Дані 24 варіанту.

Варіант 24	$A_2, B_{2u}, C_{fs}, D_{10u}, E_1, F_8, K$	74569024
------------	---------------------------------------------	----------

Згідно свого завдання написав текст асемблера, який навів у лістингу 2. Текст програми, яка реалізує збереження заданих змінних у пам'яті та виводить на екран моє прізвище, а саме «Shyryi».

Таблиця 2. Текст програми до заданого завдання.

```
.686
.model flat, stdcall
option casemap:none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
includelib \masm32\lib\kernel32.lib
.data
    A dw 1234h, 0710h, 2003h
    B dw 5678h
    Cc dd 10.25
    LBL LABEL BYTE
    D dt 7
    E db 10
    F dq 11111010011b
    K equ 74569024
    ShyryiMessage db 'Shyryi', 13, 10
    NumberOfCharsToWrite dd $-ShyryiMessage
    hConsoleOutput dd 0
    NumberOfCharsWritten dd 0
.code
start:
    push -11
    call GetStdHandle
    mov hConsoleOutput, eax
    push 0
    push offset NumberOfCharsWritten
    push NumberOfCharsToWrite
    push offset ShyryiMessage
    push hConsoleOutput
    call WriteConsoleA
    push 0
    call ExitProcess
end start
```

Скріншот пам'яті наведений на рисунку 1. Відповідний уривок пам'яті скопіював та пояснив детальніше нижче:

0x003D3FFE	00	00	34	12	10	07	03	20	78	56	..4.... xV
0x003D4008	00	00	24	41	07	00	00	00	00	00	..\$A.....
0x003D4012	00	00	00	00	0a	d3	07	00	00	00	.....y....
0x003D401C	00	00	00	53	68	79	72	79	69	0d	...Shyryi.
0x003D4026	0a	08	00	00	00	00	00	00	00	00	.....

Рис. 1. Скріншот дампу пам'яті.

0x003D3FFE	00	00	34	12	10	07	03	20	78	56	..4.... xV
0x003D4008	00	00	24	41	07	00	00	00	00	00	..\$A.....
0x003D4012	00	00	00	00	0a	d3	07	00	00	00	.....y....
0x003D401C	00	00	00	53	68	79	72	79	69	0d	...Shyryi.
0x003D4026	0a	08	00	00	00	00	00	00	00	00	.....

Відповідність між даними та тим, куди вони належать, пояснив табличкою 3.

Таблиця 3. Пояснення взаємозв'язків тексту до області пам'яті.

Змінна	Запис у пам'яті	Стрічка у коді
A	34 12 10 07 03 20	A dw 1234h, 0710h, 2003h
B	78 56	B dw 5678h
Cc	00 00 24 41	Cc dd 10.25
D	07 00 00 00 00 00 00 00 00 00	D dq 7
E	0a	E db 10
F	d3 07 00 00 00 00 00 00	F dq 11111010011b
ShyryiMessage	53 68 79 72 79 69 0d 0a	ShyryiMessage db 'Shyryi', 13, 10
NumberOfCharsToWrite	08 00 00 00	NumberOfCharsToWrite dd \$ ShyryiMessage

Відповідність також можна звірити у відповідному вікні Visual Studio, а саме “Watch”, куди заносив назви змінних для їх відслідковування. Відповідний розділ наведений на рисунку 2.

Name	Value	Type
A	0x1234	unsigned short
B	0x5678	unsigned short
Cc	0x41240000	unsigned long
D	3.458459520889e-323#DEN	long double
E	0x0a '\n'	unsigned char
F	0x000000000000007d3	unsigned __int64
ShyryiMessage	0x53 'S'	unsigned char
'h'	0x68 'h'	char
'y'	0x79 'y'	char
'r'	0x72 'r'	char
'y'	0x79 'y'	char
'i'	0x69 'i'	char
NumberOfCharsToWrite	0x00000008	unsigned long

Рис. 2. Область "Watch".

Очевидно, що шістнадцяткові числа відповідають відповідним записам у дампі. Отож, перевіримо десяткові, двійкові та числа з рухомою комою.

### ЗМІННА З РУХОМОЮ КОМОЮ:

Зробимо обчислення для числа 10.25 у форматі IEEE 754 одинарної точності, обчислення наведено у таблиці :

Таблиця 4. Обчислення числа 10,25.

<b>Запишемо число у двійковій системі числення: 1010.01.</b>
<b>Переведемо його у науковий запис з мантиєю у діапазоні [1,2): 1.01001 * 2^3.</b>
<b>Визначимо знак числа: 0 (додатнє число).</b>

**Визначимо порядок:** 3 (експонента у науковому записі). Додамо до нього зміщення для зберігання порядку у форматі IEEE 754 одинарної точності:  $3 + 127 = 130$ . Переведемо число 130 у двійкову систему числення: 10000010.

**Визначимо мантису.** Запишемо мантису числа у нормалізованому вигляді, тобто без першого значущого біта (який завжди рівний 1 у відповідному форматі з плаваючою комою). Отже, мантиса дорівнює 01001 00000000 00000000.

**Об'єднаємо отримані значення у внутрішнє представлення числа у форматі IEEE 754 одинарної точності:**

знак: 0,

порядок: 10000010,

мантиса: 01001000000000000000000000000000.

У шістнадцятковому вигляді це число записується як 0x41240000.

---

## ДЕСЯТКОВІ ЧИСЛА:

Десятибайтове число має мати 10 розрядів, а у змінну D записано число  $7_{10}$ , тому вона має вигляд 07 00 00 00 00 00 00 00 00 00, адже виділяється місце для 10-ти байтів. Змінна E зберігає однобайтове число  $10_{10}$ , тобто  $A_{16}$ , тому її вигляд 0a.

---

## ДВІЙКОВІ ЧИСЛА:

У змінній F зберігається  $111\ 1101\ 0011_2$ , що буде  $7D3_{16}$ , а оскільки вона восьмибайтова, то її вигляд d3 07 00 00 00 00 00 00.

## ВИСНОВКИ:

Протягом лабораторної роботи, дослідив способи представлення даних в пам'яті комп'ютера з архітектурою x86, а також вивчив способи задання констант та змінних в Асемблері, набути навички інтерпретування даних в пам'яті комп'ютера з архітектурою x86.

## КОНТРОЛЬНІ ПИТАННЯ:

ЯК ВИЗНАЧАЮТЬСЯ В АСЕМБЛЕРІ ДАНІ ДОВЖИНОЮ В 2 БАЙТИ?  
ЯК ВОНИ ЗАПИСУЮТЬСЯ У ПАМ'ЯТЬ?

В Асемблері дані довжиною в 2 байти зазвичай позначаються як слова (word). Для їх визначення можна використовувати директиву DW (Define Word), яка дозволяє задати значення для слова.

Наприклад, для визначення слова зі значенням 1234h можна використати такий код:

```
my_word DW 1234h
```

Цей рядок коду визначає змінну з ім'ям "my\_word", яка містить значення 1234h. При записі в пам'ять вони змінюють свою позицію, що у відповідності до попереднього прикладу буде не 12 34, а 34 12.

## ЧИМ ВІДРІЗНЯЮТЬСЯ ДИРЕКТИВА EQU ВІД DW?

Основна відмінність між директивою EQU і DW полягає в тому, що EQU використовується для визначення символів, які можна використовувати в кодї програми замість числових значень, тоді як DW використовується для визначення двобайтових значень в пам'яті.

## ЯК У ПАМ'ЯТІ ВИЗНАЧАЄТЬСЯ ЧИСЛО З РУХОМОЮ КОМОЮ?

Стандарт зберігання чисел з плаваючою комою визначає, що число зберігається у формі наступних трьох компонентів:

Таблиця 5. Компоненти стандарту зберігання чисел з рухомою комою.

Компонент	Пояснення
Знак (Sign bit)	відповідає за визначення знаку числа і зберігається в одному біті. 0 означає додатне число, 1 - від'ємне.
Мантиса (Mantissa):	це двійкове число з фіксованою кількістю бітів, яке містить значення дробової частини числа з плаваючою комою.
Експонента (Exponent):	це двійкове число з фіксованою кількістю бітів, яке відповідає за збільшення або зменшення значення мантиси.

## ЯК ІНТЕРПРЕТУВАТИ ЧИСЛО З РУХОМОЮ КОМОЮ З РОЗШИРЕНОЮ ТОЧНІСТЮ?

Щоб інтерпретувати число з рухомою комою з розширеною точністю, потрібно виконати дії, які описані у таблиці 6:

Таблиця 6. Інтерпретація числа з рухомою комою з розширеною точністю.

Визначити знак числа	перший біт числа вказує на його знак. Якщо значення першого біта дорівнює 0, число є додатнім, якщо 1 - від'ємним.
Визначити значення експоненти	експонента зазвичай зберігається у форматі «зміщення на постійну величину», тому її значення потрібно знайти, віднявши від збереженого значення 16383. Таким чином, якщо в пам'яті збережене значення експоненти дорівнює 16383, то її значення у дійсному числі дорівнює 0.
Визначити значення мантиси	мантиса складається з 64 бітів і представляє значення дробової частини числа. Зазвичай мантиса містить 1 перед комою, тому значення мантиси можна обчислити за допомогою формули: $1 + (\text{мантиса} / 2^{64})$ .
Обчислити дійсне значення числа	дійсне значення числа можна обчислити за формулою: $(-1)^{\text{знак}} * \text{мантиса} * 2^{(\text{експонента} - \text{зміщення})}$ .