

Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 2

з дисципліни: «Системного програмування. Частина 1»

«Дослідження способів представлення даних в пам'яті комп'ютера з архітектурою x86»

Виконав:

ст.гр. КІ-201

Добош М. І.

Перевірив:

Козак Н. Б.

Львів 2023

Мета: вивчити способи задання констант та змінних в Асемблері, набути навики 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 i 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 operand A є масивом з 3-ох елементів.

При оголошенні

призначити operandам 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

Варіант №8

8	A ₄ ,B _{2u} ,C _{ds} ,D _{10u} ,E ₁ ,F ₈ , K	88
---	--	----

Виконання:

Рис 1. Вміст дампу пам'яті

Дамп пам'яті відображенено на рис.1, де чорним кольором відображена перша змінна (операнд), синім – друга, червоним – третя, жовтим – четверта і т.д. К не заноситься в пам'ять а заміняється числом, що їй відповідає, у тексті програми на етапі розбору коду препроцесором.

```
A dd 0223Q, 0AABb33h, 6125233  
B dw 005533h  
Cc dd 0.5234  
D dt 00abcdfeb7722h  
E db 78  
F dq 011882266bbcch  
Zamkova db 'Zamkova',13,10
```

Перевіримо число з плаваючою комою:

0.5234₁₀

Знак числа: 0

Порядок: $-1 + 127 = 126_{10} \rightarrow 01111110_2$

Мантиса: 1.0000101111110110001010

Внутрішнє представлення числа:

0 01111110 0000101111110110001011-> 3f05fd8b₁₆

Код програми:

```
.686
.model flat, stdcall
option casemap:none
include C:\masm32\include\kernel32.inc
include lib C:\masm32\lib\kernel32.lib
.data
A dd 0223Q, 0AABb33h, 6125233
B dw 005533h
Cc dd 0.5234
LBL LABEL BYTE
D dt 00abcdfebb7722h
E db 78
F dq 011882266bbcch
K equ 88h
Dobosh db 'Dobosh',13,10
NumberOfCharsToWrite dd $-Dobosh
hConsoleOutput dd 0
NumberOfCharsWritten dd 0
.code
start:
mov eax, offset Dobosh
push -11
```

```
call GetStdHandle
mov hConsoleOutput, eax
push 0
push offset NumberOfCharsWritten
push NumberOfCharsToWrite
push offset Dobosh
push hConsoleOutput
call WriteConsoleA
push 0
call ExitProcess
end start
```

Висновок: вивчив способи задання констант та змінних в Асемблері, набув навики інтерпретування даних в пам'яті комп'ютера з архітектурою x86