­­Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Спеціалізованих комп’ютерних систем”



**Звіт**

до лабораторної роботи № 4

з дисципліни

# *​*Системне програмування

# Дослідження роботи команд переходів. Програмування задач з використанням алгоритмів розгалуження.

# **Варіант: 19**

**Виконав:**

ст. гр. КІ-207

Шаповал Віталій

**Перевірив:**

Асистент катедри ЕОМ

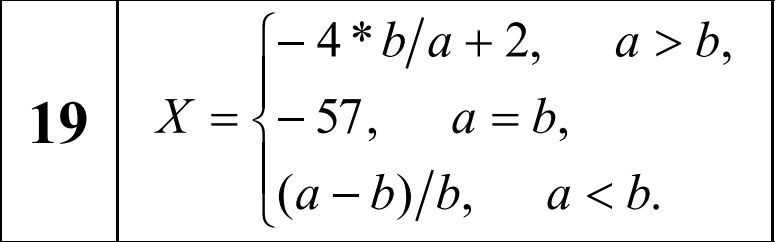
Максимів М. Р.

**Мета:** освоїти використання команд порівняння, умовного та безумовного переходів. Набути вміння використовувати арифметичні команди над знаковими даними та команди логічних операцій.

**Завдання:**

1. Створити \*.exe програму, яка реалізовує обчислення, заданого варіантом виразу. Вхідні дані слід вважати цілими числами зі знаком, розміром один байт. Результат обчислення виразу повинен записуватися у пам’ять. Уникнути випадку некоректних обчислень при діленні на нуль та при переповненні розрядної сітки (вивести відповідне текстове повідомлення).
2. За допомогою Debug, відслідкувати правильність виконання програми (продемонструвати результати проміжних та кінцевих обчислень) та проаналізувати отримані результати для різних вхідних даних.
3. Скласти звіт про виконану роботу з приведенням тексту програми та коментарів до неї.
4. Дати відповідь на контрольні запитання.

**Варіант: 19**

****

**Виконання:**

**Код:**

.686

.model flat, stdcall

option casemap:none

include \masm32\include\windows.inc

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

A   db  -50

B   db  -10

X   dw  0

Hello   db 13, 10

        db '  X = (a-b)/b    if a < b', 13, 10

        db '  X = -57        if a = b', 13, 10

        db '  X = -4b/a      if a > b', 13, 10

Operands    db 13, 10, '  A =      B =      ', 13, 10

NumberOfCharsToWrite\_Hello dd $-Hello

Error       db 13, 10, '  Error - divide by zero!', 13, 10

NumberOfCharsToWrite\_Error dd $-Error

Result      db 13, 10, '  X =           ', 13, 10

NumberOfCharsToWrite\_Result dd $-Result

format db '%hd', 0

hConsoleOutput dd 0

NumberOfCharsWritten dd 0

.code

start:

;Вивід повідомлення Hello

mov al, A

cbw

push ax

push offset format

push offset [Operands+8]

call wsprintfA

mov al, B

cbw

push ax

push offset format

push offset [Operands+17]

call wsprintfA

push -11

call GetStdHandle

mov hConsoleOutput, eax

push 0

push offset NumberOfCharsWritten

push NumberOfCharsToWrite\_Hello

push offset Hello

push hConsoleOutput

call WriteConsoleA

;перевірка на рівніст A та B

mov al, A

cmp al, B

jne A\_ne\_B

mov X, -57

jmp Output\_Result

A\_ne\_B:

jg A\_g\_B

cmp B, 0

je Output\_Error

;Обчислення Х при a<b

mov al, A

sub al, B

cbw

idiv B

mov X, ax

jmp Output\_Result

A\_g\_B:

cmp A, 0

je Output\_Error

;обчислення Х при a>b

mov al, B

cbw

mov bx, ax

mov ax, -4

imul ax, bx

idiv A

mov X, ax

jmp Output\_Result

;вивід результату

Output\_Result:

push X

push offset format

push offset [Result+8]

call wsprintfA

push offset NumberOfCharsWritten

push NumberOfCharsToWrite\_Result

push offset Result

push hConsoleOutput

call WriteConsoleA

jmp exit

;вивід повідомлення про ділення на 0

Output\_Error:

push offset NumberOfCharsWritten

push NumberOfCharsToWrite\_Error

push offset Error

push hConsoleOutput

call WriteConsoleA

jmp exit

;вихід з програми

exit:

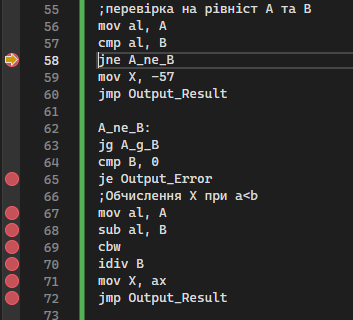
push 0

call ExitProcess

end start

**Результат виконання:**

**Дослідження правильности виконання програми для A < B**



Скрин 1. Розставлені брекпойнти

Після виконання cmp вміст регістру EFL змінився на

EFL = 00000287 = 0000 0000 0000 0000 0000 0010 1000 0111,

jne перевіряє прапорець ZF, якому відповідає 6-ий біт EFL, який рівний 0, отже A і B – нерівні, тому jne переміщає нас до A\_ne\_B

jg перевіряє прапорець SF, якому відповідає 7-ий біт EFL, який рівний 1, отже прапорець знаходиться в стані NG, тому A < B, і jg ігнорується.

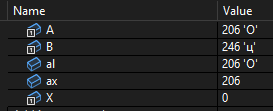
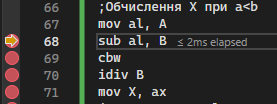


Скрин 2. Перевірка на B == 0

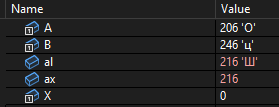
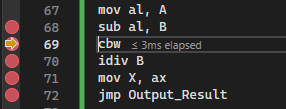
Після виконання cmp B, 0 вміст регістру EFL змінився на

EFL = 00000286 = 0000 0000 0000 0000 0000 0010 1000 0110,

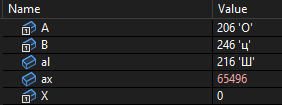
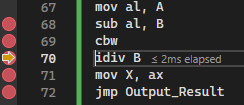
je перевіряє прапорець ZF, який рівний 0, отже B != 0, тому je ігнорується.



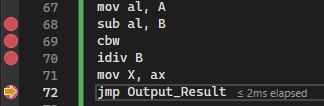
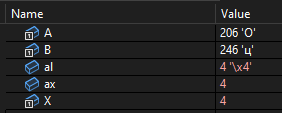
Скрин 3. Поміщення А в регістр al



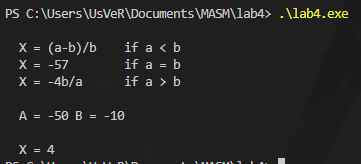
Скрин 4. A – B



Скрин 5. Розширення al до ax

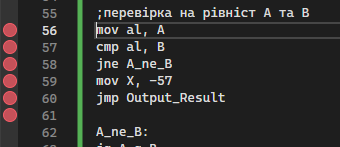
****

Скрин 6. A / B та поміщення результату з регістру ax у змінну X

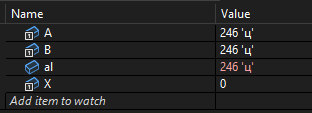
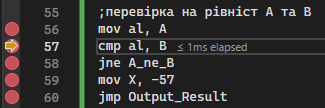
****

Скрин 7. Результат виконання програми

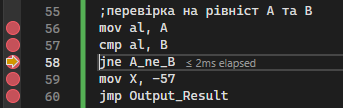
**Перевірка павильности виконання для A == B**

****

Скрин 8. Розставлені брекпойнти

****

Скрин 9. Переміщення A до регістру al

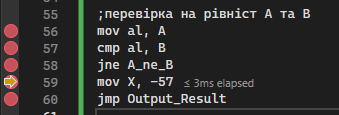


Скрин 10. A – B та внесення результату до регістру прапорців, без зміни змінних

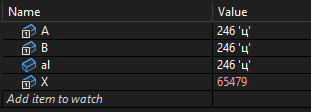
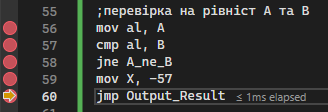
Після виконання директиви cmp al, B вміст регістру EFL змінився на:

EFL = 00000246 = 0000 0000 0000 0000 0000 0010 0100 0110

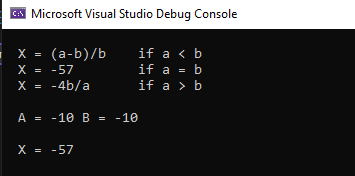
jne перевіряє прапорець ZF, якому відповідає 6-ий біт EFL, який рівний 1, отже A і B – рівні, тому jne не переміщає нас до A\_ne\_B:



Скрин 11. Jne не спрацьовує

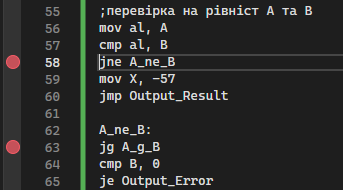


Скрин 12. Поміщення до результату X константи -57

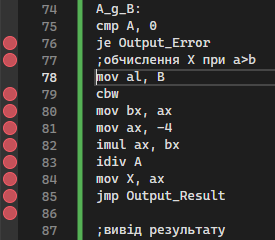
****

Скрин 13. Результат виконання програми

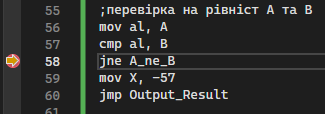
**Перевірка правильности виконання програми для A > B:**



Скрин 14. Розставлені брекпойнти



Скрин 15. Розставлені брекпойнти

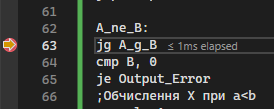


Скрин 16. A-B та занесення результату в регістр, без впливу на змінні

Після виконання cmp вміст регістру EFL змінився на

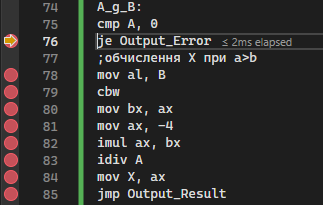
EFL = 00000217 = 0000 0000 0000 0000 0000 0010 0001 0111

jne перевіряє прапорець ZF, якому відповідає 6-ий біт EFL, який рівний 0, отже A і B – нерівні, тому jne переміщає нас до A\_ne\_B



Скрин 17. Jne спрацьовує та переміщає до A\_ne\_B

jg перевіряє прапорець SF, якому відповідає 7-ий біт EFL, який рівний 0, отже прапорець знаходиться в стані NG, тому A > B, і jg перекидає програму до A\_g\_B.

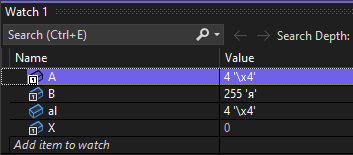
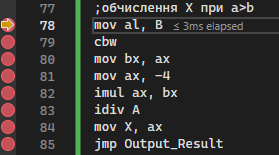


Скрин 18. Jg спрацьовує та переміщає до A\_g\_B

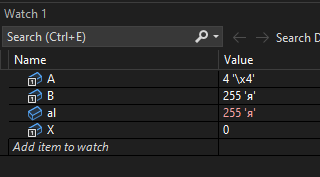
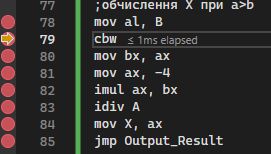
Після виконання cmp A, 0 вміст регістру EFL змінився на

EFL = 00000202 = 0000 0000 0000 0000 0000 0010 0000 0010,

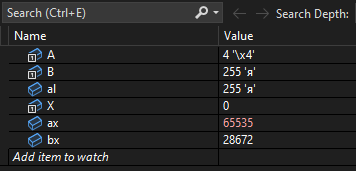
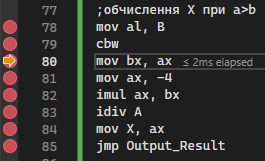
je перевіряє прапорець ZF, який рівний 0, отже A != 0, тому je ігнорується.



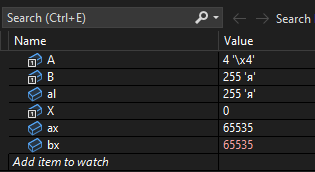
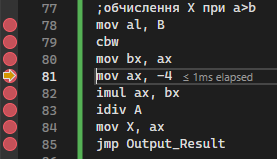
Скрин 19. Je не спрацьовує, програма продовжує послідовне виконання



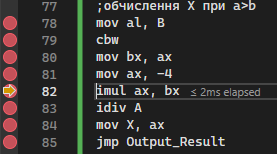
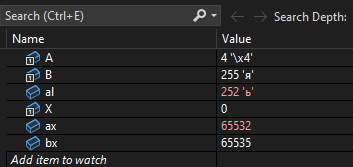
Скрин 20. Поміщення B в регістр al



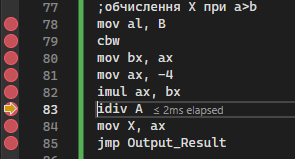
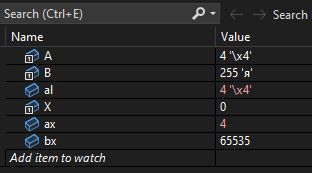
Скрин 21. Розширення al до ax



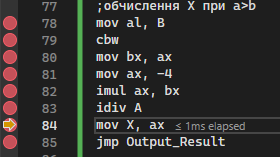
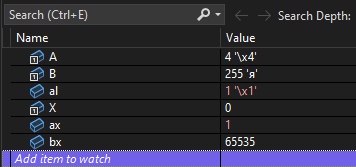
Скрин 22. Переміщення вмісту ax до bx

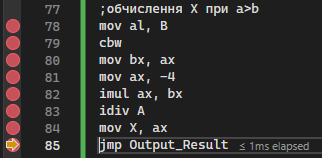
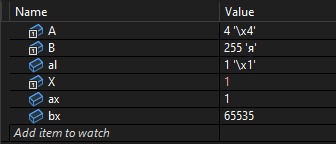
Скрин 23. Поміщення константи -4 до ax

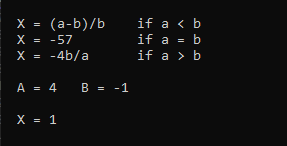
Скрин 24. -4 \* B

Скрин 25. Вміст ax ділиться на A

Скрин 26. Переміщення вмісту ax до X



Скрин 27. Результат виконання програми

**Висновок:** виконавши цю лабораторну роботу я освоїв використання команд порівняння, умовного та безумовного переходів та набув вміння використовувати арифметичні команди над знаковими даними та команди логічних операцій.