

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

19	$X = \begin{cases} -4 * b/a + 2, & a > b, \\ -57, & a = b, \\ (a - b)/b, & a < b. \end{cases}$
-----------	--

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 рівність 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 X 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 X 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:

```
PS C:\Users\UsVeR\Documents\MASM\lab4> .\lab4.exe

X = (a-b)/b    if a < b
X = -57        if a = b
X = -4b/a      if a > b

A = -50 B = -10

X = 4
```

Скрин 1. Результат виконання програми

Дослідження правильності виконання програми

Для $A < B$

```
55 ;перевірка на рівність A та B
56 mov al, A
57 cmp al, B
58 jne A_ne_B
59 mov X, -57
60 jmp Output_Result
61
62 A_ne_B:
63 jg A_g_B
64 cmp B, 0
65 je Output_Error
66 ;Обчислення X при a<b
67 mov al, A
68 sub al, B
69 cbw
70 idiv B
71 mov X, ax
72 jmp Output_Result
73
```

Скрин 2. 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.

```
64 cmp B, 0
65 je Output_Error ≤ 4ms elapsed
66 ;Обчислення X при a<b
```

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 $B \neq 0$, тому `je` ігнорується.

66	;Обчислення X при a<b	Name	Value
67	mov al, A	A	206 'O'
68	sub al, B ≤ 2ms elapsed	B	246 'ц'
69	cbw	al	206 'O'
70	idiv B	ax	206
71	mov X, ax	X	0

67	mov al, A	Name	Value
68	sub al, B	A	206 'O'
69	cbw ≤ 3ms elapsed	B	246 'ц'
70	idiv B	al	216 'Ш'
71	mov X, ax	ax	216
72	jmp Output_Result	X	0

67	mov al, A	Name	Value
68	sub al, B	A	206 'O'
69	cbw	B	246 'ц'
70	idiv B ≤ 2ms elapsed	al	216 'Ш'
71	mov X, ax	ax	65496
72	jmp Output_Result	X	0

67	mov al, A	Name	Value
68	sub al, B	A	206 'O'
69	cbw	B	246 'ц'
70	idiv B	al	4 '\x4'
71	mov X, ax	ax	4
72	jmp Output_Result ≤ 2ms	X	4

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