

Міністерство освіти та науки України
Національний технічний університет України “Київський політехнічний
інститут”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №6
з дисципліни «Системне програмування – 1»
на тему: «Програмування побітових операцій»

Виконав:
студент 2-го курсу ФІОТ
групи ІВ-71
Мазан Я. В.
Перевірив:
Старший викладач
Порєв В. М.

Мета:

Навчитися програмувати на асемблері побітові операції, вивчити основні команди обробки бітів.

Варіант завдання:

№ варіанту	Операція	Розрядність (біт)
9	Обчислення кількості старших одиниць у двійковому коді	512

Код програми:

main.asm:

```
%include "io.inc"
%include "MACRO_LONG.asm"
%include "longop.asm"

section .bss
elderOnesCounter : resd 2
numberToProcess : resd 64
section .data
inscription db "Лабораторна робота №6 Виконав Мазан Ян, група ІВ-71", 13, 13, 0
numberInscription db "Число, у якому порахуємо кількість старших одиниць: ", 0
resultInscription db "Кількість старших одиниць у двійковому коді: ", 0
section .text
global CMAIN
CMAIN:
mov ebp, esp
PRINT_STRING inscription
NEWLINE
mov ecx, 64
mov edx, 0
dec ecx
mov bl, 0b11111111
@initLoop:
mov byte [numberToProcess + edx], bl
inc edx
dec bl
dec ecx
cmp ecx, 0
jge @initLoop
PRINT_STRING numberInscription
PRINT_BYTES_HEX 64, numberToProcess
NEWLINE
push 64
push numberToProcess
push elderOnesCounter
call ElderOnesCount
NEWLINE
PRINT_STRING resultInscription
PRINT_UDEC 2, elderOnesCounter
ret
```

longop.asm: (процедури для моєї лабораторної лише)

```
SHL_1bit_LONG: ; array size, array
push ebp
mov ebp, esp
mov eax, dword [ebp + 12] ; size
mov ebx, dword [ebp + 8] ; array
```

```

and eax, 0xff
xor dl, dl
clc
@last_byte_shift: ; shifting last byte without adding anything
mov cl, byte [ebx + eax - 1]
shl cl, 1
mov dl, 0
adc dl, 0
mov byte [ebx + eax - 1], cl
dec eax ; make pointer on the end of array
@shiftCycle:
mov cl, byte [ebx + eax - 1]
shl cl, 1
pushf
mov byte [ebx + eax - 1], cl
add byte [ebx + eax - 1], dl
popf
mov dl, 0
adc dl, 0
dec eax
cmp eax, 0
jnz @shiftCycle
;xor cl, cl
leave
ret 8

```

SHL_Nbit_LONG: ; size, array, shifts number

```

push ebp
mov ebp, esp
mov eax, dword [ebp + 16] ; array size
mov ebx, dword [ebp + 12] ; array
mov edx, dword [ebp + 8] ; shifts number
@longShiftCycle:
push edx
mov eax, dword [ebp + 16] ; array size
mov ebx, dword [ebp + 12] ; array

```

```

push eax
push ebx
call SHL_1bit_LONG
pop edx
dec edx
cmp edx, 0
jnz @longShiftCycle
leave
ret 12

```

ElderOnesCount: ; array's size, array, counter's output

```

push ebp
mov ebp, esp
mov eax, dword [ebp + 16] ; size
mov ebx, dword [ebp + 12] ; array
mov edi, dword [ebp + 8] ; counter's output
@countingLoop:
mov cl, byte [ebx]
and cl, 0x80
cmp cl, 0
je @endp
@incCounter:
inc dword [edi]
@shiftInCounterCycle:
mov eax, dword [ebp + 16] ; array size
mov ebx, dword [ebp + 12] ; array
push eax
push ebx
call SHL_1bit_LONG
jmp @countingLoop

```

```
@endp:
leave
ret 12
```

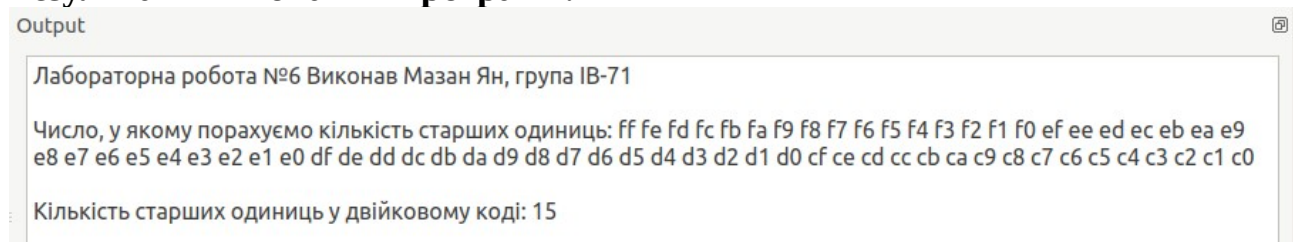
MACRO_LONG.asm:

```
%macro PRINT_BYTES_HEX 2 ; length, byte long_number
mov ebp, %1
mov ecx, 0
%%print:
dec ebp
;mov dl, byte [%2 + ecx]
;clc
;sbb dl, 0x0f
;cmp dl, 0
;jle %%elderDigitZero
;PRINT_HEX 1, [%2 + ecx]
;PRINT_STRING " "
;jmp %%continue
;%%elderDigitZero:
; PRINT_HEX 1, 0
PRINT_HEX 1, [%2 + ecx]
PRINT_STRING " "
;%%continue:
inc ecx
cmp ebp, 0
jnz %%print
%endmacro
```

```
%macro PRINT_DWORD_HEX 2 ; length, dword long_number
mov ebp, %1
mov ecx, 0
%%print:
dec ebp
PRINT_HEX 4, [%2 + ecx*4]
PRINT_STRING " "
inc ecx
cmp ebp, 0
jnz %%print
%endmacro
```

```
%macro shl_1bit_LONG 2
push %1
push %2
call SHL_1bit_LONG
%endmacro
```

Результати виконання програми:



Висновок:

Під час виконання лабораторної роботи я закріпив навички програмування на Асемблері, а саме: створення процедур, передача параметрів за допомогою стеку та регістрів, підключення модулів, механізм циклів. Також я навчився програмувати побітові операції для чисел з підвищеною розрядністю та

використовувати логічний зсув замість операцій ділення або множення на степені числа 2.