Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №3**

**з дисципліни «Архітектура комп’ютера 3»**

Виконав:

студент ІІI курсу

групи ІО-71

Петрик А. С.

Залікова книжка: 7120

Перевірив:

ас. Каплунов А. В.

Київ 2020 р.

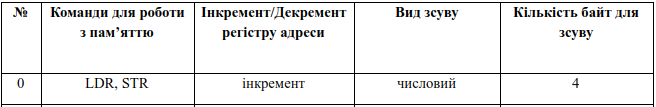
**Тема: з**авантажувач основної програми. Обробка виключень. Вивід даних на відлагоджувальний порт або консоль.

**Мета:** навчитися працювати з оперативною пам’яттю, використовувати інструкції спеціального призначення, використовувати виключення процесора Cortex-M4. Створення мінімального завантажувача системи. Навчитися користуватися виводом даних через відлагоджувальний порт (або консоль).

**Посилання на Git репозиторій:** **https://github.com/Voorhees2019/AK3/tree/master/Lab3**

**Варіант: 7120 % 16 = 0**

**Завдання:**



**Завдання з минулої лабораторної: (a+b)/2 + c!**

**Хід роботи**

**kernel.S**

.syntax unified

.cpu cortex-m4

.thumb

// Global memory locations.

.global vtable\_kernel

.global \_\_kernel\_reset\_\_

.type vtable\_kernel, %object

.type \_\_kernel\_reset\_\_, %function

.section .interrupt\_vector

vtable\_kernel:

.word \_\_stack\_start

.word \_\_kernel\_reset\_\_+1

.size vtable\_kernel, .-vtable\_kernel

.section .rodata

data: .asciz "kernel started!\n"

final: .asciz "Value in register #4: "

.section .text

\_\_kernel\_reset\_\_:

ldr r0, =data

bl dbgput\_line

// calculate

mov r0, #6 // a

mov r1, #4 // b

mov r2, #3 // c

add r4, r0, r1 // a+b

lsr r4, r4, #1 // (a+b)/2

mov r3, #1

factorial:

cmp r2, #1

beq sum

mul r3, r2

sub r2, #1

b factorial

sum:

add r4, r3

ldr r0, =final

bl dbgput

mov r0, r4

bl dbgput\_num

end:

b end

**bootloader.S**

.syntax unified

.cpu cortex-m4

//.fpu softvfp

.thumb

.global bootload

.section .rodata

image: .incbin "kernel.bin"

end\_of\_image:

str\_boot\_start: .asciz "bootloader started"

str\_boot\_end: .asciz "bootloader end"

str\_boot\_indicate: .asciz "#"

.section .text

bootload:

ldr r0, =str\_boot\_start

bl dbgput\_line

ldr r0, =end\_of\_image

ldr r1, =image

ldr r2, =\_ram\_start

loop:

ldr r3, [r1], #4

str r3, [r2], #4

cmp r0, r1

bhi loop

bl newline

ldr r0, =str\_boot\_end

bl dbgput\_line

ldr lr, =bootload\_end

add lr, #1

ldr r2, =\_ram\_start

add r2, #4 // go to \_\_reset\_kernel\_\_

ldr r0, [r2]

bx r0

bootload\_end:

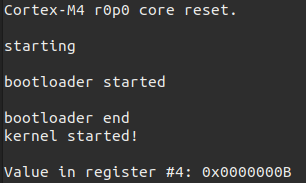
b bootload\_end

**Порядок виконання команд:**

**make**

**make qemu\_run**

**Результат виконання:**

****

**Перевірка результату**

**a = 6**

**b = 4**

**c = 3**

**(a+b)/2 + c!**

**(6+4)/2 + 3! = 1110 = B16**

**Обчислення виконано правильно.**

**Висновки:** Упродовж виконання лабораторної роботи я навчився працювати з оперативною пам’яттю, використовувати інструкції спеціального призначення, використовувати виключення процесора Cortex-M4. Створив мінімальний завантажувач системи. Навчився користуватися виводом даних через відлагоджувальний порт(консоль у нашому випадку).