Національний Технічний Університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра обчислювальної техніки

**Лабораторна робота №3**

з курсу «Архітектура комп’ютерів»

Виконав:

студент 3 курсу

групи ІВ-83

Литвиненко Олександр

Київ 2021

**Тема:** Завантажувач 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).

Варіант: 8317%16 = 3



**Код програми:**

**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

sub r6, r0, r1

add r4, r6, r2

loop:

ldr r3, [r0], #-4

str r3, [r4], #-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

**kernel.S**

.syntax unified

.cpu cortex-m4

.thumb

// The character # can be used for immediates (numbers)

// or if it stands at the first position of a line

// the whole line is counted as a comment.

#define a #3

#define b #7

#define c #5

.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 #0: "

.section .text

\_\_kernel\_reset\_\_:

ldr r0, =data

bl dbgput\_line

// r0-r12 general purpose registers

mov r0, a

mov r1, b

mov r2, c

// bl - move to calc and save return address to LR

bl calc

mov r5, r0

ldr r0, =final

bl dbgput

mov r0, r5

bl dbgput\_num

end:

b end

calc:

push { r3, r4, lr }

// adds r0 and r1. Result in r5

add r4, r0, r1 //add a + b

mov r3, #2 // 2

// unsigned div

sdiv r4, r4, r3 //(a+b)/2

mov r0, r2 // mov r0, c (because bne compares r0)

bl factorial

add r0, r0, r4

pop { r3, r4, pc }

factorial:

push { r4 }

mov r4, 1

.loop:

mul r4, r4, r0

// sub and change flags

// N - result greater or equal to 0

// Z - result is zero

// C - carry (did not fit)

// V - Overflow

subs r0, r0, #1

// if r0 is not zero jump to func

bne .loop

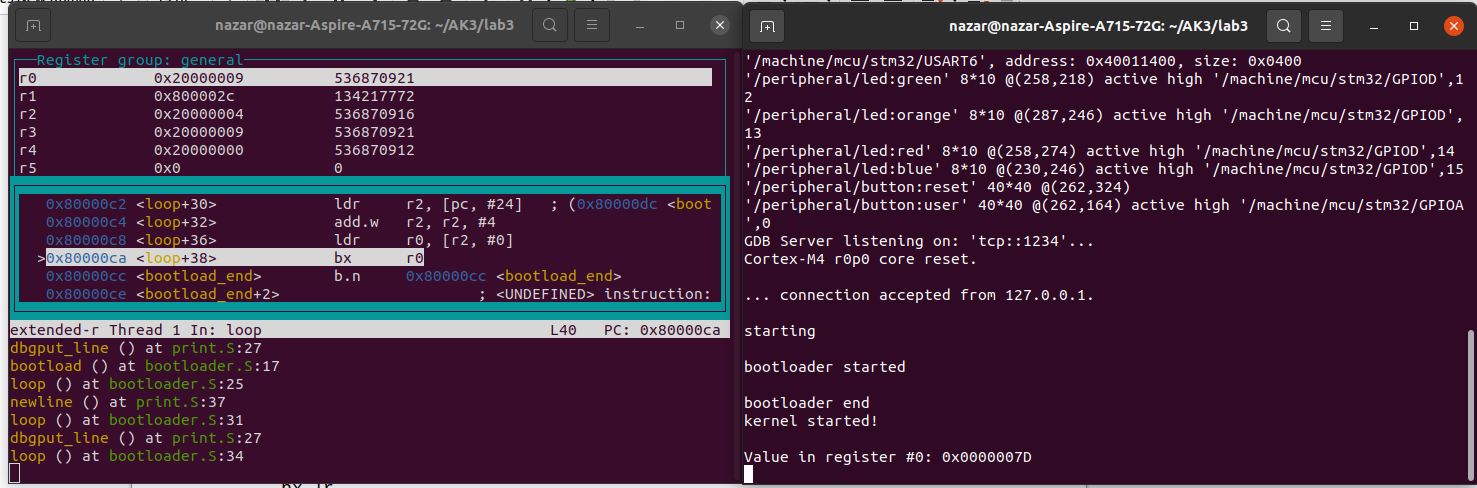
mov r0, r4

pop { r4 }

// go to link register

bx lr

**Приклади роботи програми:**



**Висновки:** У ході виконання лабораторної роботи було створено простий завантажувач за варіантом, який завантажує програму обчислення виразу в оперативну пам’ять