НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №1.3

з дисципліни“Архітектура комп’ютерів – 3”

Виконав:

студент групи ІВ-83

ЗК ІВ-8311

Бондаренко М.О

Київ 2021

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

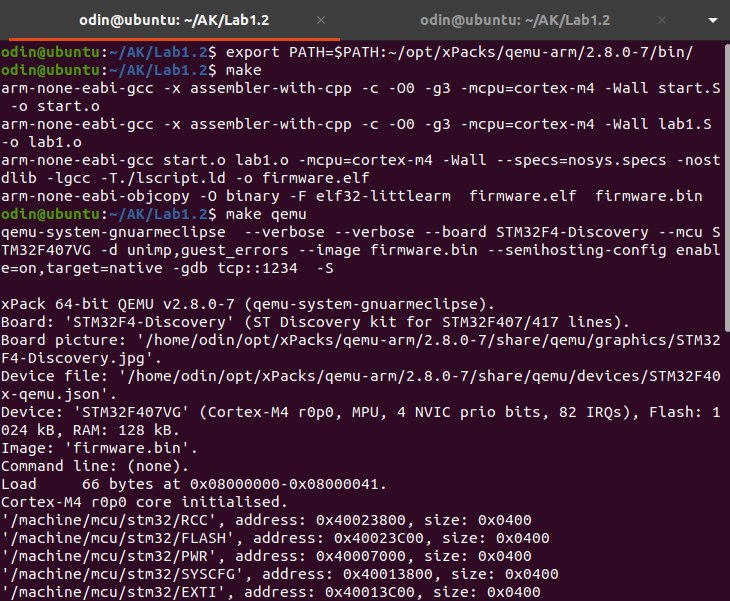
**Мета**: Навчитися працювати з оперативною пам’яттю, використовувати інструкції спеціального призначення, використовувати виключення процесора Cortex-M4. Створення мінімального завантажувача системи. Навчитися користуватися виводом даних через відлагоджувальний порт (або консоль).

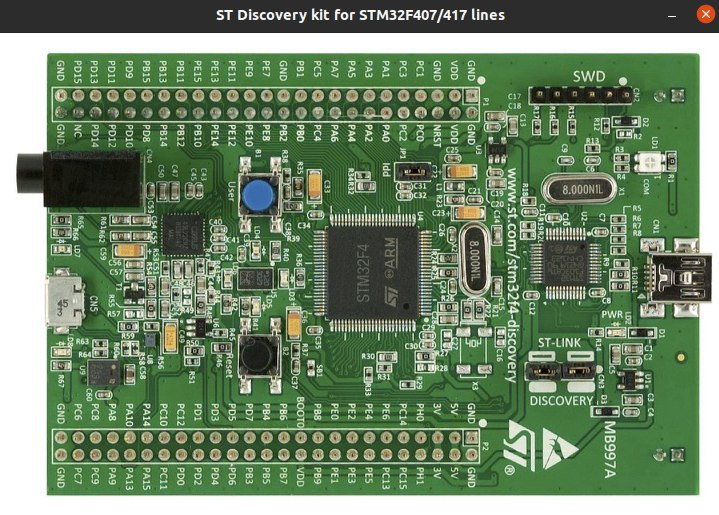
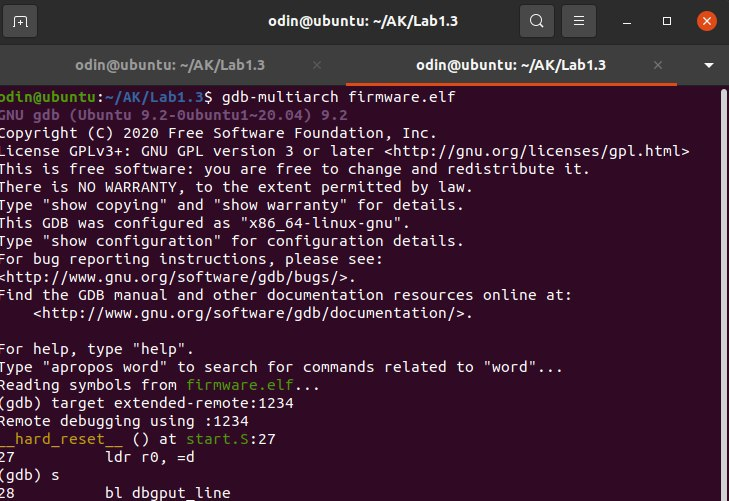
**8311 mod 16 = 7**

**Варіант:**



**Скріншоти роботи:**



**Лістинг коду**

1. **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 start..."

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 // blt

bl newline

ldr r0, =str\_boot\_end

bl dbgput\_line

ldr lr, =bootload\_end

add lr, #1

ldr r2, =\_ram\_start

add r2, #4

ldr r0, [r2]

bx r0

bootload\_end:

b bootload\_end

1. **kernel.S**

.syntax unified

.cpu cortex-m4

//.fpu softvfp

.thumb

.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

start: .asciz "kernel start\n"

finish: .asciz "result in reg #3: "

.section .text

\_\_kernel\_reset\_\_:

ldr r0, =start

bl dbgput\_line

// calc part

// (A-B)

mov r0, #10

// ^ r0 = A = 10

mov r1, #5

// ^ r1 = B = 5

SUB r0, r0, r1

// (A-B)\*3

mov r1, #3

MUL r0, r0, r1

// 2^C

mov r1, #3

// ^ r1 = C = 3

mov r2, #1

mov r3, #2

Start:

cmp r1, #0

ble End

mul r2, r2, r3

sub r1, r1, #1

b Start

End:

// (A-B)\*3+2^C

add r3, r0, r2

ldr r0, =finish

bl dbgput

mov r0, r3 // Move result to r0

bl dbgput\_num

end:

b end

**Висновок**

Було створено програмний проект на мові асемблера та перевірено його виконання відлагоджувачем.

В результаті, отримано очікувані результати, що показано на скріншотах виконання програми.