НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №2**

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

Виконав:

студент 3 курсу

групи ІВ-83

Литвиненко Олександр

Київ 2021 р.

**Лістинг програми:**

**lab1.S**

.global lab1

.syntax unified

#define P0\_DATA #3

#define P1\_DATA #5

#define P2\_DATA #4

lab1:

push {lr}

// calculate

mov r0, P0\_DATA

mov r1, P1\_DATA

mov r2, P2\_DATA

bl calc

pop {pc}

calc:

push { r4, r5, lr }

add r5, r0, r1

mov r4, #2

sdiv r5, r4

mov r0, r2

bl factorial

add r0, r5

pop { r4, r5, pc }

factorial:

//param: 1 number

push { r4 }

mov r4, 1

.fact\_calc:

mul r4, r0

subs r0, #1

bne .fact\_calc

mov r0, r4

pop { r4 }

bx lr

**start.S**

.syntax unified

.cpu cortex-m4

//.fpu softvfp

.thumb

// Global memory locations.

.global vtable

.global \_\_hard\_reset\_\_

/\*

\* vector table

\*/

.type vtable, %object

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

vtable:

.word \_\_stack\_start

.word \_\_hard\_reset\_\_+1

.size vtable, .-vtable

\_\_hard\_reset\_\_:

// initialize stack here

// if not initialized yet

bl lab1

\_loop: b \_loop

.size \_\_hard\_reset\_\_, .-\_\_hard\_reset\_\_

**Iscript.Id**

MEMORY

{

FLASH ( rx )

 : ORIGIN = 0x08000000, LENGTH = 1M

RAM ( rxw )

 : ORIGIN = 0x20000000, LENGTH = 128K

}

\_\_stack\_start = ORIGIN(RAM) + LENGTH(RAM);

**Makefile**

SDK\_PREFIX?=arm-none-eabi-

CC = $(SDK\_PREFIX)gcc

LD = $(SDK\_PREFIX)ld

SIZE = $(SDK\_PREFIX)size

OBJCOPY = $(SDK\_PREFIX)objcopy

QEMU = qemu-system-gnuarmeclipse

BOARD ?= STM32F4-Discovery

MCU=STM32F407VG

TARGET=firmware

CPU\_CC=cortex-m4

TCP\_ADDR=1234

deps = \

start.S \

lscript.ld

all: target

target:

$(CC) -x assembler-with-cpp -c -O0 -g3 -mcpu=$(CPU\_CC) -Wall start.S -o start.o

$(CC) -x assembler-with-cpp -c -O0 -g3 -mcpu=$(CPU\_CC) -Wall lab1.S -o lab1.o

$(CC) start.o lab1.o -mcpu=$(CPU\_CC) -Wall --specs=nosys.specs -nostdlib -lgcc -T./lscript.ld -o $(TARGET).elf

$(OBJCOPY) -O binary -F elf32-littlearm $(TARGET).elf $(TARGET).bin

qemu:

$(QEMU) --verbose --verbose --board $(BOARD) --mcu $(MCU) -d unimp,guest\_errors --image $(TARGET).bin --semihosting-config enable=on,target=native -gdb tcp::$(TCP\_ADDR) -S

clean:

-rm \*.o

-rm \*.elf

-rm \*.bin

flash:

st-flash write $(TARGET).bin 0x08000000

**Скріншоти**



