НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №3**

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

Виконав:

студент 3 курсу

групи ІВ-81

Бєлов М.С.

Перевірив:

Нікольський С.С.

Київ 2021 р.

Варіант 8102 % 16 = 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | LDRB, STRB | декремент\* | числовий | 1 |

**Лістинг 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 r4, r1, r0

sub r2, r4

loop:

ldrb r3, [r0], #-1

strb r3, [r2], #-1

cmp r1, r0

blt 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

//.fpu softvfa = 5

.thumb

#define \_a #5

#define \_b #7

#define \_c #2

// 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 #3: "

.section .text

\_\_kernel\_reset\_\_:

ldr r0, =data

bl dbgput\_line

// push {lr}

// calculate

mov r0, \_a

mov r1, \_b

mov r2, \_c

bl calc

ldr r0, =final

bl dbgput

mov r0, r3

bl dbgput\_num

// pop {pc}

calc:

push { r0, r1, r2, lr }

ands r3, r0, r1

//mov r1, #1

asr r1, r3, #1

mov r0, r2

bl factorial

add r3, r1, r0

pop { r0, r1, r2, pc }

factorial:

//param: 1 number

push { r4 }

mov r4, 1

.fact\_calc:

mul r4, r0

subs r0, #1

bne .fact.syntax unified

.cpu cortex-m4

//.fpu softvfa = 5

.thumb

#define \_a #5

#define \_b #7

#define \_c #2

// 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 #3: "

.section .text

\_\_kernel\_reset\_\_:

ldr r0, =data

bl dbgput\_line

// push {lr}

// calculate

mov r0, \_a

mov r1, \_b

mov r2, \_c

bl calc

ldr r0, =final

bl dbgput

mov r0, r3

bl dbgput\_num

// pop {pc}

calc:

push { r0, r1, r2, lr }

ands r3, r0, r1

//mov r1, #1

asr r1, r3, #1

mov r0, r2

bl factorial

add r3, r1, r0

pop { r0, r1, r2, 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

\_calc

mov r0, r4

pop { r4 }

bx lr

**Скріншоти**



**Висновок**

У ході виконання лабораторної роботи було створено простий завантажувач за варіантом, який завантажує програму обчислення виразу в оперативну пам’ять. Роботу програми було протестовано для всіх можливих випадків. Програма працює коректно.