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 ІB-81

Мороз Д. Ю.

Київ-2021 р.

**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:**

8120%16 = 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Команди для роботи з пам’яттю** | **Інкремент/Декремент регістру адреси** | **Вид зсуву** | **Кількість байт для зсуву** |
| 8 | LDRH, STRH | інкремент | числовий | 2 |

**Репозиторій:** [**https://github.com/HPMortys/Lab3\_AK3**](https://github.com/HPMortys/Lab3_AK3)

**Лістинг:**

(програма за варіантом)

bootloader.S

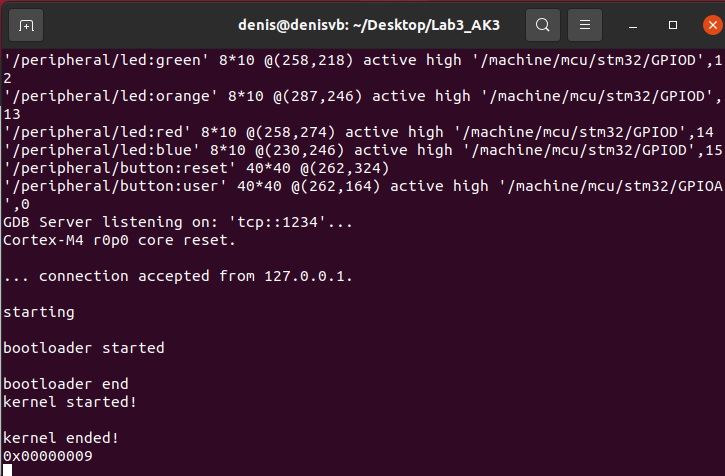
|  |
| --- |
| .syntax unified |
|  | .cpu cortex-m4 |
|  | //.fpu softvfp |
|  | .thumb |
|  | .global bootload |
|  | .section .rodata |
|  | image: .incbin "kernel.bin" |
|  | end\_of\_image: |
|  | str\_boot\_start: .asciz "bootloader started" |
|  | str\_boot\_end: .asciz "bootloader end" |
|  | 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: |
|  | ldrh r3, [r1], #2 |
|  | strh r3, [r2], #2 |
|  |  |
|  | 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 |
|  | //.fpu softvfp |
|  | .thumb |
|  |  |
|  | #define A #4 |
|  | #define B #2 |
|  | #define C #3 |
|  |  |
|  | .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" |
|  | data\_end: .asciz "kernel ended!\n" |
|  | .section .text |
|  | \_\_kernel\_reset\_\_: |
|  | ldr r0, =data |
|  | bl dbgput\_line |
|  |  |
|  |  |
|  | // calculate (a+b)/2 + c! |
|  | mov r0, A |
|  | mov r1, B |
|  | mov r2, C |
|  | bl test\_var |
|  | test\_var: |
|  | push { r0, r1, r2, lr } |
|  | ADD r0, r0, r1 // a + b |
|  | ASR r0, #1 // / 2 |
|  | mov r3, #1 |
|  | cfact: // ! |
|  | cmp r2, #1 |
|  | bEQ res |
|  | MUL r3, r2 |
|  | SUB r2, #1 |
|  | bNE cfact |
|  | res: |
|  | ADD r0, r3 |
|  | mov r3, r0 |
|  | ldr r0, =data\_end |
|  | bl dbgput |
|  | mov r0, r3 |
|  | bl dbgput\_num |
|  |  |
|  | end: |
|  | b end |

**Результат**





Перевірка:

a = 4

b = 2

c = 3

(4+2)/2 + 3! =3 + 3! = 9

**Висновок**

В ході виконання лабораторної роботи створено мінімальний завантажувач системи. Також, навчилися працювати з оперативною пам’яттю, використовуючи інструкції спеціального призначення, використовуючи виключення процесора Cortex-M4.