

**Отчёт по лабораторной работе №3**

Дисциплина: Низкоуровневое программирование

Тема: Программирование RISC-V

Выполнил студент гр. 3530901/10005 \_\_\_\_\_ Кашин В.А.  
(подпись)

Преподаватель \_\_\_\_\_ Коренев Д. А.  
(подпись)

“ \_\_\_\_ ” \_\_\_\_\_ 2022 г.

Санкт-Петербург  
2022

### Вариант задания – 3: Сортировка обменом массива чисел in-place.

#### Реализация программы 1:

```
1 .text
2 __start:
3 .globl __start
4 lw a2, array_lenght #Длина массива
5 li a3, 0             #i
6 li a4, 0             #j
7 la a5, array         #addr1 min
8 la a6, array         #addr2 least
9 la a7, array
10
11 loop_i:
12 bgeu a3, a2, loops_exit #if (a3>=a2) exit if (i>= arr_len)
13 addi a6, a5, 0 #a6 = a5 + 0
14 addi a3, a3, 1 #a3 = a3 + 1
15 addi a4, a3, 0 #a4 = a3 + 0
16 loop_j: #j= min + 1
17 bgeu a4, a2, loop_j_exit #if (a4>=a2) exit
18 slli t1, a4, 2 #t1 = a4 << 2 = a4*4
19 addi a4, a4, 1 #a4 = a4 + 1
20 add t1, a7, t1 #t1 = a7 + t1
21 lw t2, 0(t1) #t2 = arr[j]
22 lw t3, 0(a6) #t3 = arr[least]
23 bgtu t2, t3, loop_j #arr[j] > arr[least]
24 addi a6, t1, 0
25 j loop_j
26
27 loop_j_exit:
28 lw t4, 0(a5)
29 lw t5, 0(a6)
30 sw t4, 0(a6) #least=min
31 sw t5, 0(a5) #min =least
32 addi a5, a5, 4
33 j loop_i
34
35 loops_exit:
36 li a0, 10 # x10 = 10
37 ecall # ecall при значении x10 = 10 => останов симулятора
38
39 .rodata
40 array_lenght:
41 .word 4
42
43 .data
44 array:
45 .word 4, 3, 2, 1
```

(Код основной программы)

### Работа программы 1:

|            |    |    |    |    |
|------------|----|----|----|----|
| 0x00010094 | 00 | 00 | 00 | 01 |
| 0x00010090 | 00 | 00 | 00 | 02 |
| 0x0001008c | 00 | 00 | 00 | 03 |
| 0x00010088 | 00 | 00 | 00 | 04 |
| 0x00010084 | 00 | 00 | 00 | 04 |

(до запуска)

|            |    |    |    |    |
|------------|----|----|----|----|
| 0x00010094 | 00 | 00 | 00 | 04 |
| 0x00010090 | 00 | 00 | 00 | 03 |
| 0x0001008c | 00 | 00 | 00 | 02 |
| 0x00010088 | 00 | 00 | 00 | 01 |
| 0x00010084 | 00 | 00 | 00 | 04 |

(после запуска)

Программа работает корректно

## Реализация программы 2 с подпрограммой :

```
37 .text
38 sortSup:
39 .globl sortSup
40 loop_i:
41 bgeu a3, a2, loops_exit #if ((a3>=a2) exit if (i>= arr_len)
42 addi a6, a5, 0
43 addi a3, a3, 1 #min++
44 addi a4, a3, 0
45 loop_j: #j= min + 1
46 bgeu a4, a2, loop_j_exit
47 slli t1, a4, 2
48 addi a4, a4, 1
49 add t1, a7, t1
50 lw t2, 0(t1) #t2 = arr[j]
51 lw t3, 0(a6) #t3 = arr[least]
52 bgtu t2, t3, loop_j #arr[j] > arr[least]
53 addi a6, t1, 0
54 j loop_j
55
56 loop_j_exit:
57 lw t4, 0(a5)
58 lw t5, 0(a6)
59 sw t4, 0(a6) #least=min
60 sw t5, 0(a5) #min =least
61 addi a5, a5, 4
62 j loop_i
63
64 loops_exit:
65 ret
```

(подпрограмма sortSup)

```
10 .text
11 sortMain:
12 .globl sortMain
13 lw a2, array_lenght #Длина массива
14 li a3, 0 #i
15 li a4, 0 #j
16 la a5, array #addr1 min
17 la a6, array #addr2 least
18 la a7, array
19
20 addi sp, sp, -16 # выделение памяти в стеке
21 sw ra, 12(sp) # сохранение ra
22
23 call sortSup # call fun
24
25 lw ra, 12(sp) # восстановление ra
26 addi sp, sp, 16 # освобождение памяти в стеке
27
28 ret # } return 0;
29 .rodata
30 array_lenght:
31 .word 4
32
33 .data
34 array:
35 .word 4, 3, 2, 1
36
```

(подпрограмма sortMain)

### Работа программы 2 :

|            |    |    |    |    |
|------------|----|----|----|----|
| 0x000100c0 | 00 | 00 | 00 | 01 |
| 0x000100bc | 00 | 00 | 00 | 02 |
| 0x000100b8 | 00 | 00 | 00 | 03 |
| 0x000100b4 | 00 | 00 | 00 | 04 |
| 0x000100b0 | 00 | 00 | 00 | 04 |

(До запуска)

|            |    |    |    |    |
|------------|----|----|----|----|
| 0x000100c0 | 00 | 00 | 00 | 04 |
| 0x000100bc | 00 | 00 | 00 | 03 |
| 0x000100b8 | 00 | 00 | 00 | 02 |
| 0x000100b4 | 00 | 00 | 00 | 01 |
| 0x000100b0 | 00 | 00 | 00 | 04 |

(После запуска)

Программа работает корректно