

Санкт-Петербургский политехнический университет Петра Великого  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе № 3**

**Дисциплина:** Низкоуровневое программирование

**Тема:** Программирование RISC-V

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

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

“ ” \_\_\_\_\_

Санкт-Петербург

## 1 Техническое задание

Определение наиболее часто встречающегося в массиве элемента.

## 2 Метод решения

Подсчет количества вхождений для каждого элемента в массиве. Создание вложенного цикла, в котором считается количество совпадений для текущего элемента (из внешнего цикла, который проходится по всем элементам). Если совпадений больше максимального количества совпадений, то перезапись этого максимума и сохранение данного элемента. В качестве maxcounter используется t3, maxCounteredElement – t5, a2 – счетчик для первого цикла, a4 – счетчик для внутреннего цикла

## 3 Реализация программы

```
1 .text
2 start:
3 .globl start
4     la a3, array_length
5     lw a3, 0(a3)           # a3 = длина массива
6
7     li a2, 0               #a2 = 0
8     li t3, 0 #maxcounter
9     li t5, 0 #maxnumber
10 loop1:
11     bgeu a2, a3, loop1_exit # a2 >= a3 goto loop1_exit
12     li t4, 0 #currentcounter
13
14     li a4, 0
15     la a5, array # a5 - адрес нулевого элемента
16     slli a6, a2, 2 # a6 = a2 * 4
17     add a6, a5, a6 # a6 = a5 + a6
18     lw t0, 0(a6) # t0 = array[i]
19     loop2:
20         bgeu a4, a3, loop2_exit
21         slli a7, a4, 2 # a7 = a4 * 4
22         add a7, a5, a7 # a7 = a5 + a7
23         lw t1, 0(a7) # t1 = array[j]
24         #addi t2, t1, -t0 # t2 = t1 - t0
25         bne t1, t0, continueloop
26         #bnez t2, continueloop # t1-t0 != 0
```

```

27     addi t4, t4, 1 # counter++
28     bltu t4, t3, continueloop
29     mv t3, t4
30     mv t5, t0
31     continueloop:
32     addi a4, a4, 1
33     jal zero, loop2
34 loop2_exit:
35     addi a2, a2, 1
36     jal zero, loop1 # goto loop
37 loop1_exit:
38     la a2, result
39     sw t5, 0(a2)
40     li a0, 10 # x10 =10
41     ecall
42
43 .rodata
44 array_length:
45     .word 10
46 array:
47     .word 1, 2, 1, 2, 1, 3, 6, 6, 6, 1
48 .data
49 result:
50     .word 0

```

#### 4 Реализация с подпрограммой

```

1 #setup.s
2 .text
3 start:
4 .globl start
5     call main
6
7 finish:
8     mv a1, a0
9     li a0, 17
10    ecall

```

```

1 #main.s
2 .text
3 main:
4 .globl main
5     la a0, array
6     lw a1, array_length
7
8     addi sp, sp, -16
9     sw ra, 12(sp)
10
11     call find
12
13     lw ra, 12(sp)
14     addi sp, sp, 16
15
16     la a2, result
17     sw a0, 0(a2)
18     li a0, 0
19     ret
20
21 .rodata
22 array_length:
23     .word 10
24 array:
25     .word 1, 2, 1, 2, 1, 3, 6, 6, 6, 1
26 .data
27 result:
28     .word 0

```

```

1 #find
2 .text
3 find:
4 .globl find
5     # a0 - адрес 0го элемента
6     # a1 - длина массива
7
8     li a2, 0           #a2 = 1
9     li t3, 0 #maxcounter
10    li t5, 0 #maxnumber
11 loop1:
12    bgeu a2, a1, loop1_exit # a2 >= a3 goto loop1_exit
13    li t4, 0 #currentcounter

```

```

14  li a4, 0
15  slli a6, a2, 2 # a6 = a2 * 4
16  add a6, a0, a6 # a6 = a5 + a6
17  lw t0, 0(a6) # t0 = array[i]
18  loop2:
19      bgeu a4, a1, loop2_exit
20      slli a7, a4, 2 # a7 = a4 * 4
21      add a7, a0, a7 # a7 = a5 + a7
22      lw t1, 0(a7) # t1 = array[j]
23      bne t1, t0, continueloop
24      addi t4, t4, 1 # counter++
25      bltu t4, t3, continueloop
26      mv t3, t4
27      mv t5, t0
28      continueloop:
29          addi a4, a4, 1
30          jal zero, loop2
31  loop2_exit:
32      addi a2, a2, 1
33      jal zero, loop1 # goto loop
34  loop1_exit:
35      mv a0, t5
36      ret

```

## 5 Руководство программисту

Начальные данные к программе: адрес нулевого элемента массива (и соответственно сам массив) и его длина. В реализации без подпрограммы адрес и длина хранятся в регистрах a5 и a3 соответственно. В реализации через подпрограмму предполагается, что нулевым аргументом (регистр a0) передается адрес нулевого элемента массива и первым аргументов (регистр a1) — длина массива.

## 6 Испытание программ

### 6.1 Без подпрограмм

array_length	0x00010088
array	0x0001008c
result	0x000100b4

Адрес результата

0x000100b4	00	00	00	01
0x000100b0	00	00	00	01
0x000100ac	00	00	00	06
0x000100a8	00	00	00	06
0x000100a4	00	00	00	06
0x000100a0	00	00	00	03
0x0001009c	00	00	00	01
0x00010098	00	00	00	02
0x00010094	00	00	00	01
0x00010090	00	00	00	02
0x0001008c	00	00	00	01
0x00010088	00	00	00	0a

Результат в 0x000100b4, массив не изменился

С подпрограммой:

array_length	0x000100b8
array	0x000100bc
result	0x000100e4

Адрес результата

0x000100e4	00	00	00	01
0x000100e0	00	00	00	01
0x000100dc	00	00	00	06
0x000100d8	00	00	00	06
0x000100d4	00	00	00	06
0x000100d0	00	00	00	03
0x000100cc	00	00	00	01
0x000100c8	00	00	00	02
0x000100c4	00	00	00	01
0x000100c0	00	00	00	02
0x000100bc	00	00	00	01
0x000100b8	00	00	00	0a

Результат в 0x000100e4, массив не изменился