

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                      гр.3530901/10005    Севдат Туфаногуллари

Преподаватель

“   ” \_\_\_\_\_

Санкт-Петербург

2022

## Оглавление

1. Техническое задание
2. Метод решения
3. Руководство программисту
4. Реализация программы 1
5. Работа программы 1
6. Реализация программы 2 с подпрограммой
7. Работа программы 2

## 1. Техническое задание

Реализовать нахождение наибольшего общего делителя (НОД) для массива чисел при помощи EDSAC.

## 2. Метод решения

Идея алгоритма заключается в том, что мы вычитаем из большего числа меньшее и заменяем первое на их разность до тех пор, пока их разность не станет равна нулю. В таком случае уменьшаемое и вычитаемое как раз и будут искомым числом.

### Способ решения

Наименьший число, найденное в списке, вычитается из каждого числа до тех пор, пока не будет найдено значение, которое при вычитании из каждого числа в списке приведет к нулю. Пример:

- $A = 10; B = 2$ 
  - $A - B = 8$
  - $A - B = 6$
  - $A - B = 4$
  - $A - B = 2$
  - $A - B = 0$
- Ответ = 2
- 

## 3. Руководство программисту

Программа должна иметь размер списка и сам список, чтобы найти наибольшее число, которое можно разделить для каждого элемента списка. Размер списка сохраняется в переменной в t2, а адрес первого элемента сохраняется в t3. Ответ сохраняется в t4. Переменные t2 и t3 являются константами и не изменяются на протяжении всей программы.

#### 4. Реализация программы 1

```
● Working1-riscv3.s ×
1 .globl __start
2 .rodata
3 array_length:
4 .word 5
5 .data
6 array:
7 .word 10, 5, 20, 60, 200
8 .text
9
10 __start:
11 li t0, 1
12 la t1, array_length #address
13 la t3, array #addressOfFirst
14 lw t2, 0(t1) #size 5
15 lw t4, 0(t3) #firstElement 50
16
17 Loop0:
18 beq t2, t0, continue
19 addi t3, t3, 4 #nextAddress
20 lw t5, 0(t3) #nextElement 20
21 bgeu t4, t5, goToMarking
22 addi t2, t2, -1 #size -= 1
23 j Loop0
24 goToMarking:
25 mv t4, t5
26 addi t2, t2, -1 #size -= 1
27 j Loop0
28
29 continue:
30
31 li t0, 1
32 la t1, array_length #address
33 la t3, array #addressOfFirst
34 lw t2, 0(t1) #size 5
35 li t6, -1
36 addi t4, t4, 1 # smal += 1
37 mul t4, t4, t6 # smal = -7
38
```

## ● Working1-riscv3.s ×

```
33 la t3, array #addressOfFirst
34 lw t2, 0(t1) #size 5
35 li t6, -1
36 addi t4,t4,1 # smal += 1
37 mul t4,t4,t6 # smal = -7
38
39
40 restart:
41 addi t4, t4, 1 #smal += 1
42 beq t4, zero, finish
43 mv a2, t2 # a2 = size
44 mv a3, t3 #a3 = adressOfFirst
45 mul a5, t4, t6
46
47 nextElement:
48 addi a2, a2, -1 # size -= 1
49 beq a2, zero, finish # if (size == 0) finish
50 lw a4, 0(a3) #element = 80
51
52 repeat:
53 beq a4, a5, Next # if (element == positsmal) Next
54 bgeu a5,a4, restart # if (-1 >= element) restart
55 add a4, a4, t4 # element -= 6
56 j repeat
57
58 Next:
59 addi a3,a3, 4 #NextElement
60 j nextElement
61
62 finish:
63 mul t4,t6,t4
64 li a0, 1
65 mv a1,t4
66 ecall
67 li a0, 10
68 ecall
```

## 5. Работа программы 1

Программа выполнена при таких входных данных:

```
2 .rodata
3 array_length:
4 .word 5
5 .data
6 array:
7 .word 10, 5, 20, 60, 200
8 .text
```

Jupyter

File Edit Run Settings Help

Editor Simulator

▶ ⏮ ⏪ ⏩ ⏭ 🔍 📄 🖨

Bkpt	Address	Machine Code	Basic Code	Source Code
<input type="checkbox"/>	0x00010094	0x01d70733	add x14, x14, x29	add a4, a4, t4
<input type="checkbox"/>	0x00010098	0xff5ff06f	jal x0, -12	j repeat
<input type="checkbox"/>	0x0001009c	0x00468693	addi x13, x13, 4	addi a3, a3, 4
<input type="checkbox"/>	0x000100a0	0xfe1ff06f	jal x0, -32	j nextElement
<input type="checkbox"/>	0x000100a4	0x03df8eb3	mul x29, x31, x29	mul t4, t6, t4
<input type="checkbox"/>	0x000100a8	0x00100513	addi x10, x0, 1	li a0, 1
<input type="checkbox"/>	0x000100ac	0x000e8593	addi x11, x29, 0	mv a1, t4
<input type="checkbox"/>	0x000100b0	0x00000073	ecall	ecall
<input type="checkbox"/>	0x000100b4	0x00a00513	addi x10, x0, 10	li a0, 10
<input type="checkbox"/>	0x000100b8	0x00000073	ecall	ecall

Registers			Memory	Cache
Mnemonic	Number	Value		
s7	x23	0x00000000		
s8	x24	0x00000000		
s9	x25	0x00000000		
s10	x26	0x00000000		
s11	x27	0x00000000		
t3	x28	0x000100c0		
t4	x29	0x00000005		
t5	x30	0x000000c8		
t6	x31	0xffffffff		

Integer (X) Floating (F)

Console

5

Ответ:            **t4**            **x29**            **0x00000005**

## 6. Реализация программы 2 с подпрограммой

```
start.s × | main.s × | NOD2.s × |
1 # setup.s
2 .text
3
4 __start:
5
6 .globl __start
7     call main
8
9 finish:
10
11     li a0, 10 # a0 = 17
12     ecall # выход с кодом завершения
13
14
15
16
17
```

start.s × main.s × NOD2.s ×

```
1
2 .rodata
3 array_length:
4 .word 5
5 msg1: .string "Hello"
6 .data
7 array:
8 .word 10, 5, 20, 60, 200
9 .text
10
11 main:
12 .global main
13 li t0, 1
14 la t1, array_length #address
15 la t3, array #addressOfFirst
16 mv a7, t1
17 mv a6, t3
18 lw t2, 0(t1) #size 5
19 lw t4, 0(t3) #firstElement 50
20
21 addi sp, sp, -16 #выделение памяти в стеке
22 sw ra, 12(sp) # сохранение
23 call NOD2
24 lw ra, 12(sp) # восстанавливаем
25 addi sp, sp, 16 # освобождение памяти в сетке
```

```
23 call NOD2
24 lw ra, 12(sp) # восстанавливаем
25 addi sp, sp, 16 # освобождение памяти в сетке
26
27 finish:
28 mul t4, t6, t4
29 li a0, 1
30 mv a1, t4
31 ecall
32 ret
```



start.s × | main.s × | NOD2.s ×

```
1 .text
2 NOD2:
3 .global NOD2
4
5 Loop0:
6 beq t2, t0, continue
7 addi t3, t3, 4 #nextAddress
8 lw t5, 0(t3) #nextElement 20
9 bgeu t4, t5, goToMarking
10 addi t2, t2, -1 #size -= 1
11 j Loop0
12 goToMarking:
13 mv t4, t5
14 addi t2, t2, -1 #size -= 1
15 j Loop0
16
17 continue:
18
19 li t0, 1
20 mv t1, a7 #adress
21 mv t3, a6 #adressOfFirst
22 lw t2, 0(t1) #size 5
23 li t6, -1
24 addi t4, t4, 1 # smal += 1
25 mul t4, t4, t6 # smal = -7
26
27
```

```

27
28 restart:
29 addi t4, t4, 1 #smal += 1
30 beq t4, zero, finish
31 mv a2, t2 # a2 = size
32 mv a3, t3 #a3 = adressOfFirst
33 mul a5, t4, t6
34
35 nextElement:
36 addi a2, a2, -1 # size -= 1
37 beq a2, zero, finish # if (size == 0) finish
38 lw a4, 0(a3) #element = 80
39
40 repeat:
41 beq a4, a5, Next # if (element == positsmal) Next
42 bgeu a5, a4, restart # if (-1 >= element) restart
43 add a4, a4, t4 # element -= 6
44 j repeat
45
46 Next:
47 addi a3, a3, 4 #NextElement
48 j nextElement
49
50 finish:
51 ret

```

## 7. Работа программы 2

Программа выполнена при таких входных данных:

```
2 .rodata
3 array_length:
4 .word 5
5 .data
6 array:
7 .word 10, 5, 20, 60, 200
8 .text
```

Jupyter

File Edit Run Settings Help

Editor Simulator

Registers Memory Cache

Bkpt	Address	Machine Code	Basic Code	Source Code	Mnemonic	Number	Value
<input type="checkbox"/>	0x00010000	0x00000317	auipc x6, 0	auipc x6, 0	s6	x22	0x00000000
<input type="checkbox"/>	0x00010004	0x00830007	jalr x0, x6, 8	jalr x0, x6, 8	s7	x23	0x00000000
<input type="checkbox"/>	0x00010008	0x00000317	auipc x6, 0	call main	s8	x24	0x00000000
<input type="checkbox"/>	0x0001000c	0x010300e7	jalr x1, x6, 16	call main	s9	x25	0x00000000
<input type="checkbox"/>	0x00010010	0x00a00513	addi x10, x0, 10	li a0, 10	s10	x26	0x00000000
<input type="checkbox"/>	0x00010014	0x00000073	ecall	ecall	s11	x27	0x00000000
<input type="checkbox"/>	0x00010018	0x00100293	addi x5, x0, 1	li t0, 1	t3	x28	0x000100e8
<input type="checkbox"/>	0x0001001c	0x00000317	auipc x6, 0	la t1, array_length	t4	x29	0x00000005
<input type="checkbox"/>	0x00010020	0x00c830313	addi x6, x6, 200	la t1, array_length	t5	x30	0x000000c8
<input type="checkbox"/>	0x00010024	0x00000e17	auipc x28, 0	la t3, array	t6	x31	0xffffffff

Integer (X) Floating (F)

Console

5

Ответ: t4 x29 0x00000005