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:

•
$$A = 10$$
; $B = 2$

$$\circ$$
 A-B = 8

$$\circ$$
 A-B = 6

$$\circ$$
 A-B = 4

$$\circ$$
 A-B = 2

$$\circ$$
 A-B = 0

 \bullet Other = 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:
8 .text
10 start:
12 la t1, array length #adress
13 la t3, array #adressOfFirst
14 lw t2, 0(t1) #size 5
17 Loop0:
18 beq t2, t0, continiue
21 bgeu t4, t5, goToMarking
23 j Loop0
24 goToMarking:
25 mv t4, t5
26 addi t2, t2, -1 #size -= 1
27 j Loop0
29 continiue:
32 la t1, array length #adress
33 la t3, array #adressOfFirst
37 mul t4, t4, t6 \# smal = -7
```

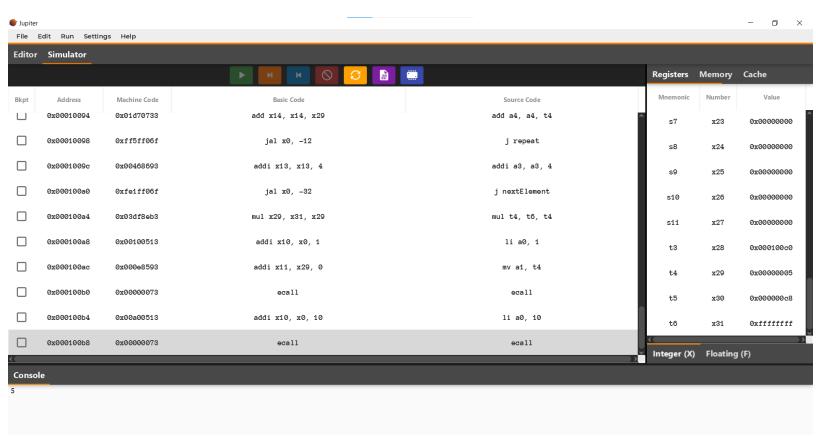
Working1-riscv3.s X

```
33 la t3, array #adressOfFirst
34 lw t2, 0(t1) #size 5
36 addi t4,t4,1 # smal += 1
37 mul t4, t4, t6 \# smal = -7
40 restart:
42 beq t4, zero, finish
43 mv a2, t2 \# a2 = size
45 mul a5, t4, t6
47 nextElement:
48 addi a2, a2, -1 # size -= 1
49 beq a2, zero, finish # if (size == 0) finish
50 \text{ lw a4, 0 (a3) } \#element = 80
52 repeat:
53 beq a4, a5, Next # if (element == positsmal) Next
54 bgeu a5, a4, restart # if (-1 >= element) restart
56 j repeat
58 Next:
59 addi a3,a3, 4 #NextElement
60 j nextElement
62 finish:
63 mul t4,t6,t4
65 mv a1,t4
67 li a0, 10
```

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
```



t4 x29 0x00000005

Ответ:

6. Реализация программы 2 с подпрограммой

```
start.s X main.s X NOD2.s X
2 .rodata
3 array length:
4 .word 5
5 msq1: .string "Hello"
6 .data
7 array:
11 main:
12 .global main
14 la t1, array length #adress
15 la t3, array #adressOfFirst
18 lw t2, 0(t1) #size 5
19 lw t4, O(t3) #firstElement 50
21 addi sp, sp, -16 #выделение памяти в стеке
22 sw ra, 12(sp) # сохранение
23 call NOD2
25 addi sp, sp, 16 # освобождение памяти в сетке
```

```
23 call NOD2
24 lw ra, 12(sp) # востанавливаем
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
```

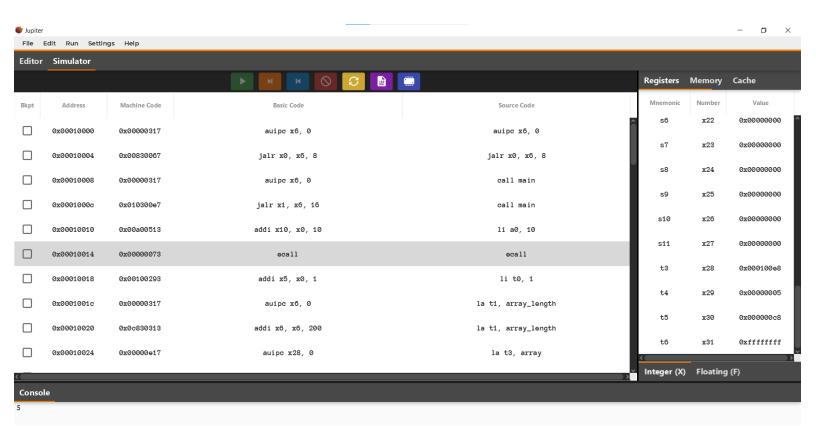
```
start.s × main.s × NOD2.s ×
 1 .text
2 NOD2:
3 .global NOD2
 5 Loop0:
6 beq t2, t0, continiue
7 addi t3, t3, 4 #nextAdress
8 lw t5, 0(t3) #nextElement 20
9 bgeu t4, t5, goToMarking
11 j Loop0
12 goToMarking:
13 mv t4, t5
14 addi t2, t2, -1 #size -= 1
15 j Loop 0
17 continiue:
19 li t0, 1
20 mv t1, a7 #adress
22 lw t2, 0(t1) #size 5
23 li t6, -1
24 addi t4,t4,1 # smal += 1
25 \text{ mul } t4, t4, t6 \# smal = -7
```

```
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
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
```



t4 x29 0x00000005 Ответ: