

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 RISC-V.

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

```
painP.s X ...
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
12 la t0, array #firstElementAdress
13 la t1, array_length #sizeAdress
14 lw t1, 0(t1) #size
15 li a2, 0 #count
16 li a3, 0 #save
17 li t2, -1
18
19 smallest:
20 bgeu a2, t1, finish
21 lw a4, 0(t0) #firstElement 10
22 bge a4, zero, positiveOne
23 mul a4, a4, t2
24 positiveOne:
25 beq a2, zero, addSmallest
26 bgeu a3,a4, addSmallest
27 addi t0, t0, 4
28 addi a2, a2, 1
29 j smallest
30 addSmallest:
```

```

30 addSmallest:
31 mv a3, a4
32 addi t0, t0, 4
33 addi a2, a2, 1
34 j smallest
35 finish:
36
37 mv a5, a3 #smallestInList
38 addi a5, a5, 1
39
40 restart:
41 addi a5, a5, -1
42 la t0, array #firstElementAdress
43 la t1, array_length #sizeAdress
44 lw t1, 0(t1) #size
45 li a2, 0 #count
46 li a3, 0 #save
47
48 nextNumber:
49 lw a4, 0(t0) #firstElement 10
50 bge a4, zero, positiveTwo
51 mul a4, a4, t2
52 positiveTwo:
53 beq a2, t1, end
54
55 addition:
56 add a3, a3, a5
57 beq a4, a3, possible
58 bgeu a3, a4, restart
59 j addition

```

```

65 j nextNumber
66
67
68 end:
69 mv a1, a5
70 li a0, 1
71 ecall
72 li a0, 10
73 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

Registers Memory Cache

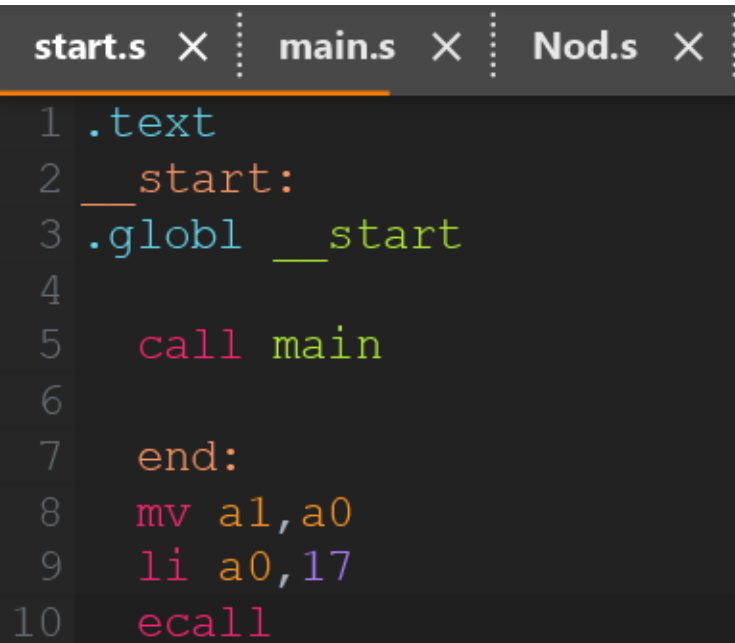
Bkpt	Address	Machine Code	Basic Code	Source Code	Mnemonic	Number	Value
<input type="checkbox"/>	0x000100a4	0x00000693	addi x13, x0, 0	li a3, 0	zero	x0	0x00000000
<input type="checkbox"/>	0x000100a8	0x00428293	addi x5, x5, 4	addi t0, t0, 4	ra	x1	0x00000000
<input type="checkbox"/>	0x000100ac	0x00160613	addi x12, x12, 1	addi a2, a2, 1	sp	x2	0xbfffffff0
<input type="checkbox"/>	0x000100b0	0xfd5ff06f	jal x0, -44	j nextNumber	gp	x3	0x10080000
<input type="checkbox"/>	0x000100b4	0x00078593	addi x11, x15, 0	mv a1, a5	tp	x4	0x00000000
<input type="checkbox"/>	0x000100b8	0x00100513	addi x10, x0, 1	li a0, 1	t0	x5	0x000100e0
<input type="checkbox"/>	0x000100bc	0x00000073	ecall	ecall	t1	x6	0x00000005
<input type="checkbox"/>	0x000100c0	0x00a00513	addi x10, x0, 10	li a0, 10	t2	x7	0xffffffff
<input type="checkbox"/>	0x000100c4	0x00000073	ecall	ecall	a0	x8	0x00000000

Integer (X) Floating (F)

Console

Ответ: a5 x15 0x00000005

6. Реализация программы 2 с подпрограммой



The image shows a screenshot of an IDE with three tabs: `start.s`, `main.s`, and `Nod.s`. The `start.s` tab is active and contains the following assembly code:

```
1 .text
2 __start:
3 .globl __start
4
5     call main
6
7     end:
8     mv a1,a0
9     li a0,17
10    ecall
```

start.s X | main.s X | Nod.s X |

```
1 .rodata
2 array_length:
3 .word 5
4 .data
5 array:
6 .word -10, 5, 20, -60, 200
7 .text
8
9 main:
10 .globl main
11
12 la a0, array #firstElementAddress
13 la a1, array_length #sizeAddress
14 lw a1, 0(a1) #size
15
16 addi sp, sp, -16 #выделение памяти в стеке
17 sw ra, 12(sp) # сохранение
18 call Nod
19 lw ra, 12(sp) # восстанавливаем
20 addi sp, sp, 16 # освобождение памяти в сетке
21
22 end:
23 mv a1, a0
24 li a0, 1
25 ecall
26 ret
```


start.s × | main.s × | **Nod.s** ×

```
1 .text
2 Nod:
3 .globl Nod
4 mv t0,a0
5 mv t1,a1
6 li t2, -1
7
8 li a2, 0      #count
9 li a3, 0      #save
10
11 smallest:
12 bgeu a2, t1, finish
13 lw a4, 0(t0) #firstElement 10
14 bge a4, zero, positiveOne
15 mul a4, a4, t2
16 positiveOne:
17 beq a2, zero, addSmallest
18 bgeu a3,a4, addSmallest
19 addi t0, t0, 4
20 addi a2, a2, 1
21 j smallest
22 addSmallest:
23 mv a3, a4
24 addi t0, t0, 4
25 addi a2, a2, 1
26 j smallest
27 finish:
28
29 mv a5,a3 #smallestInList
30 addi a5,a5,1
```

start.s ×

main.s ×

Nod.s ×

```
30 addi a5,a5,1
31
32 restart:
33 addi a5,a5,-1
34 mv t0,a0
35 mv t1,a1
36 li a2, 0      #count
37 li a3, 0      #save
38
39 nextNumber:
40 lw a4, 0(t0) #firstElement 10
41 bge a4, zero, positiveTwo
42 mul a4, a4, t2
43 positiveTwo:
44 beq a2,t1,end
45
46 addition:
47 add a3,a3,a5
48 beq a4,a3, possible
49 bgeu a3,a4, restart
50 j addition
51
52 possible:
53 li a3, 0      #save
54 addi t0,t0,4
55 addi a2,a2,1
56 j nextNumber
57 end:
58 mv a0,a5
59 ret
```

7. Работа программы 2

Программа выполнена при таких входных данных:

```
1 .rodata
2 array_length:
3 .word 5
4 .data
5 array:
6 .word -10, 5, 20, -60, 200
7 .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	zero	x0	0x00000000
<input type="checkbox"/>	0x00010004	0x00830067	jalr x0, x6, 8	jalr x0, x6, 8	ra	x1	0x00010010
<input type="checkbox"/>	0x00010008	0x00000317	auipc x6, 0	call main	sp	x2	0xbfffffff0
<input type="checkbox"/>	0x0001000c	0x014300e7	jalr x1, x6, 20	call main	gp	x3	0x10008000
<input type="checkbox"/>	0x00010010	0x00050593	addi x11, x10, 0	mv a1, a0	tp	x4	0x00000000
<input type="checkbox"/>	0x00010014	0x01100513	addi x10, x0, 17	li a0, 17	t0	x5	0x0001010c
<input type="checkbox"/>	0x00010018	0x00000073	ecall	ecall	t1	x6	0x00000005
<input type="checkbox"/>	0x0001001c	0x00000517	auipc x10, 0	la a0, array	t2	x7	0xffffffff
<input type="checkbox"/>	0x00010020	0x0dc50513	addi x10, x10, 220	la a0, array	a0	x8	0x00000000

Integer (X) Floating (F)

Console

5

Ответ: a5 x15 0x00000005