Санкт-Петербургский политехнический университет Петра Великого

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе №3**

**Дисциплина**: Низкоуровневое программирование

**Тема**: Программирование RISK-V

Выполнил студент гр. 3530901/90004 М. Д. Балкин

(подпись)

Преподаватель А. О. Алексюк

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

Оглавление

[1. Техническое задание 3](#_Toc71904913)

[2. Метод Решения 3](#_Toc71904914)

[3. Программа на языке ассемблера RISC-V 3](#_Toc71904915)

[3.1 Инициализация 3](#_Toc71904916)

[3.2 Программа 4](#_Toc71904917)

[3.3 Данные 5](#_Toc71904918)

[3.4 Результаты 5](#_Toc71904919)

[4. Подпрограмма и тестовая программа 5](#_Toc71904920)

[4.1 Запуск подпрограмм: 5](#_Toc71904921)

[4.2 Данные 8](#_Toc71904922)

[4.3 Ответ 8](#_Toc71904923)

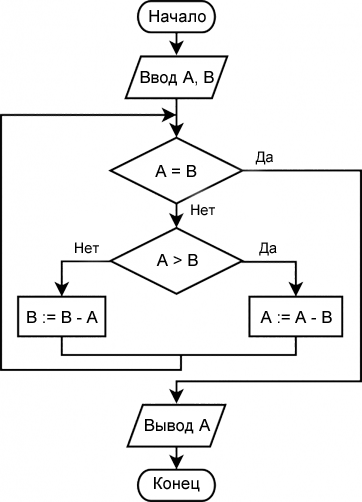
[5. Вывод 8](#_Toc71904924)

# 1. Техническое задание

Определение НОД всех элементов массива.

# 2. Метод Решения

Используя алгоритм Евклида



будем искать НОД массива таким образом:

НОД (НОД (1 элемент массива, 2 элемент массива), 3 элемент массива)) и так далее.

# 3. Программа на языке ассемблера RISC-V

# 3.1 Инициализация

* Для обхода нам понадобится счетчик, расположим его в ячейке a4, сразу же отняв от него 2(т.к. первые два элемента берутся вне цикла)
* В a3(регистр для ответа) положим нулевой элемент массива
* В t1 положим первый элемент массива, это будет его рабочая ячейка

# 3.2 Программа

.text

start:

.globl start

lw a4, arr\_length # счетчик

addi a4, a4, -2 # уменьшили счетчик

la a5, arr # указатель на 0й элем массива

lw a3, 0(a5) # a3 = arr[0]

addi a5, a5, 4 # указатель перешел к сл элем

lw t1, 0(a5) # t1 = arr[1]

loop\_start:

beq a3, t1, next # a3 == t1 -> next

bgeu a3, t1, loop1 # a3 > t1 -> loop1

bgeu t1, a3, loop2 # t1 > a3 -> loop2

loop1: #1ое > 2го

sub a3, a3, t1

jal zero, loop\_start

loop2: #2ое > 1го

sub t1, t1, a3

jal zero, loop\_start

next: #нод найден

addi a5, a5, 4 # указатель перешел к сл элем

lw t1, 0(a5)

addi a4, a4, -1

bltz a4, loop\_exit

jal zero, loop\_start

loop\_exit:

li a0, 10

ecall

.rodata

arr\_length:

.word 6

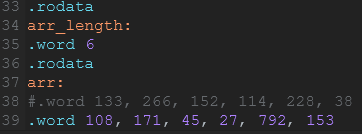
.rodata

arr:

#.word 133, 266, 152, 114, 228, 38

.word 108, 171, 45, 27, 792, 153

# 3.3 Данные



# 3.4 Результаты



Как видно из рисунка выше, в регистре a3 появился верный ответ – НОД чисел массива.

# 4. Подпрограмма и тестовая программа

# 4.1 Запуск подпрограмм:

Тестовая программа:

.text

setup:

.globl setup

call Nod\_main

finish:

li a0, 10

ecall

Подпрограмма:

.text

Nod\_main:

.globl Nod\_main

la a5, arr # для передачи sub

lw a4, arr\_length # для передачи sub

addi sp, sp, -16 #делаем для того чтобы не было зацикливания и сохраняем

sw ra, 12(sp) # загружаем в память

call NOD\_sub

lw ra, 12(sp)

addi sp, sp, 16 # возвращаем все обратно

ret # return

.rodata

arr\_length:

.word 6

.rodata

arr:

.word 133, 266, 152, 114, 228, 38

#.word 108, 171, 45, 27, 792, 153

Функционал:

.text

NOD\_sub:

.globl NOD\_sub

addi a4, a4, -2 # уменьшили счетчик

lw a3, 0(a5) # a3 = arr[0]

addi a5, a5, 4 # указатель перешел к сл элем

lw t1, 0(a5) # t1 = arr[1]

loop\_start:

beq a3, t1, next # a3 == t1 -> next

bgeu a3, t1, loop1 # a3 > t1 -> loop1

bgeu t1, a3, loop2 # t1 > a3 -> loop2

loop1: #1ое > 2го

sub a3, a3, t1

jal zero, loop\_start

loop2: #2ое > 1го

sub t1, t1, a3

jal zero, loop\_start

next: #нод найден

addi a5, a5, 4 # указатель перешел к сл элем

lw t1, 0(a5)

addi a4, a4, -1

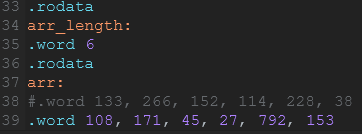
bltz a4, loop\_exit

jal zero, loop\_start

loop\_exit:

ret

# 4.2 Данные



# 4.3 Ответ



Как видно из рисунка выше, в регистре a3 появился верный ответ – НОД чисел массива.

# 5. Вывод

Разработаны две программы для вычисления НОД элементов массива для RISC-V, одна из которых использует вызов подрограмм.