Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчет по лабораторной работе №2**

**Дисциплина:** Низкоуровневое программирование

**Тема:** RISC-V

Выполнил

студент гр. 5130901/20001 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Жуков И.Д.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Коренев Д.А.

(подпись)

« 22 » ноября 2023 г.

Санкт-Петербург   
2023

**Оглавление**

[**ПОСТАНОВКА ЗАДАЧИ** 3](#_Toc64232757)

[**ПРЕДЛОЖЕННОЕ РЕШЕНИЕ** 4](#_Toc64232758)

**ПОСТАНОВКА ЗАДАЧИ**

1. Разработать программу под систему команд RISC-V, которая реализует определение k-й порядковой статистики.

**ПРЕДЛОЖЕННОЕ РЕШЕНИЕ**

Для решения поставленной задачи необходим массив чисел и натуральное число k, меньшее или равное длине массива и большее нуля. В результате выполнения программы ответ будет напечатан в командной строке. Для определения k-й порядковой статистики нужно отсортировать все числа в массиве с помощью сортировки пузырьком.

Наша программа на языке асемблер будет содержать 6 меток кода и 3 метки данных.

1. Метка **\_\_start**

С этой метки начинается выполнение нашей программы. В регистре а2 хранится итерируемое значение для внешнего цикла. В регистре а4 – длина массива. В а5 – адрес 0 элемента массива. В а6 – длина массива минус 1.

\_\_start:

li a2, 0 #inside a2: iterable var = i

la a4, array\_length

lw a4, 0 (a4) #inside a4: int array\_len value

la a5, array #inside a5: address of 0's array element

addi a6, a4, -1 #inside a6: int array\_len value - 1

1. Метка **outer\_loop**

Здесь мы принимаем решение: продолжать внешний цикл или нет. В регистр а3 будет храниться итерируемое значения для внутреннего цикла. Во временном регистре t5 будет храниться значение, до которого будет итерироваться значение регистра a3.

1. Метра **inner\_loop**

outer\_loop:  
bgeu a2, a6, outer\_loop\_exit # if( a2 >= a3 ) goto outer\_loop\_exit

li a3, 0 #inside a3: iterable var = j

sub t5, a6, a2

addi a2, a2, 1

Во внутреннем цикле мы должны сравнивать числа и менять их местами, если arr[i] > arr[i+1]. В регистр t0 сохраняем адрес i элемента массива. В регистр t1 сохраняем адрес i + 1 элемента массива. В t3 регистр сохраняем значение i элемента массива. В регистр t4 сохраняем значение i + 1 элемента массива. Если t3 > t4, то выполняем код, соответствующий метке **swap**

inner\_loop:

bgeu a3, t5, inner\_loop\_exit

slli t2, a3, 2 # t2 = j \* 4

add t0, a5, t2 # t0 = address of 0's element + j; arr[i] element

addi t1, t0, 4 # t address of arr[i + 1] element

addi a3, a3, 1

lw t3, 0 (t0) # array[i]

lw t4, 0 (t1) # array[i + 1]

bgeu t3, t4, swap

jal zero, inner\_loop

1. Метка **swap**

Записываем на место i элемента i + 1 элемент. На место i + 1 элемента i элемент.

swap:

sw t3, 0 (t1) # array[i] = t1

sw t4, 0 (t0) # array[i-1] = t0

jal zero, inner\_loop

1. Метка **inner\_loop\_exit**

Безусловный переход к метке outer\_loop

inner\_loop\_exit:

jal zero, outer\_loop

1. Метка **outer\_loop\_exit**

Сохраняем k в регистр t0. Вычитаем 1, т.к. k > 0. Прибавляем полученное значение к адресу 0 элемента массива a5 и сохраняем в регистр t0. Записываем полученное значение в регистр a1 для дальнейшей печати.

outer\_loop\_exit:

la t0, k

lw t0, 0(t0)

addi t0, t0, -1

slli t0, t0, 2

add t0, a5, t0

lw t0, 0 (t0)

li a0, 1 # command to print in result in console

addi a1, t0, 0

ecall

li a0, 10

ecall

**Хранение данных**

Значение k и длина массива – неизменяемые значения, поэтому read only.

.rodata

k: .word 6

array\_length: .word 6

.data

array:

.word 11, 10, 2, 1, 10101, 3