Санкт–Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

**Кафедра «Информатика и вычислительная техника»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

Дисциплина: Низкоуровневое программирование

Тема: программирование на RISC-V

Выполнил

студент гр.3530901/10005 Б. А. Терехов

Руководитель Коренев Д.А.

«1» декабря 2022 г.

Санкт–Петербург

2022

Санкт–Петербургский государственный политехнический университет

**Оглавление**

[1. Техническое задание 3](#_Toc121132552)

[2. Метод решения 3](#_Toc121132553)

[Описание алгоритма: 3](#_Toc121132554)

[3. Программа 1 3](#_Toc121132555)

[Исходные данные: 3](#_Toc121132556)

[Код программы 3](#_Toc121132557)

[Результат программы: 4](#_Toc121132558)

[4. Программа 2 (ABI) 5](#_Toc121132559)

[Исходные данные: 5](#_Toc121132560)

[Код программы *sort.s* 5](#_Toc121132561)

[Код программы *main.s* 6](#_Toc121132562)

[Код программы *setup.s* 6](#_Toc121132563)

[Результат программы: 6](#_Toc121132564)

[5. Вывод 7](#_Toc121132565)

1. Техническое задание

Написать программу сортировка выбором массива чисел in-place (на месте старого массива).

2. Метод решения

Описание алгоритма:

За каждый проход по массиву выбрать минимальный элемент и поменять его местами с первым элементом в еще не отсортированном участке массива, тем самым уменьшив длину этого участка на один, и так до тех пор, пока не будут отсортированы все элементы.

3. Программа 1

Исходные данные:

Массив [1, 5, 3, 6, 9, 2, 4, 7, 10, 8]

Код программы

|  |
| --- |
| .text  \_\_start:  .globl \_\_start  lw a3, array\_length # a3 — длина<массива>  la a4, array # ad = anpecc 0гo— элемент массива>  slli a6, a3, 2 # a6 = a3\*4  add a6, a4, a6 # a6 = ad+a3ed  addi a6, a6, -4 #last el  loop:  bgeu zero, a3, loop\_exit # if( 0 >= a3 ) goto loop\_exit  li a2, 1 # ad=1  addi a7, a4, 0 #max place  lw t2, 0(a7) #max = 1 element  add a5, a4, zero #a5 — first element  loop2:  bgeu a2, a3, loop2\_exit # if( a2 > a3 ) goto loop\_exit  addi a5, a5, 4 # a5 = a5 + 4  addi a2, a2, 1 # a2 4-1  lw t3, 0(a5)  bgeu t2, t3, loop2 # if( t2 >= t3 ) goto loop\_exit  addi a7, a5, 0 #max place  lw t2, 0(a7) #max  jal zero, loop2  loop2\_exit:  addi a3, a3, -1 # ab = ad - 4 len -1  lw t1, 0(a6) # tl = array[last]  lw t0, 0(a7) # tO = array[i]  sw t1, 0(a7) # array[i] — tl  sw t0, 0(a6) # array[last] — t0  addi a6, a6, -4 # a6 ~ a6 + (4) a6—4 last element -1    jal zero, loop # goto loop  loop\_exit:  finish:  li a0, 10  li a1, 0  ecall  .rodata  array\_length:  .word 10  .data  array:  .word 1, 5, 3, 6, 9, 2, 4, 7, 10, 8 |

Результат программы:

4. Программа 2 (ABI)

Исходные данные:

Массив [1, 5, 3, 6, 9, 2, 4, 7, 10, 8]

Код программы *sort.s*

|  |
| --- |
| #sort  .text  sort:  .globl sort  slli a6, a3, 2 # a6 = a3\*4  add a6, a4, a6 # a6 = ad+a3ed  addi a6, a6, -4 #last el  loop:  bgeu zero, a3, loop\_exit # if( 0 >= a3 ) goto loop\_exit  li a2, 1 # ad=1  addi a7, a4, 0 #max place  lw t2, 0(a7) #max = 1 element  add a5, a4, zero #a5 — first element  loop2:  bgeu a2, a3, loop2\_exit # if( a2 > a3 ) goto loop\_exit  addi a5, a5, 4 # a5 = a5 + 4  addi a2, a2, 1 # a2 4-1  lw t3, 0(a5)  bgeu t2, t3, loop2 # if( t2 >= t3 ) goto loop\_exit  addi a7, a5, 0 #max place  lw t2, 0(a7) #max  jal zero, loop2  loop2\_exit:  addi a3, a3, -1 # ab = ad - 4 len -1  lw t1, 0(a6) # tl = array[last]  lw t0, 0(a7) # tO = array[i]  sw t1, 0(a7) # array[i] — tl  sw t0, 0(a6) # array[last] — t0  addi a6, a6, -4 # a6 ~ a6 + (4) a6—4 last element -1    jal zero, loop # goto loop  loop\_exit:  ret |

Код программы *main.s*

|  |
| --- |
| # main.s  .text  main:  .globl main  addi sp, sp, -16 # выделение памяти в стеке  sw ra, 12 ( sp ) # сохранение ra  la a0, array #  lw a1, array\_length # sort ( array, array\_length );  call sort  li a0, 0  lw ra, 12 ( sp ) # восстановление ra  addi sp, sp, 16 # освобождение памяти в стеке  ret # return 0;  .rodata  array\_length:  .word 10  .data  array:  .word 1, 5, 3, 6, 9, 2, 4, 7, 10, 8 |

Код программы *setup.s*

|  |
| --- |
| .text  \_\_start:  .globl \_\_start  call main  finish:  mv a1, a0 # a1 = a0  li a0, 17 # a0 = 17  ecall #выход с кодом завершения |

Результат программы:

5. Вывод

В результате выполнения работы был получен опыт программирования на RISC-V, а также был написан алгоритм сортировки выбором массива чисел in-place.