Санкт-Петербургский политехнический университет Петра Великого

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе №3**

**Дисциплина**: Низкоуровневое программирование

**Тема**: программирование на языке riscV

Выполнил студент гр. 3530901/90003 В.Д. Коротких

(подпись)

Преподаватель А. О. Алексюк

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

Оглавление

[1. Техническое задание 3](#_Toc68519279)

[2. Метод решения 3](#_Toc68519280)

[3. Программа KSearchPrg 3](#_Toc68519281)

[4. Работа программы KSearchPrg 5](#_Toc68519282)

[5. Программа KSearch 5](#_Toc68519283)

[6. Работа программы KSearch 7](#_Toc68519284)

# Техническое задание

* Разработать программу на языке ассемблера RISC-V определение K-ой порядковой статистики in-place, отладить программу в симуляторе VSim/Jupiter.Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.
* Выделить определение K-ой порядковой статистики in-place в подпрограмму, организованную в соответствии с ABI, разработать использующую ее тестовую программу. Адрес обрабатываемого массива данных и другие значения передавать через параметры подпрограммы в соответствии с ABI. Тестовая программа должна состоять из инициализирующего кода, кода завершения, подпрограммы main и тестируемой подпрограммы.

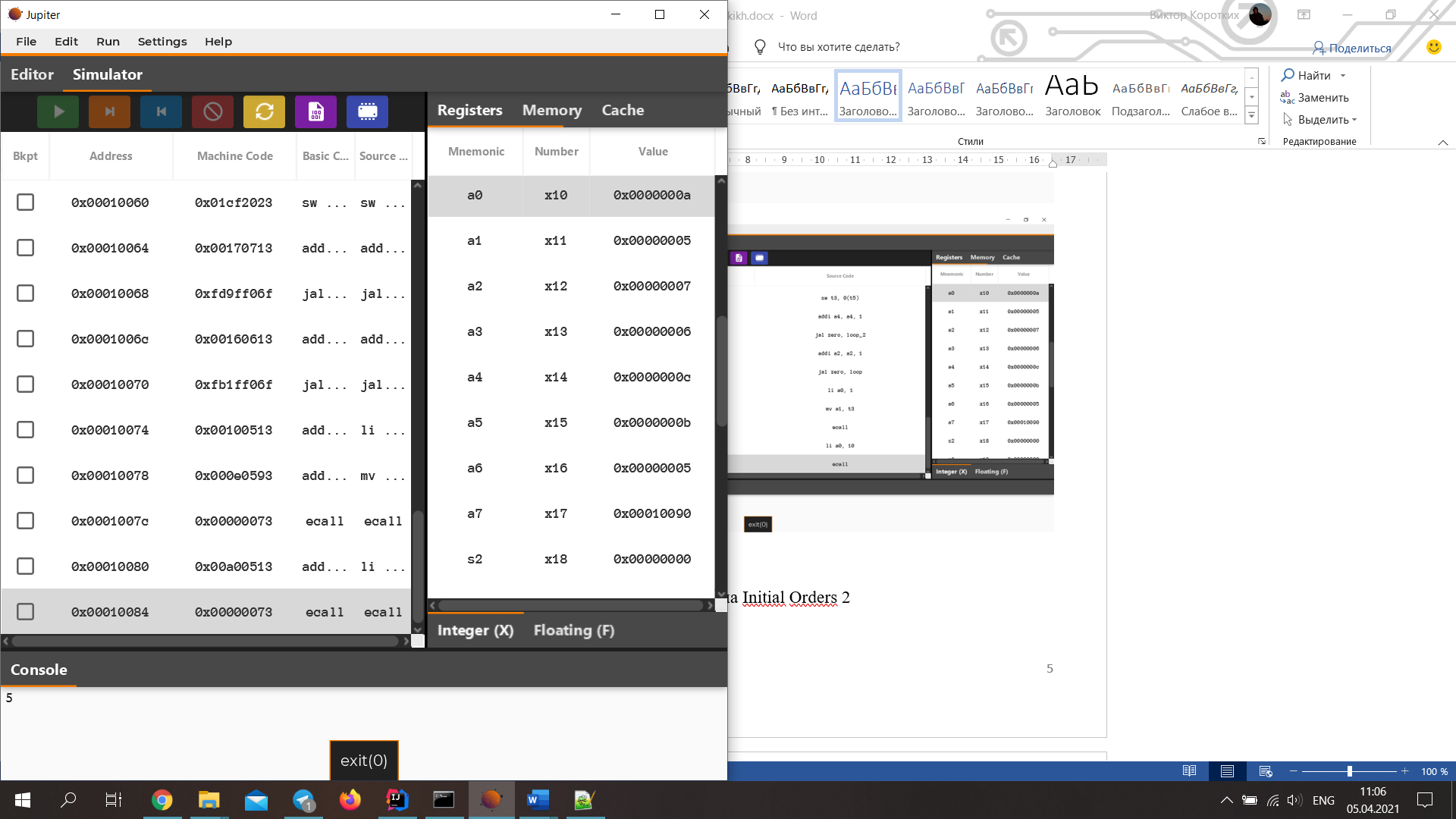
# Метод решения

Поиск k-ой порядковой статистики выполнен на основе сортировки обменом: упорядочиваются первые k элементов, после чего k-ый (максимальный на текущий момент) элемент выводится в консоль.

# Программа KSearchPrg

|  |
| --- |
| #kSearchPrg  #Техническое задание - Определение K-ой порядковой статистики in-place  .globl \_\_start  #Входные данные:  .rodata  array\_length: #длина массива  .word 11  k: # значение к  .word 6  .data  array: #массив  .word 0,3,7,9,2,1,10,8,4,5,6  .text #Секция кода  \_\_start:  kSearch:  #for var i:=0 to (k-1) do  la a3, k #  lw a3, 0(a3) # a3 := k  li a2, 0 # a2 = 0    la a7, array #a7 = a[0]    loop:  bgtu a2, a3, loop\_exit #if (a2 > a3) goto loop\_exit    #t5:=addr(a[i]);  slli t5, a2, 2 # t5 = a2 << 2 = a2 \* 4  add t5, a7, t5 # t5 = a7 + t5 = a7 + a2 \* 4    #min:=a[i];  lw t3, 0(t5) #t3 := array[i]    #for var j:=i+1 to (a.Length) do  la a5, array\_length #  lw a5, 0(a5) # a5 := array\_length  addi a4, a2, 1 # a4 = a2 + 1  loop\_2:  bgtu a4, a5, loop\_2\_exit #if (a4 > a5) goto loop\_2\_exit    #t6 := addr(a[j])  slli t6, a4, 2 # t6 = a4 << 2 = a4 \* 4  add t6, a7, t6 # t6 = a7 + t6 = a7 + a4 \* 4  lw a6, 0(t6) # a6 := a[j]    #if (a[j]<min) then  bleu t3, a6, endif #if (min <= a[j]) goto endif  mv t4, t3 #pMin := min;  lw t3, 0(t6) #min:=a[j];  sw t4, 0(t6) #a[j]:=pMin;  endif:  sw t3, 0(t5) #a[i]:=min; (t5 = addr(a[i])    addi a4, a4, 1 # a4 += 1  jal zero, loop\_2 # goto loop\_2  loop\_2\_exit:  addi a2, a2, 1 # a2 += 1  jal zero, loop # goto loop  loop\_exit:  #result = min;  # prints the result in t3  li a0, 1  mv a1, t3  ecall  kSearch\_exit:    finish:  li a0, 10 # x10 := 10  ecall # ecall при значении x10 = 10 => останов |

# Работа программы KSearchPrg



# Программа KSearch

Основная программа

|  |
| --- |
| # main.s  .global \_\_start  #Определим данные:  .rodata  array\_length:  .word 11  k:  .word 6  #Секция изменяемых данных:  .data  array:  .word 0,3,7,9,2,1,10,8,4,5,6    .text  \_\_start:  addi sp, sp, -16 # выделение памяти в стеке  sw ra, 12(sp) # сохранение ra  la a0, array # \  lw a1, array\_length # \  lw a2, k # } kSearchSub(array, array\_length, k);  call KSearchSub # /    lw ra, 12(sp) # восстановление ra  addi sp, sp, 16 # освобождение памяти в стеке  li a1, 0  li a0, 17  ecall |

Подпрограмма для поиска k-й порядковой статистики

|  |
| --- |
| # kSearch.s  # подпрограмма для поиска к-ой порядковой статистики  .global KSearchSub  .text  KSearchSub:  #Arguments    #Addr(a[0]) -> a0  #array\_length -> a1  #k -> a2  kSearch:  #for var i:=0 to (k-1) do  li a6, 0 # a6 := 0    loop:  bgtu a6, a2, loop\_exit #if (a6 > a2) goto loop\_exit    #t5:=addr(a[i]);  slli t5, a6, 2 # t5 := a6 << 2 = a6 \* 4  add t5, a0, t5 # t5 := a0 + t5 = a0 + a6 \* 4    #min:=a[i];  lw t3, 0(t5) #t3 := array[i]    #for var j:=i+1 to (a.Length-1) do  addi a4, a6, 1 # a4 := a6 + 1  loop\_2:  bgtu a4, a1, loop\_2\_exit #if (a4 > a1) goto loop\_2\_exit    #t6 := addr(a[j])  slli t6, a4, 2 # t6 := a4 << 2 = a4 \* 4  add t6, a0, t6 # t6 := a0 + t6 = a0 + a4 \* 4  lw t0, 0(t6) # t0 := a[j]    #if (a[j]<min) then  bleu t3, t0, endif #if (min <= a[j]) goto endif  mv t4, t3 #pMin := min;  lw t3, 0(t6) #min := a[j];  sw t4, 0(t6) #a[j] := pMin;  endif:  sw t3, 0(t5) #a[i] := min; (t5 = addr(a[i])    addi a4, a4, 1 # a4 += 1  jal zero, loop\_2 # goto loop\_2  loop\_2\_exit:  addi a6, a6, 1 # a6 += 1  jal zero, loop # goto loop  loop\_exit:  #result = min;  # prints the result in t3  li a0, 1  mv a1, t3  ecall  kSearch\_exit:    ret # jalr zero, ra, 0 |

# Работа программы KSearch

