Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина**: Низкоуровневое программирование

**Тема**: RISC-V

Выполнила студентка гр. 3530901/90004

Д.В. Кузнецова

(подпись)

Преподаватель А.О. Алексюк

(подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Санкт-Петербург

2021

**Задача:**

* Разработать программу на языке ассемблера RISC-V, реализующую определенную вариантом задания функциональность, отладить программу в симуляторе Jupiter.
* Выделить определенную вариантом задания функциональность 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.

**Вариант 4:**

Реализовать алгоритм сортировки вставкой.

**Описание работы**

**Алгоритм:**

Сортировка вставками - алгоритм сортировки, в котором элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов.

**Алгоритм на языке Python:**

arr = [1, 3, 0, 2, 4, 6, 5]

for i in range(1, len(arr)):

item\_to\_insert = arr[i]

j = i – 1

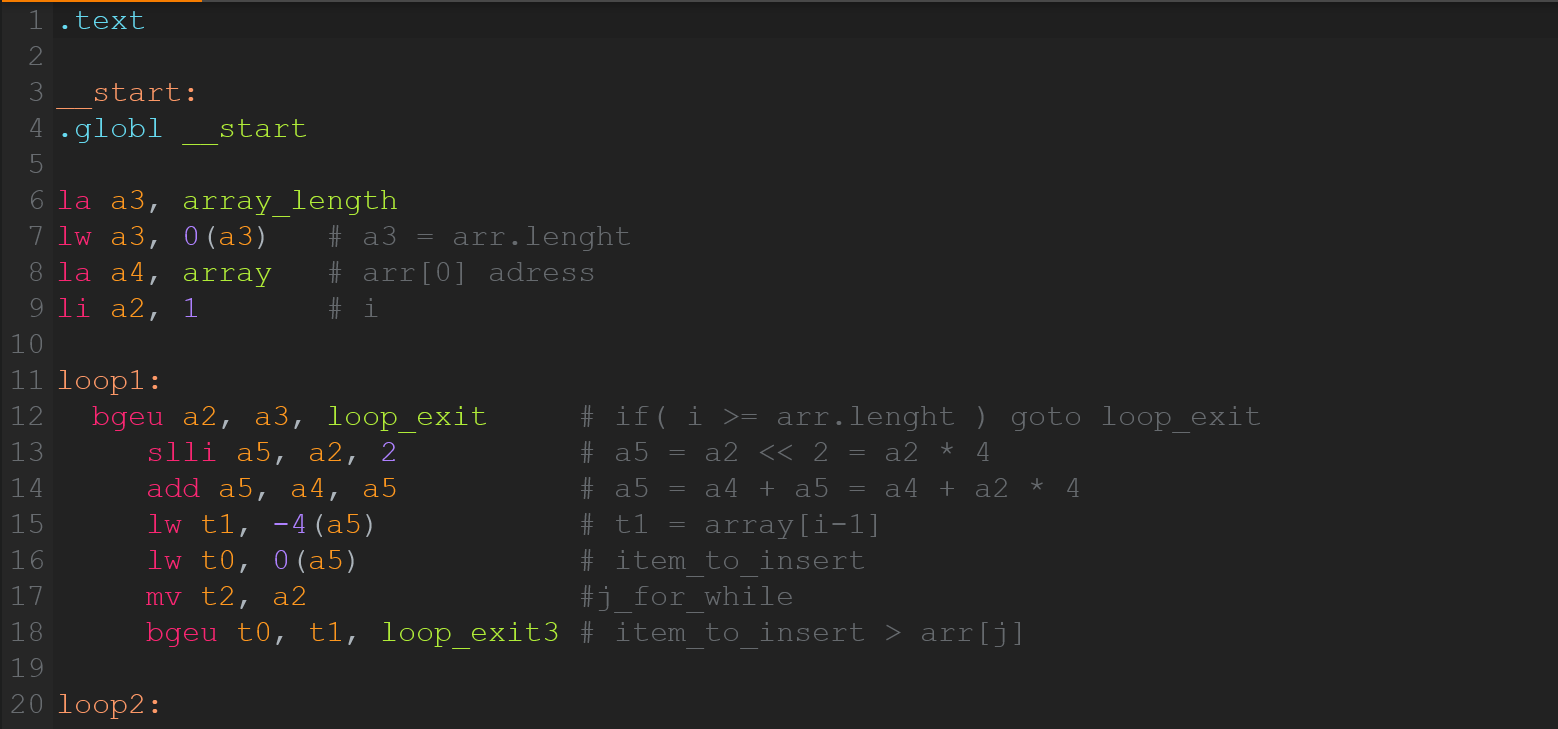
while j >= 0 and arr[i] > item\_to\_insert:

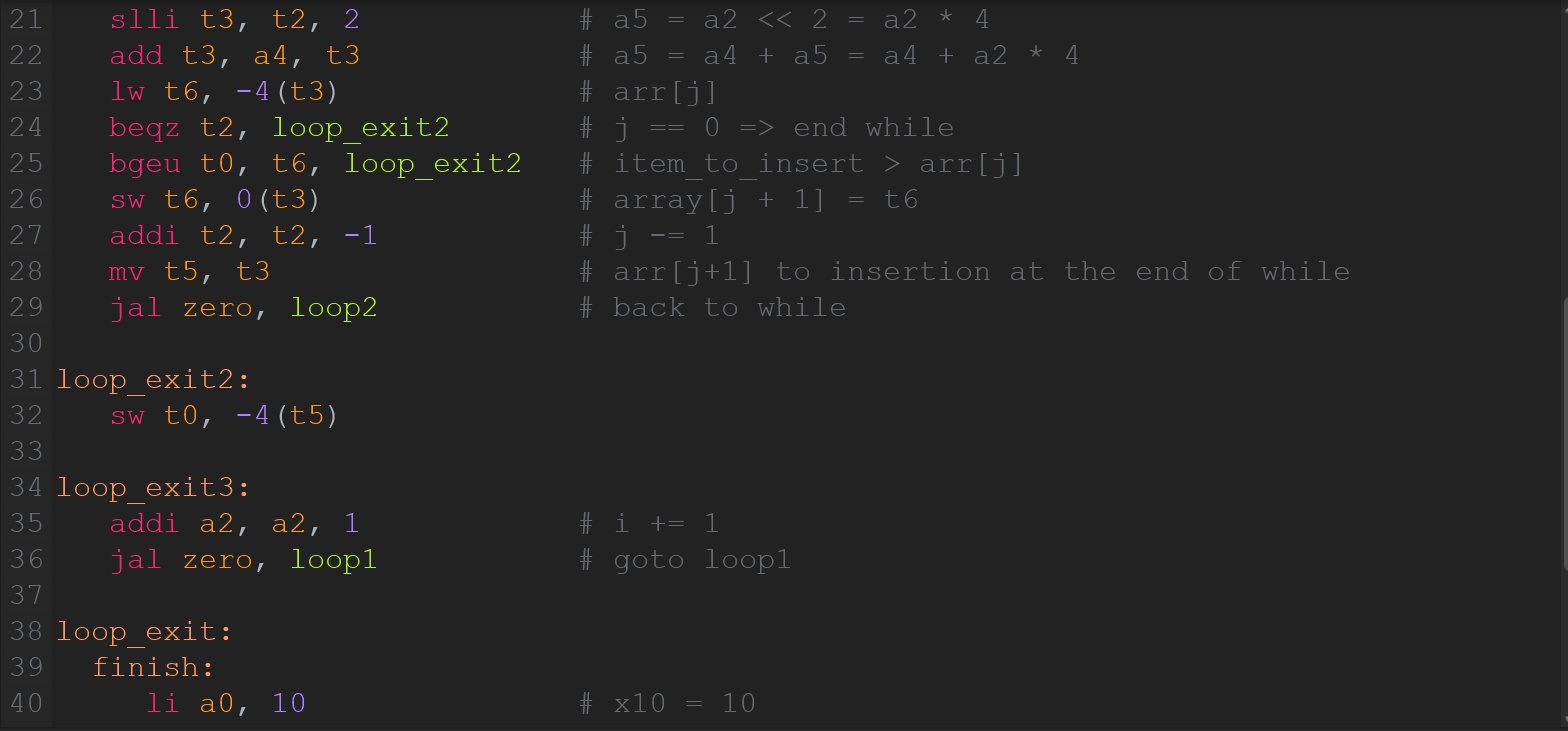
arr[i + 1] = arr[i]

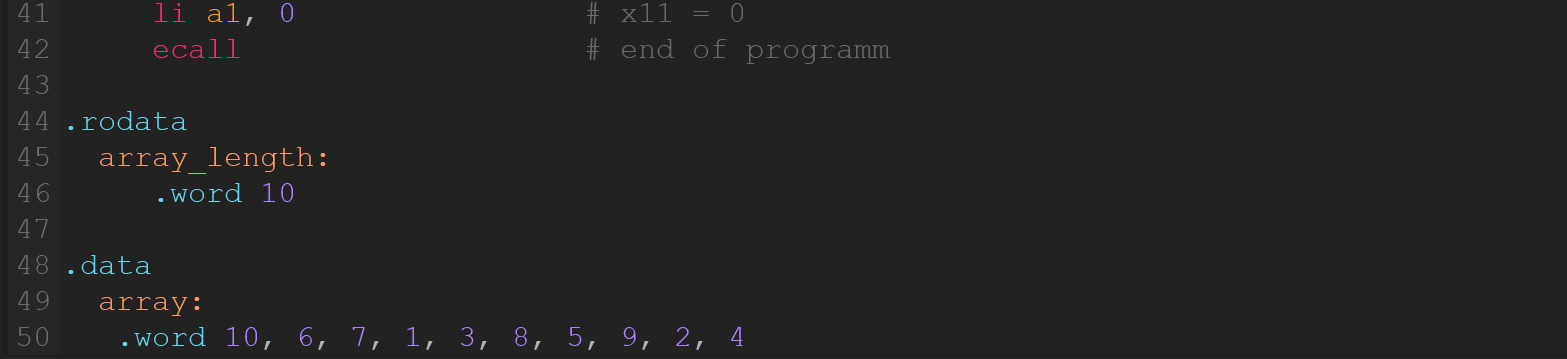
j -=1

arr[i + 1] = item\_to\_insert

**Код программы**







**Результаты работы алгоритма:**

|  |
| --- |
| arr = [10, 6, 7, 1, 3, 8, 5, 9, 2, 4] |
|  |

**Описание программы:**

“text” – указатель на основную часть кода,

“start” – точка начала выполнения программы. В строках 4-7 написаны псевдоинструкции установки значений регистров.

a2 = i.

a3 = длинна массива.

a4 = адрес первого элемента массива.

8 - Вход в цикл for. Инструкция в строке служит условием выхода из цикла for (i >= длинна массива), если true, переходим к 33 строчке. Иначе идем далее по программе.

9-11 - Инструкции, отвечающие за вычисление адреса arr[i] при помощи сдвига влево i-ого числа и прибавления этого выражения к нулевому элементу массива. Значение заносится в регистр a5.

12-14 - В регистры t1, t0, t2 записывается arr[i - 1], arr[i] и j = i соответственно.

15 проверяется условие (item\_to\_insert > arr[j]), если true – мы переходим к строке 29, где увеличиваем значение i и переходим к следующей итерации. Иначе заходим в цикл While.

17-19 - получаем значение arr[j] и записываем его в регистр t6

20-21 – Условия цикла while: j = 0, item\_to\_insert > arr[j], если хоть одно из условий true – выходим из while в строку 27.

22-25 – Записываем в arr[j + 1] значение элемента arr[j], уменьшает j на единицу и записываем в регистр t5 ссылку на элемент arr[j + 1] для последующей вставки числа в это значение после выхода из цикла. Далее прыгаем в строку 16.

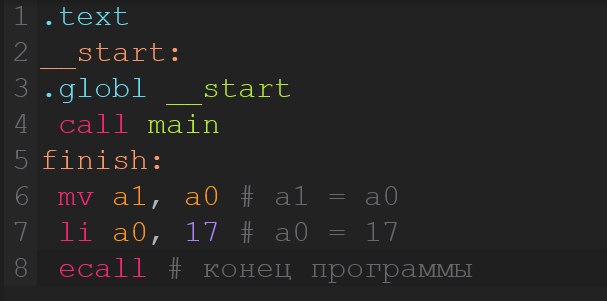
26-30 – Вставляем элемент для вставки (item\_to\_insert) в ячейку arr[j + 1], увеличиваем i на единицу и переходим к следующей итерации обратно в строку 8.

33-35 – Завершение работы программы

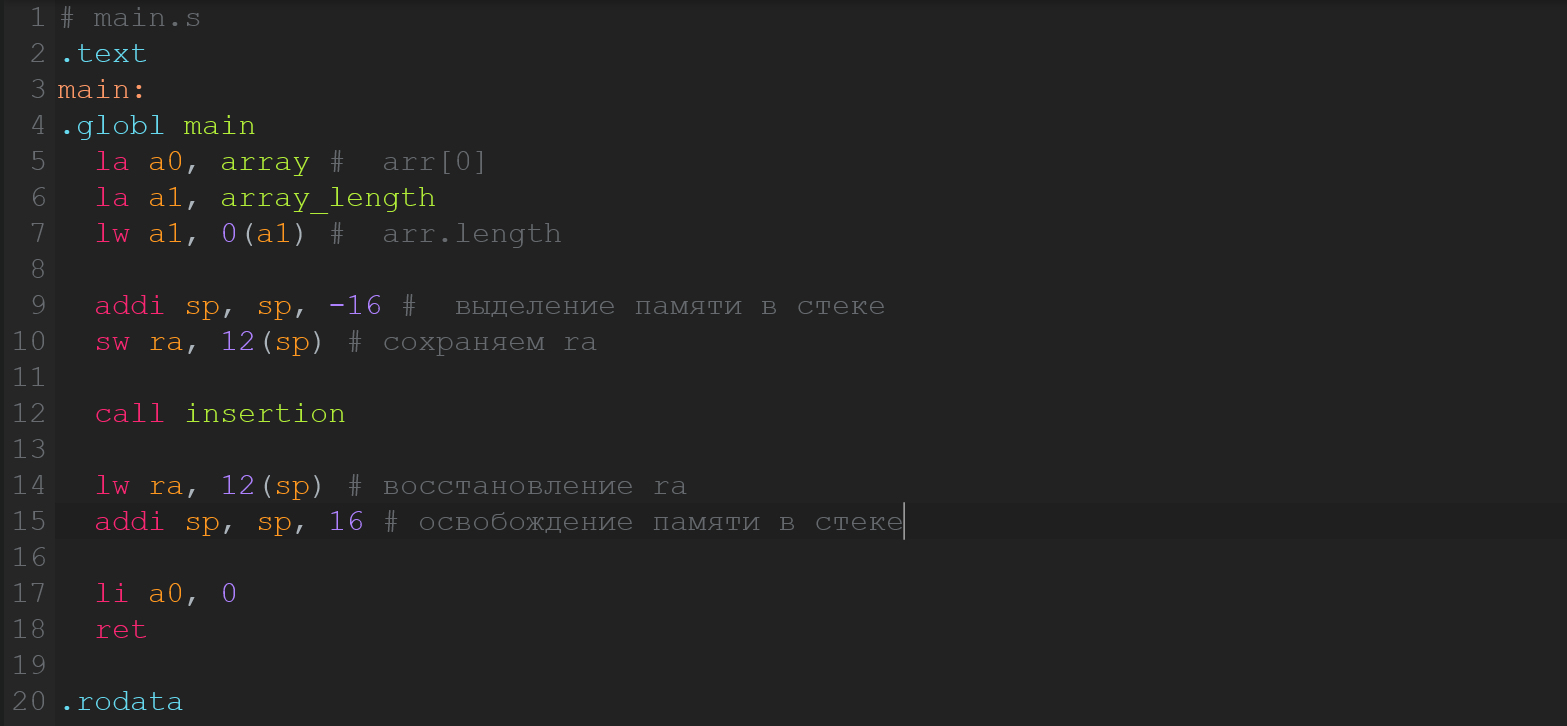
36-41 – Данные программы. .word - означает, что мы используем 32-битные слова (4 байта). То есть они занимают 4 восьмибитных секции. .rodata – константы, .data – переменные.

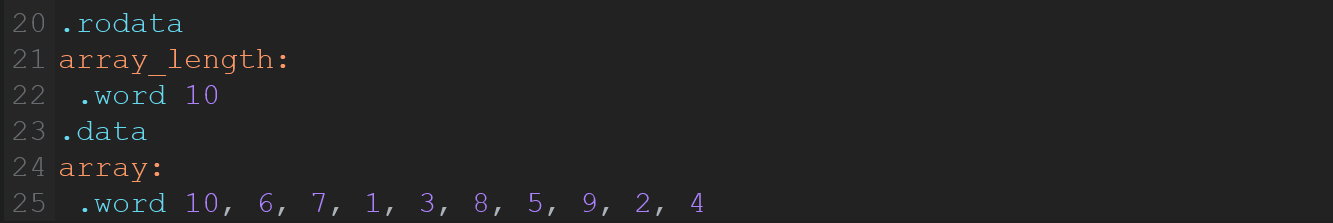
**Реализация подпрограммы**

**Setup:**

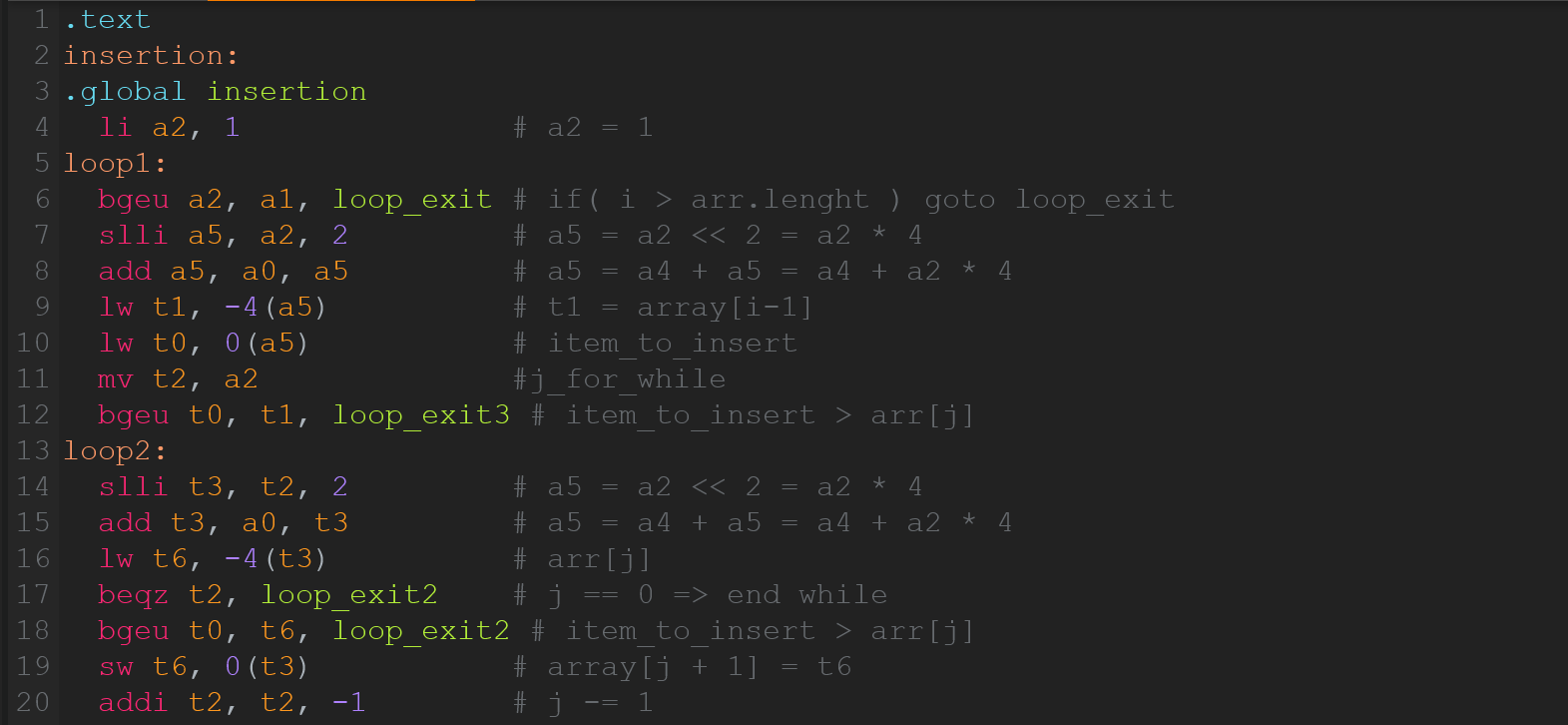


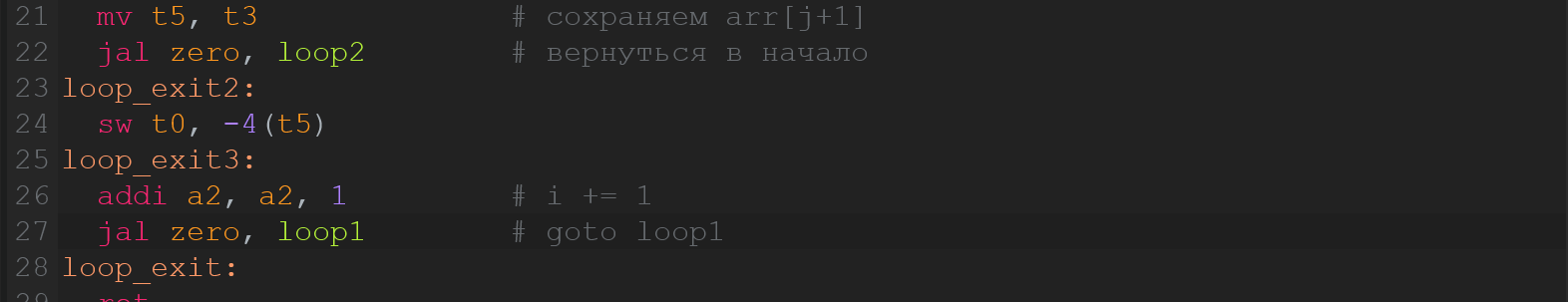
**Подпрограмма main:**





**Подпрограмма insertion:**





**Результат:**

|  |
| --- |
| arr = [10, 6, 7, 1, 3, 8, 5, 9, 2, 4] |
|  |

**Вывод:** В ходе лабораторной работы были получены навыки работы в симуляторе Jupiter и разработана программа на языке ассемблера RISC-V, реализующая сортировку массива вставками. Также, были изучены основные аспекты RISC-V.