

Постановка задачи

1. Изучить методические материалы, доступные на сайте курса.
2. Разработать программу на языке ассемблера RISC-V (<https://github.com/riscv/riscv-asm-manual/blob/master/riscv-asm.md>), реализующую определенную вариантом задания функциональность, отладить программу в симуляторе VSim/Jupiter (<https://github.com/andrescv/Jupiter/releases>). Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.
3. Выделить определенную вариантом задания функциональность в подпрограмму, организованную в соответствии с ABI (<https://github.com/riscv/riscv-elf-psabi-doc/blob/master/riscv-elf.md>), 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.

Варианты заданий

Используются адаптированные варианты заданий на лабораторную работу №2. Точная формулировка задания предлагается студентом и согласовывается преподавателем.

Симулятор VSim/Jupiter

Дистрибутивы симулятора: <https://github.com/andrescv/Jupiter/releases>.

Требования к отчету

Отчет должен содержать:

- развернутую формулировку задачи;
- текст программы с комментариями по 1-й части задания;
- текст подпрограммы и тестовой программы с комментариями по 2-й части задания;
- правила кодирования исходных данных и результатов;
- руководство программиста.

Руководство программиста должно содержать пояснения, необходимые для понимания организации программы (подпрограммы).