

Отчёт по лабораторной работе № 2

Дисциплина: Низкоуровневое программирование

Тема: Программирование RISC-V

Вариант: 4

Выполнил студент гр. 3530901/00002 _____ Н.А. Васишин
(подпись)

Принял преподаватель _____ Д.С. Степанов
(подпись)

“ ____ ” _____ 2021 г.

Санкт-Петербург

2021

Постановка задачи:

1. Изучить методические материалы, доступные на сайте курса.
2. Разработать программу на языке ассемблера RISC-V, реализующую определенную вариантом задания функциональность, 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.
3. 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. Тестовая программа должна состоять из инициализирующего кода, кода завершения, подпрограммы main и тестируемой подпрограммы.

Вариант:

Сортировка вставкой массива чисел in-place.

2. Код программы:

```
.text
start:
.globl start
la a3, array_length
lw a3, 0(a3)
la a4, array
li a2, 1
loop1:
bgeu a2, a3, loop_exit
slli a5, a2, 2
add a5, a4, a5
lw t1, -4(a5)
lw t0, 0(a5)
mv t2, a2
bgeu t0, t1, loop_exit3 [j]
loop2:
slli t3, t2, 2
```

```

add t3, a4, t3
lw t6, -4(t3)
beqz t2, loop_exit2
bgeu t0, t6, loop_exit2
sw t6, 0(t3)
addi t2, t2, -1
mv t5, t3
jal zero, loop2
loop_exit2:
sw t0, -4(t5)
loop_exit3:
addi a2, a2, 1
jal zero, loop1
loop_exit:
finish:
li a0, 10
li a1, 0
ecall
.rodata
array_length:
.word 10
.data
array:
.word 10,1,8,7,6,5,4,3,3,4

```

Данная программа осуществляет сортировку массива с входными значениями {10,1,8,7,6,5,4,3,3,4}. Результирующий массив таков {1,3,3,4,4,5,6,7,8,10}.

3. Код программы, тестовой программы и подпрограммы :

Программа:

```

.text
start:
.globl start
call subprog_main
finish:
mv a1, a0
li a0, 17
ecall

```

Тестовая программа:

```
.text
subprog_main:
.globl subprog_main
la a0, array
la a1, array_length
lw a1, 0(a1)

addi sp, sp, 16
sw ra, 12(sp)

call subprog

lw ra, 12(sp)
addi sp, sp, 16

li a0, 0
ret

.rodata
array_length:
.word 10
.data
array:
.word 10,1,8,7,6,5,4,3,3,4
```

Подпрограмма:

```
.text
subprog:
.globl subprog
li a2, 1
loop1:
bgeu a2, a1, loop_exit
slli a5, a2, 2
add a5, a0, a5
lw t1, -4(a5)
lw t0, 0(a5)
mv t2, a2
```

```

    bgeu t0, t1, loop_exit3
loop2:
    slli t3, t2, 2
    add t3, a0, t3
    lw t6, -4(t3)
    beqz t2, loop_exit2
    bgeu t0, t6, loop_exit2
    sw t6, 0(t3)
    addi t2, t2, -1
    mv t5, t3
    jal zero, loop2
loop_exit2:
    sw t0, -4(t5)
loop_exit3:
    addi a2, a2, 1
    jal zero, loop1
loop_exit:
    ret

```

Вывод:

В ходе лабораторной работы были изучены принципы программирования на RISC-V ассемблер, получены практические навыки построения программ.

Список используемых источников:

<https://github.com/riscv-non-isa/riscv-asm-manual/blob/master/riscv-asm.md>

<http://kspt.icc.spbstu.ru/course/lowlevelprog>