

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе № 3

Дисциплина: Низкоуровневое программирование

Тема: Программирование RISC-V

Вариант №14

Выполнил студент гр. 3530901/00002 _____ Чэнь Аосюань
(подпись)

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.

Варианты заданий

14) Определение наиболее часто встречающегося в массиве значения

Решение

```
for (int i = 0; i < length; i++) {  
    for (int y = 0; y < length; y++) {  
        if (elementI == elementY) {  
            timeNow++;  
        }  
    }  
    if (timeNow > timeMax) {  
        timeMax = timeNow;  
        timeMax = elementI;  
    }  
    timeMax = 0;  
}
```

(Рис 1 Решение в коде)

1-ий части задания

.text

start:

.globl start

la a2, length_arr

lw a2, 0(a3) #the length of arr

la a3, array #0th adress for array

li a4, 0#a4 time max

li a5, 0#a5 type max

li a6, 0# for (int i = 0)

loop1:

bgeu a6, a2, loop1_exit #(if i >= length exit)

slli t1, a6, 2

add t1, t1, a3#t1 i-th element of array, type now

li a7, 0#a7 for (int y = 0)

addi a6, a6, 1# i++

li t3, 0#t3 time now

loop2:

bgeu a6, a2, loop2_exit #(if y >= length exit)

slli t2, a7, 2

add t2, t2, a3#t2 y-th element of array, type now

addi t3, t3, 1# y++

bltu t2, t1, loop2

bltu t1, t2, loop2# if t1 = t2

addi t3, t3, 1#time now + 1

bgeu a4, t3, loop2 #if time max >= time now return loop2

sw t3, 0(a4)

sw t1, 0(a5)

jal zero, loop2

loop2_exit:

jal zero, loop1

```

        loop1_exit:
        finish:
        li a0, 10 # x10 = 10
        li a1, 0 # x11 = 0
        ecall
        .rodata
length_arr:
        .word 5
        .data
array:
        .word 1,1,2,3,4

```

2-ий части задания

```

# setup.s
.text
start:
.globl start
call main
finish:
mv a1, a0 # a1 = a0
li a0, 17
ecall

```

Тестовая программа

```

#main.s
.text
main:
.globl main
        addi sp, sp, -16 # выделение памяти в стеке
        sw ra, 12(sp) # сохраняем ra
        la a0, array # 0 - adress for array

```

```

lw a1, array_length # the length of arr
call function
li a0, 0
lw ra, 12(sp)
addi sp, sp, 16
ret # return 0;
.rodata
array_length:
.word 5
.data
array:
.word 1,1,2,3,4

```

Подпрограмма

```
#function.s
```

```
.text
```

```
function:
```

```
.globl function
```

```
#now in a0 – 0 - adress for array a3 - a0
```

```
# in a1 – the length of arr a2 - a1
```

```
li a4, 0#a4 time max
```

```
li a5, 0#a5 type max
```

```
li a6, 0# for (int i = 0 .....)
```

```
loop1:
```

```
bgeu a6, a1, loop1_exit #(if i >= length exit)
```

```
slli t1, a6, 2
```

```
add t1, t1, a0#t1 i-th element of array, type now
```

```
li a7, 0#a7 for (int y = 0 .....)
```

```
addi a6, a6, 1# i++
```

```

li t3, 0#t3 time now
loop2:
bgeu a6, a1, loop2_exit #(if y >= length exit)
slli t2, a7, 2
add t2, t2, a0#t2 y-th element of array, type now
addi t3, t3, 1# y++
bltu t2, t1, loop2
bltu t1, t2, loop2# if t1 = t2
addi t3, t3, 1#time now + 1
bgeu a4, t3, loop2 #if time max >= time now return loop2
sw t3, 0(a4)
sw t1, 0(a5)
jal zero, loop2
loop2_exit:
jal zero, loop1
loop1_exit:
ret

```

Описание работы

При использовании. Можно менять исходный массив(элементы и длинна) в часть data, Тествая программа и подпрограмма в отличии от программы использует a0 и a1 для сохранения адреса 0-ого элемента и для сохранения длины.

Список использованных источников

http://kspt.icc.spbstu.ru/media/files/2020/lowlevelprog/riscv_prgc.pdf

<https://github.com/andrescv/Jupiter>

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

http://kspt.icc.spbstu.ru/media/files/2020/lowlevelprog/riscv_subprgc.pdf