Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина**: Низкоуровневое программирование

**Тема**: RISC-V

Выполнил студент гр. 3530901/90004 Н.А. Пудов

(подпись)

Преподаватель А.О. Алексюк

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

**Вариант 8**

**Формулировка задачи**

Разработать программу на RISC-V c использованием функции main и подпрограммы по расчёту к-го члена Фибоначчи.

**Код программы.**

Файл fibonachi.s

.text

fibonachi:

.globl fibonachi

add t0, a0, zero #t0 = a0

la t1, numfib

sw t0, 0(t1)

la a3, numfib

lw a3, 0(a3) #numfib = 8

li a7, 2 #fib(1)=1 and fib(2)=1

li a4, 1 #first = 1

li a5, 1 #last = 1

li a6, 0 #ans = 0

li a2, 2

bge zero, a3, finish #if numfib <= 0 go to finish

bgeu a7, a3, case1 #if numfib <= 2 go to case1

jal zero, loop\_check

loop:

add a6, a4, a5 # ans = first + last

#сделать переприсваивание

add t0, a6, zero # t0 = ans

add t1, a5, zero # t1 = last

add a4, t1, zero # first = last

add a5, t0, zero # last = ans

addi a2, a2, 1 #a2+=1

bgeu a2, a3, finish # if (a2 >= a3) go to loop\_exit

loop\_check:

bltu a2 , a3, loop # if (a2 < a3) go to loop

case1:

addi a6, a6, 1 # a6+=1

jal zero, finish

finish:

#li a0, 1 #x10

addi a1, a6, 0 #a1 = ans

#ecall

ret

.data

numfib:

.word 0

Файл mainfunction.s

.text

main:

.globl main

# в a3 - номер члена фибоначчи

# в a1 - ответ

li a0, 4 #x10 = 4

la a1, str

ecall #вызов строки перед вводом числа

li a0, 5

ecall #ввод числа

call fibonachi

li a0, 1 #x10 = 1 (вывод результата)

ecall

li a0, 17 # ecall code

li a1, 0 # status code value

ecall #завершение работы программы с кодом завершения

.rodata

str:

.string "Please enter the number of row Fibonacci: "

**Руководство программиста.**

Краткий алгоритм.

Чтобы посчитать требуемое число Фибоначчи, нужно запоминать текущее и предыдущее число. Также нужно хранить результат. Будем суммировать два числа, затем к первому числу присваивать второе, а ко второму присвоим получившийся результат. Повторяем суммирование для вычисления следующего члена.

Ход работы программы.

1) В файле mainfunction вызываем текстовое поле для ввода порядкового номера члена Фибоначчи и сохраняем данное значение в регистре a0. Вызываем функцию fibonachi

2) В функции fibonachi во временный регистр t0 записываем значение из регистра а0. В регистре t1 мы ссылаемся на адрес numfib, в котором должен лежать порядковый номер члена Фибоначчи. И затем в регистр t1 записываем значение из регистра t0

3) В регистре a3 ссылаемся на адрес numfib. Фактически в регистре записано значение из t0 (а то получилось из a0)

4) В регистр a7 записывается константа. В регистр a4 – первое число, a5 – второе, a6 – ответ (по факту он будет в a0 в конце программы)

5) Перед входом в цикл осуществляется безусловный переход на условие совершения цикла. Если условие верно, программы продолжает свое выполнения, прыгая к метке loop:

6) В цикле происходит описанный алгоритм

7) Затем после цикла в регистр a1 записываем значение ответа из регистра a6 и делаем возврат из функции. Выполнение программы продолжится со следующей инструкции, следующей после вызова fibonachi. Адрес этой инструкции сохранён в регистре ra.

8) Далее выводим результат, записанный нами в регистр a1 и завершаем программу с кодом завершения 0.

**Вывод.**

В ходе выполнения работы были изучены возможности языка RISC-V. Разработана программа по расчёту к-го члена Фибоначчи, реализующая указанную функциональность в задании. Осуществлен вызов программы через функцию main и вызова соответствующей подпрограммы. Полученные результаты соответствуют ожидаемым.