

Домашняя работа 2 (Гринченко Евгений, БПИ 236)

riscv_hw2.s

```
1 main:
2 mv t0, zero #псевдокоманда, которая ассемблируется в базовую инструкцию add(тип «регистр-регистр-регистр»)
3 li t1, 1     #псевдокоманда, которая ассемблируется в базовую инструкцию addi(тип «непосредственное значение-регистр»)
4
5 li a7, 5     #псевдокоманда, которая ассемблируется в базовую инструкцию addi(тип «непосредственное значение-регистр»)
6 ecall       #базовая инструкция, которая выполняет системный вызов, указанный значением в a7 (тип «непосредственное значение-регистр»)
7 mv t3, a0    #псевдокоманда, которая ассемблируется в базовую инструкцию add(тип «регистр-регистр-регистр»)
8 fib:
9 beqz t3, finish #псевдокоманда, которая ассемблируется в базовую инструкцию beq, SB-type
10 add t2, t1, t0 #базовая инструкция, тип «регистр-регистр-регистр» (Register)
11 mv t0, t1     #псевдокоманда, которая ассемблируется в базовую инструкцию add(тип «регистр-регистр-регистр»)
12 mv t1, t2     #псевдокоманда, которая ассемблируется в базовую инструкцию add(тип «регистр-регистр-регистр»)
13 addi t3, t3, -1 #базовая инструкция, тип «непосредственное значение-регистр-регистр» (Immediate)
14 j fib        #псевдокоманда, которая ассемблируется в базовую инструкцию jal, UJ-type
15 finish:
16 li a7, 1     #псевдокоманда, которая ассемблируется в базовую инструкцию addi(тип «непосредственное значение-регистр»)
17 mv a0, t0    #псевдокоманда, которая ассемблируется в базовую инструкцию add(тип «регистр-регистр-регистр»)
18 ecall       #базовая инструкция, которая выполняет системный вызов, указанный значением в a7 (тип «непосредственное значение-регистр»)
19
20
```

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x000002b3	add x5,x0,x0	2: mv t0, zero #псевдокоманда, которая ассемблируется в ...
	0x00400004	0x00100313	addi x6,x0,1	3: li t1, 1 #псевдокоманда, которая ассемблируется в ...
	0x00400008	0x00500893	addi x17,x0,5	5: li a7, 5 #псевдокоманда, которая ассемблируется в ...
	0x0040000c	0x00000073	ecall	6: ecall #базовая инструкция, которая выполняет си...
	0x00400010	0x00a00e33	add x28,x0,x10	7: mv t3, a0 #псевдокоманда, которая ассемблируется в ...
	0x00400014	0x000e0c63	beq x28,x0,0x00000018	9: beqz t3, finish #псевдокоманда, которая ассемблируе...
	0x00400018	0x005303b3	add x7,x6,x5	10: add t2, t1, t0 #базовая инструкция, тип «регистр-р...
	0x0040001c	0x006002b3	add x5,x0,x6	11: mv t0, t1 #псевдокоманда, которая ассемблируе...
	0x00400020	0x00700333	add x6,x0,x7	12: mv t1, t2 #псевдокоманда, которая ассемблируе...
	0x00400024	0xffff0e13	addi x28,x28,0xfffffff	13: addi t3, t3, -1 #базовая инструкция, тип «непосред...
	0x00400028	0xfedff06f	jal x0,0xfffffffec	14: j fib #псевдокоманда, которая ассемблируе...
	0x0040002c	0x00100893	addi x17,x0,1	16: li a7, 1 #псевдокоманда, которая ассемблирует...
	0x00400030	0x00500533	add x10,x0,x5	17: mv a0, t0 #псевдокоманда, которая ассемблирует...
	0x00400034	0x00000073	ecall	18: ecall #базовая инструкция, которая выполня...

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010200	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Messages Run I/O

```
20
6765
-- program is finished running (dropped off bottom) --
```

Clear