### Д3-1. Отчёт.

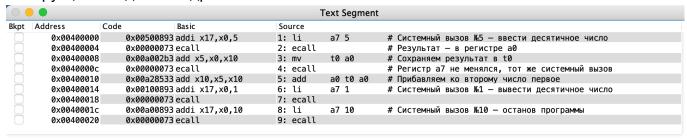
Васюков Александр Владимирович, БПИ235.

#### Task 1. 01-add-int01

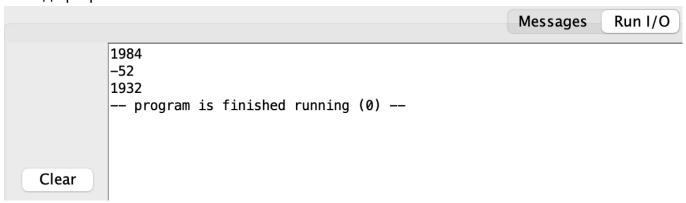
Ввод двух целых чисел (-2^32 <= х <= 2^32 - 1) и вывод их суммы.

```
Execute
                                                                   Edit
                                                                            2.asm
                                                                   1.asm
    li
            a7 5
                         # Системный вызов №5 — ввести десятичное число
 1
 2
    ecall
                         # Результат — в регистре а0
                         # Сохраняем результат в t0
 3
    mν
            t0 a0
    ecall
                         # Регистр а7 не менялся, тот же системный вызов
 4
                         # Прибавляем ко второму число первое
    add
 5
            a0 t0 a0
                         # Системный вызов №1 — вывести десятичное число
 6
    li
            a7 1
    ecall
 7
            a7 10
 8
    li
                         # Системный вызов №10 — останов программы
 9
    ecall
10
```

### Инструкции с кодами и адресами:



#### Вывод программы:



#### Task 2. 02-hello01

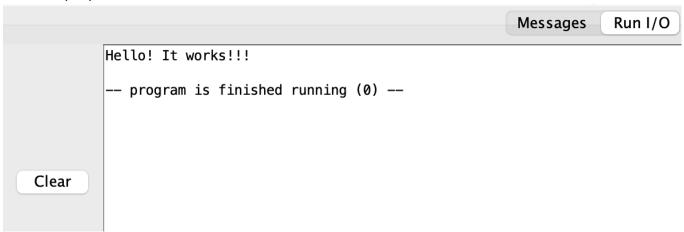
Вывод строки "Hello! It works!!!\n", где \n' - перенос на новую строку.

```
.text
1
           la a0, string
2
                            # buffer
                               # syscall write (4)
            li a7, 4
3
            ecall
4
           li a0, 0
5
                              # exit code
           li a7, 10
                           # syscall exit
6
           ecall
7
    .data
8
           string: .asciz "Hello! It works!!!\n"
9
10
```

## Инструкции с кодами и адресами:

					Text Segment			
Bkpt	Address	Code	Basic	Source				
	0x00400000	0x0fc10517	auipc x10,0x0000fc10	2:	la a0,	string	# buf	fer
	0x00400004	0x00050513	addi x10,x10,0					
	0x00400008	0x00400893	addi x17,x0,4	3:	li a7,	4	# sys	call write (4)
	0x0040000c	0x00000073	ecall	4:	ecall			
	0x00400010	0x00000513	addi x10,x0,0	5:	li a0,	0	# exi	t code
	0x00400014	0x00a00893	addi x17,x0,10	6:	li a7,	10	# sys	call exit
	0x00400018	0x00000073	ecall	7:	ecall			

### Вывод программы:

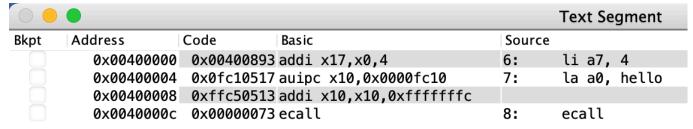


Task 3. 03-hello01

Вывод строки "Hello, world!".

```
1.asm 2.asm
                                                                                 3.asm
       .data
1
2
  hello:
3
       .asciz "Hello, world!"
4
       .text
5
  main:
       li a7, 4
6
       la a0, hello
7
8
       ecall
9
```

Инструкции с кодами и адресами:



### Вывод программы:

```
Hello, world!
--- program is finished running (dropped off bottom) ---

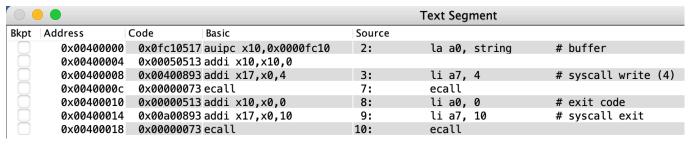
Clear
```

#### Task 4. 04-hello01

Вывод строки "Hello! It works!!!\n", где '\n' - перевод на новую строку.

```
Edit
                                                                         Execute
                                                        1.asm
                                                                 2.asm
                                                                           3.asm
                                                                                    4.asm
1
    .text
                                # buffer
2
            la a0, string
3
            li a7, 4
                                # syscall write (4)
4
    .data
        string: .asciz "Hello! It works!!!\n"
5
6
    .text
7
            ecall
                                # exit code
8
            li a0, 0
9
            li a7, 10
                                # syscall exit
10
            ecall
11
```

Инструкции с кодами и адресами:



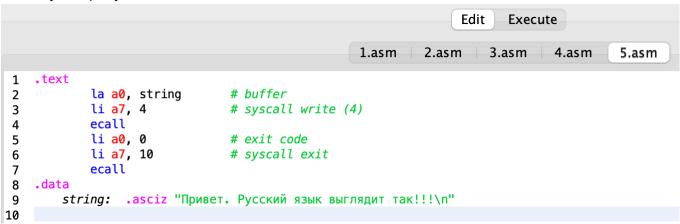
#### Вывод программы:

```
Hello! It works!!!

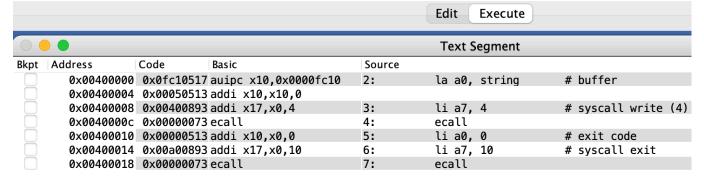
-- program is finished running (0) --
```

#### Task 5. 05-hello01

Вывод строки "Привет. Русский язык выглядит так!!!\n" на русском языке, где '\n' - перевод на новую строку.



Инструкции с кодами и адресами:



#### Вывод программы:

```
Меssages Run I/O
Привет. Русский язык выглядит так!!!
-- program is finished running (0) --
```

#### Task 6. 06-add-int02

Вывод подсказки и ввод первого целого числа.

Вывод подсказки и ввод второго целого числа.

Вывод результата суммы этих чисел.

```
2.asm
                                                               3.asm
                                                                                 5.asm
                                            1.asm
                                                                        4.asm
                                                                                          6.asm
   .data
        arg01: .asciz "Input 1st number: "
2
        arg02: .asciz "Input 2nd number: "
3
        result: .asciz "Result = "
 4
               .asciz "\n"
 5
        ln:
6
   .text
           la
                   a0, arg01 # Подсказка для ввода первого числа
7
           li
                               # Системный вызов №4
8
                   a7, 4
           ecall
9
           li
                   a7 5
                               # Системный вызов №5 — ввести десятичное число
10
           ecall
                               # Результат — в регистре а0
11
12
                   t0 a0
                               # Сохраняем результат в t0
           ΜV
13
           la
                   a0, arg02 # Подсказка для ввода второго числа
14
                               # Системный вызов №4
15
           li
                   a7, 4
16
           ecall
                   a7 5
17
           li
                               # Системный вызов №5 — ввести десятичное число
18
           ecall
                               # Результат — в регистре а0
                   t1 a0
                               # Сохраняем результат в t1
19
           mν
20
           la a0, result # Подсказка для выводимого результата
21
           li a7, 4
                               # Системный вызов №4
22
           ecall
23
           add
                   a0 t0 t1 # Складываем два числа
24
                   a7 1
                               # Системный вызов №1 — вывести десятичное число
           li
25
           ecall
26
27
28
           la a0, ln
                               # Перевод строки
29
           li a7, 4
                               # Системный вызов №4
           ecall
30
31
           li
                   a7 10
32
                               # Системный вызов №10 — останов программы
           ecall
33
34
```

Инструкции с кодами и адресами:

			E	dit Exec	ute							
○ ● Text Segment												
Bkpt	Address	Code Basic	Source									
	0x00400000	0x0fc10517 auipc x10,0x0000fc10	7:	la	a0, arg01	# Подсказка для ввода первого числа						
	0x00400004	0x00050513 addi x10,x10,0										
	0x00400008	0x00400893 addi x17,x0,4	8:	li	a7 <b>,</b> 4	# Системный вызов №4						
	0x0040000c	0x00000073 ecall	9:	ecall								
	0x00400010	0x00500893 addi x17,x0,5	10:	li	a7 5	# Системный вызов №5 — ввести десятичное число						
	0x00400014	0x00000073 ecall	11:	ecall		# Результат — в регистре а0						
	0x00400018	0x00a002b3 add x5,x0,x10	12:	mv	t0 a0	# Сохраняем результат в t0						
	0x0040001c	0x0fc10517 auipc x10,0x0000fc10	14:	la	a0, arg02	# Подсказка для ввода второго числа						
	0x00400020	0xff750513 addi x10,x10,0xfffffff7										
	0x00400024	0x00400893 addi x17,x0,4	15:	li	a7 <b>,</b> 4	# Системный вызов №4						
	0x00400028		16:	ecall								
	0x0040002c	0x00500893 addi x17,x0,5	17:	li	a7 5	# Системный вызов №5 — ввести десятичное число						
	0x00400030	0x00000073 ecall	18:	ecall		# Результат — в регистре а0						
	0x00400034		19:	mv	t1 a0	# Сохраняем результат в t1						
	0x00400038		21:	la a0,	result	# Подсказка для выводимого результата						
	0x0040003c											
	0x00400040	,	22:	li a7,	4	# Системный вызов №4						
	0x00400044	0x00000073 ecall	23:	ecall								
	0x00400048	,,	24:	add	a0 t0 t1	# Складываем два числа						
	0x0040004c		25:	li	a7 1	# Системный вызов №1 — вывести десятичное число						
	0x00400050		26:	ecall								
	0x00400054		28:	la a0,	ln	# Перевод строки						
	0x00400058	,										
	0x0040005c		29:	li a7,	4	# Системный вызов №4						
	0×00400060		30:	ecall								
	0x00400064		32:	li	a7 10	# Системный вызов №10 — останов программы						
	0x00400068	0x00000073 ecall	33:	ecall								

### Вывод программы:

Input 1st number: 12
Input 2nd number: 7
Result = 19
-- program is finished running (0) --

# Псевдокоманды в программе 06-add-int02:

- 1. la преобразуется в auipc и addi
- 2. li преобразуется в addi
- 3. mv преобразуется в addi

### Типы форматов команд в программе 06-add-int02:

- 1. Іа псевдокоманда, преобразующаяся в І-type и U-type
- 2. Іі псевдокоманда, преобразующаяся в І-type
- 3. ecall I-type
- 4. mv псевдокоманда, преобразующаяся в R-type
- 5. add R-type

#### Системные вызовы:

- 1. №1 вывод целого десятичного числа
- 2. №4 вывод строки

- 3. №5 ввод целого десятичного числа
- 4. №10 выход из программы с кодом ошибки 0

# Видео с запуском программ на гугл-диске:

 $\underline{https://drive.google.com/drive/folders/1IeUhx9Oj5TaZygGf7uCI13eoc8RrP5Rs?usp=sharing}$