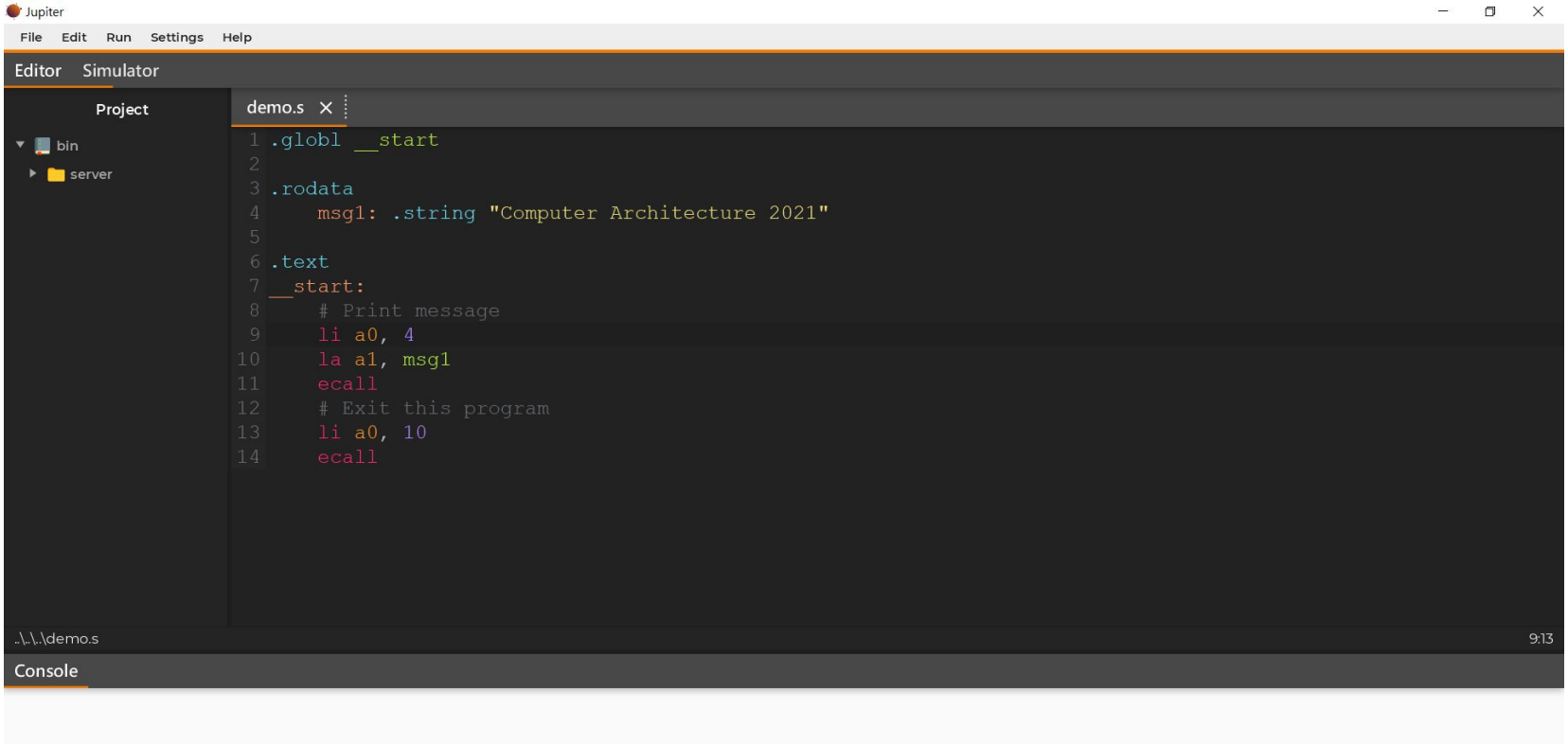
CA2021 Spring HW2

RISC-V Assmbly Code

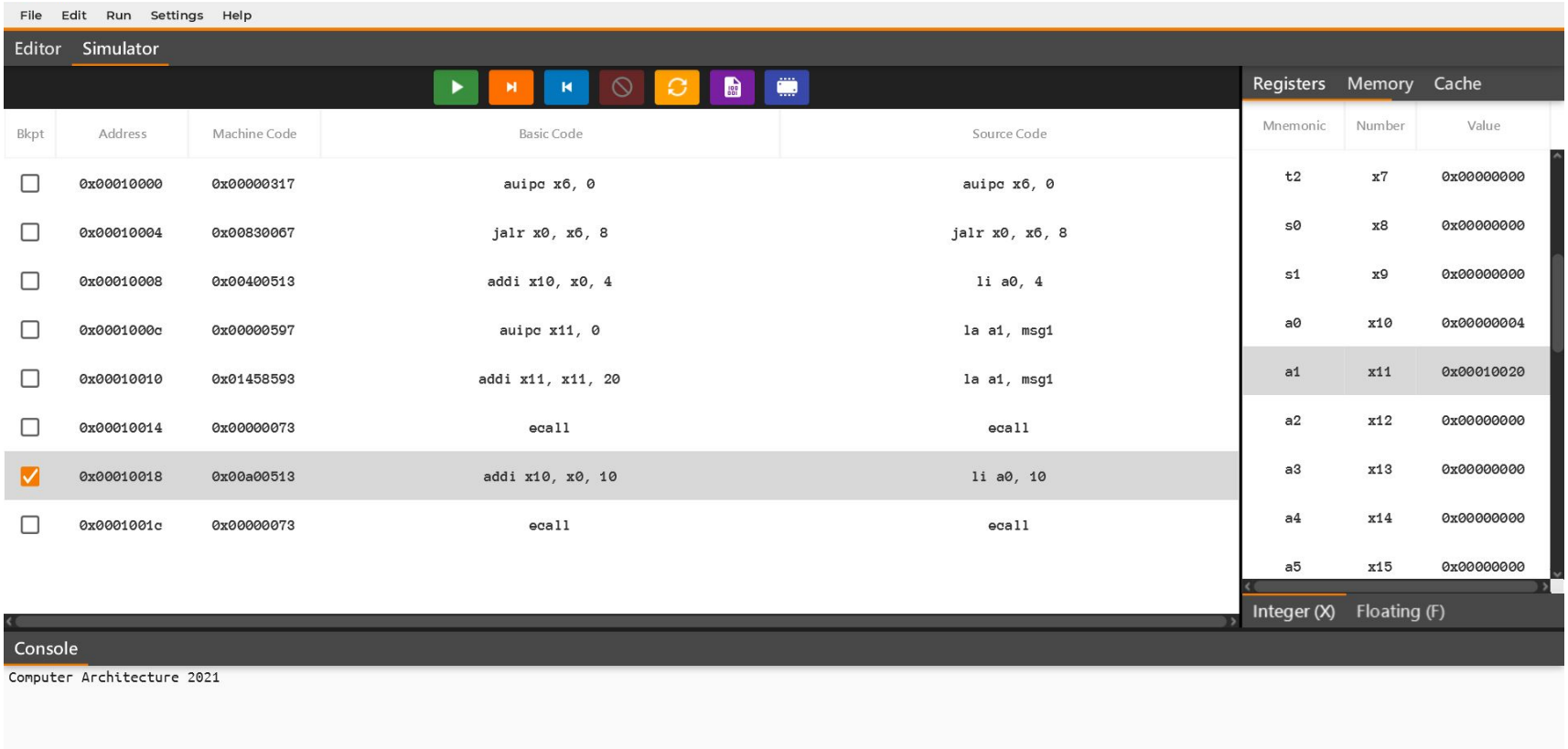
Description

- In this homework, you are going to use [Jupiter RISC-V simulator] to develop a simple calculator.

Jupiter


Jupiter
File Edit Run Settings Help
Editor Simulator
Project
bin
server
demo.s
1 .globl __start
2
3 .rodata
4 msg1: .string "Computer Architecture 2021"
5
6 .text
7 __start:
8 # Print message
9 li a0, 4
10 la a1, msg1
11 ecall
12 # Exit this program
13 li a0, 10
14 ecall
..\..\..\demo.s
9:13
Console

Jupiter


File Edit Run Settings Help
Editor Simulator
Bkpt Address Machine Code Basic Code Source Code
0x00010000 0x00000317 auipc x6, 0 auipc x6, 0
0x00010004 0x00830067 jalr x0, x6, 8 jalr x0, x6, 8
0x00010008 0x00400513 addi x10, x0, 4 li a0, 4
0x0001000c 0x00000597 auipc x11, 0 la a1, msg1
0x00010010 0x01458593 addi x11, x11, 20 la a1, msg1
0x00010014 0x00000073 ecall ecall
0x00010018 0x00a00513 addi x10, x0, 10 li a0, 10
0x0001001c 0x00000073 ecall ecall
Registers Memory Cache
Mnemonic Number Value
t2 x7 0x00000000
s0 x8 0x00000000
s1 x9 0x00000000
a0 x10 0x00000004
a1 x11 0x00010020
a2 x12 0x00000000
a3 x13 0x00000000
a4 x14 0x00000000
a5 x15 0x00000000
Integer (X) Floating (F)
Console
Computer Architecture 2021

TO-DO

- You are going to develop a simple calculator, which supports six operations.
- Addition(0), subtraction(1), multiplication(2), integer division(3), power(4), and factorial(5).
    - For simplicity, we use the numbers in the quote to represent the operations.

- Input file contains 3 lines, operand *A*, operation *op*, operand *B*, respectively. ($0 \le A, B \le 1024$, $op \in \{0, 1, 2, 3, 4, 5\}$)
- Your program should output the correct result (*A op B*).

Sample I/O

```
r09922113@linux1 [~/CA] jupiter hw2.s
5
0
2
7

Jupiter: exit(0)
r09922113@linux1 [~/CA] jupiter hw2.s
7
3
4
1

Jupiter: exit(0)
r09922113@linux1 [~/CA] jupiter hw2.s
4
5
0
24

Jupiter: exit(0)
```

Sample Code

- In the sample code, you don't need to do I/O operations on yourself. *A*, *op*, *B* will be stored at register s0, s1, s2 registers. And you need to store the result to register s3.

```
20
21  ###################################
22  #  TODO: Develop your calculator  #
23  #                                 #
24  ###################################
25
```

- If *op=3* and *B=0*, just jump to zero_except block.

```
35  zero_except:
36      # Divide by zero exception
37      li a0, 4
38      la a1, divide_by_zero
39      ecall
```

Grading Policy

- We will judge the correctness of your program on CSIE workstation.

  ```
  $jupiter [student_id]_hw2.s < input_file
  ```

- Don’t worry about overflow and underflow.
- 10 points off per day for late submission.
- You will get 0 point for plargism.

Submission

- Due date: 3 / 30 23:59 (Tuesday)
- Please rename your program **[student_id]_hw2.s** and upload it to NTU COOL.