
CA2021 Fall HW2

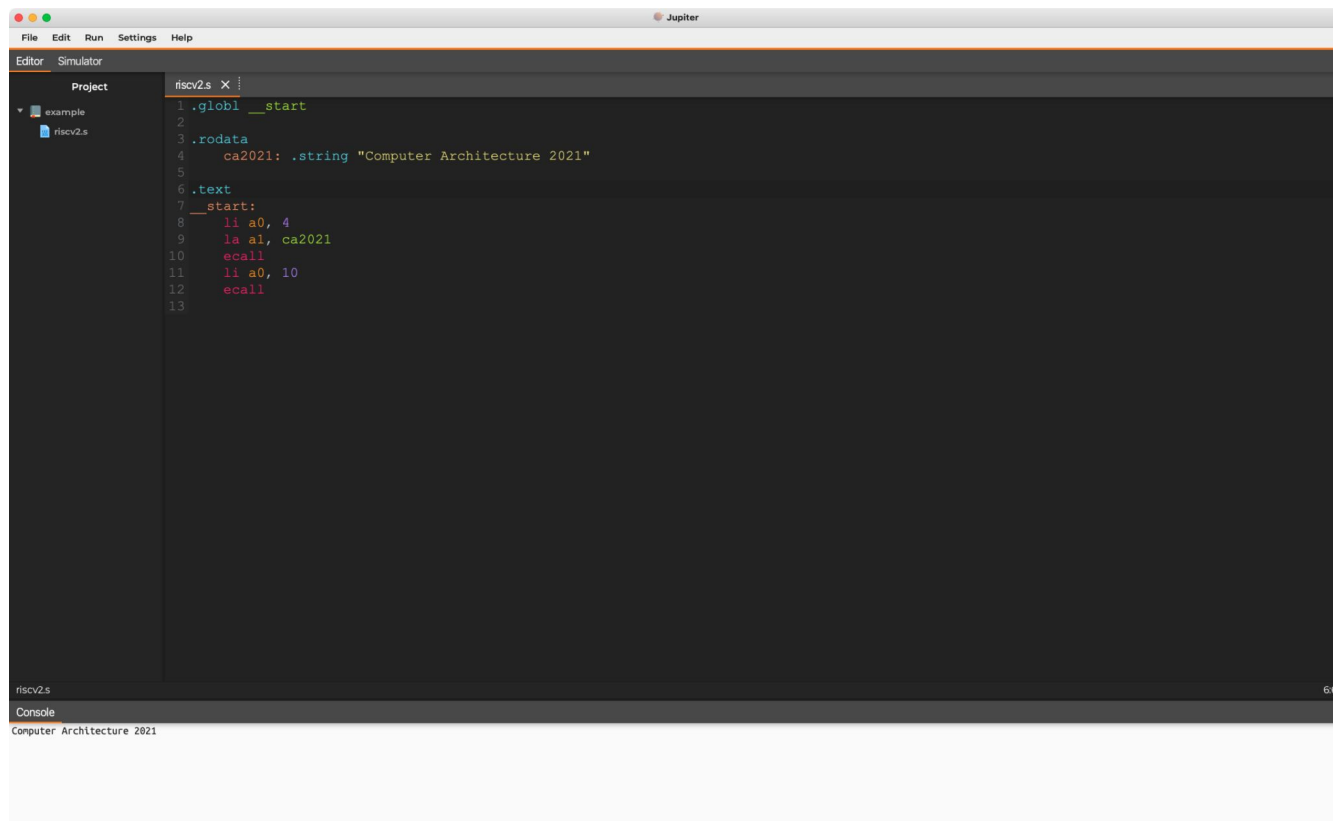
— RISC-V Assembly Code —

Description

- In this homework, you are going to use [Jupiter RISC-V simulator](#) to develop a simple calculator.
- After finishing this homework, you will be familiar with the usage of Jupiter RISC-V simulator, register definition, and some basic operations in RV32I Base Integer Instruction Set.



Jupiter GUI



Jupiter GUI

The screenshot displays the Jupiter GUI, a tool for simulating MIPS assembly code. The interface is divided into several sections:

- Menu Bar:** File, Edit, Run, Settings, Help.
- Editor/Debugger Tabs:** Editor and Simulator. The Simulator tab is active.
- Simulation Controls:** A row of buttons including a green play button (highlighted with a red box), a red stop button, a blue refresh button, a yellow save button, and a purple debug button.
- Assembly Table:** A table with columns: Bkpt, Address, Machine Code, Basic Code, and Source Code. It lists assembly instructions with their corresponding addresses and machine codes.
- Registers Panel:** A panel on the right showing the state of MIPS registers (zero, ra, sp, gp, tp, t0, t1, t2, s0, s1, a0, a1, a2, a3, a4, a5) with their Mnemonic, Number, and Value.
- Console:** A panel at the bottom showing the output of the simulation, including the text "Computer Architecture 2021" (highlighted with a red box).

Bkpt	Address	Machine Code	Basic Code	Source Code
<input type="checkbox"/>	0x00010000	0x00000317	auipc x6, 0	auipc x6, 0
<input type="checkbox"/>	0x00010004	0x00030067	jair x0, x6, 8	jair x0, x6, 8
<input type="checkbox"/>	0x00010008	0x00400513	addi x10, x0, 4	li a0, 4
<input type="checkbox"/>	0x0001000c	0x00000597	auipc x11, 0	la a1, ca2021
<input type="checkbox"/>	0x00010010	0x01458593	addi x11, x11, 20	la a1, ca2021
<input type="checkbox"/>	0x00010014	0x00000073	ecall	ecall
<input type="checkbox"/>	0x00010018	0x00a00513	addi x10, x0, 10	li a0, 10
<input type="checkbox"/>	0x0001001c	0x00000073	ecall	ecall

Registers	Memory	Cache
Mnemonic	Number	Value
zero	x0	0x00000000
ra	x1	0x00000000
sp	x2	0xbffffff0
gp	x3	0x10000000
tp	x4	0x00000000
t0	x5	0x00000000
t1	x6	0x00010000
t2	x7	0x00000000
s0	x8	0x00000000
s1	x9	0x00000000
a0	x10	0x0000000a
a1	x11	0x00010020
a2	x12	0x00000000
a3	x13	0x00000000
a4	x14	0x00000000
a5	x15	0x00000000

Console
Computer Architecture 2021

exit(0)

Jupiter CLI

```
> jupiter riscv2.s  
Computer Architecture 2021  
  
Jupiter: exit(0)
```

TODO

- You are going to develop a simple calculator, which supports seven operations.
- Addition(0), subtraction(1), multiplication(2), integer division(3), integer remainder(4), power(5), and factorial(6).
 - For simplicity, we use the numbers in the quote to represent the operations.

Sample I/O

- Input file contains 3 lines, operand A, operation op, operand B, respectively. ($0 \leq A, B \leq 1024$, $op \in \{0, 1, 2, 3, 4, 5, 6\}$)
- Your program should output the correct result ($A \text{ op } B$).

```
> jupiter hw2.s
10
0
10
20

Jupiter: exit(0)
```

```
> jupiter hw2.s
10
1
10
0

Jupiter: exit(0)
```

```
> jupiter hw2.s
10
2
10
100

Jupiter: exit(0)
```

```
> jupiter hw2.s
10
3
0
division by zero

Jupiter: exit(0)
```

Sample Code

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

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


Sample Code

- If op=3 and B=0, just jump to **division_by_zero_except**

```
38 division_by_zero_except:
39     li a0, 4
40     la a1, division_by_zero
41     ecall
42     jal zero, exit
```

- If op=4 and B=0, just jump to **remainder_by_zero_except**

```
44 remainder_by_zero_except:
45     li a0, 4
46     la a1, remainder_by_zero
47     ecall
48     jal zero, exit
```

Grading Policy

- Total 100%
 - For operations +, -, x, / and %, each has 4 test cases, 3 points per test case.
 - For operations ^ and !, each has 5 test cases, 4 points per test case.
- We will judge the correctness of your program on CSIE workstation.

```
$ jupyter [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 plagiarism.

Submission

- Due date: 10/26 23:59 (Tuesday)
- Please rename your program [student_id]_hw2.s and upload it to NTU COOL.
 - For example, if your student id is b12345678, your program file name should be b12345678_hw2.s.