

Computer Architecture hw2

Solving Recurrence Equation in RISC-V

TA: 曾奕青

email: d08922025@ntu.edu.tw

RISC-V Simulator

- **Jupiter** repository: <https://github.com/andrescv/Jupiter>
- Jupiter is an open source RISC-V assembler and runtime simulator

Jupiter

RISC-V Assembler & Runtime Simulator



RISC-V Simulator

Installation

Download the app image for your operating system and unzip the file:

- [Jupiter v3.1 - Linux \(Ubuntu\)](#)
- [Jupiter v3.1 - macOS](#)
- [Jupiter v3.1 - Windows](#)

Running Jupiter on Linux or macOS

```
./image/bin/jupiter # for GUI mode  
./image/bin/jupiter [options] <files> # for CLI mode
```

Running Jupiter on Windows


```
image\bin\jupiter # for GUI mode  
image\bin\jupiter [options] <files> # for CLI mode
```

RISC-V Simulator

- Installation

Run the following commands in a terminal

```
1  sudo add-apt-repository ppa:andrescv/jupiter  
2  sudo apt-get update  
3  sudo apt-get install jupiter
```



Jupiter only release the installation guide for linux. You can run it on other environment if you can install it without installation guide, if not, please use virtual machine.

[Ref: https://docs.riscvsim.com/installation-1/linux](https://docs.riscvsim.com/installation-1/linux)

RISC-V Simulator

- Command line

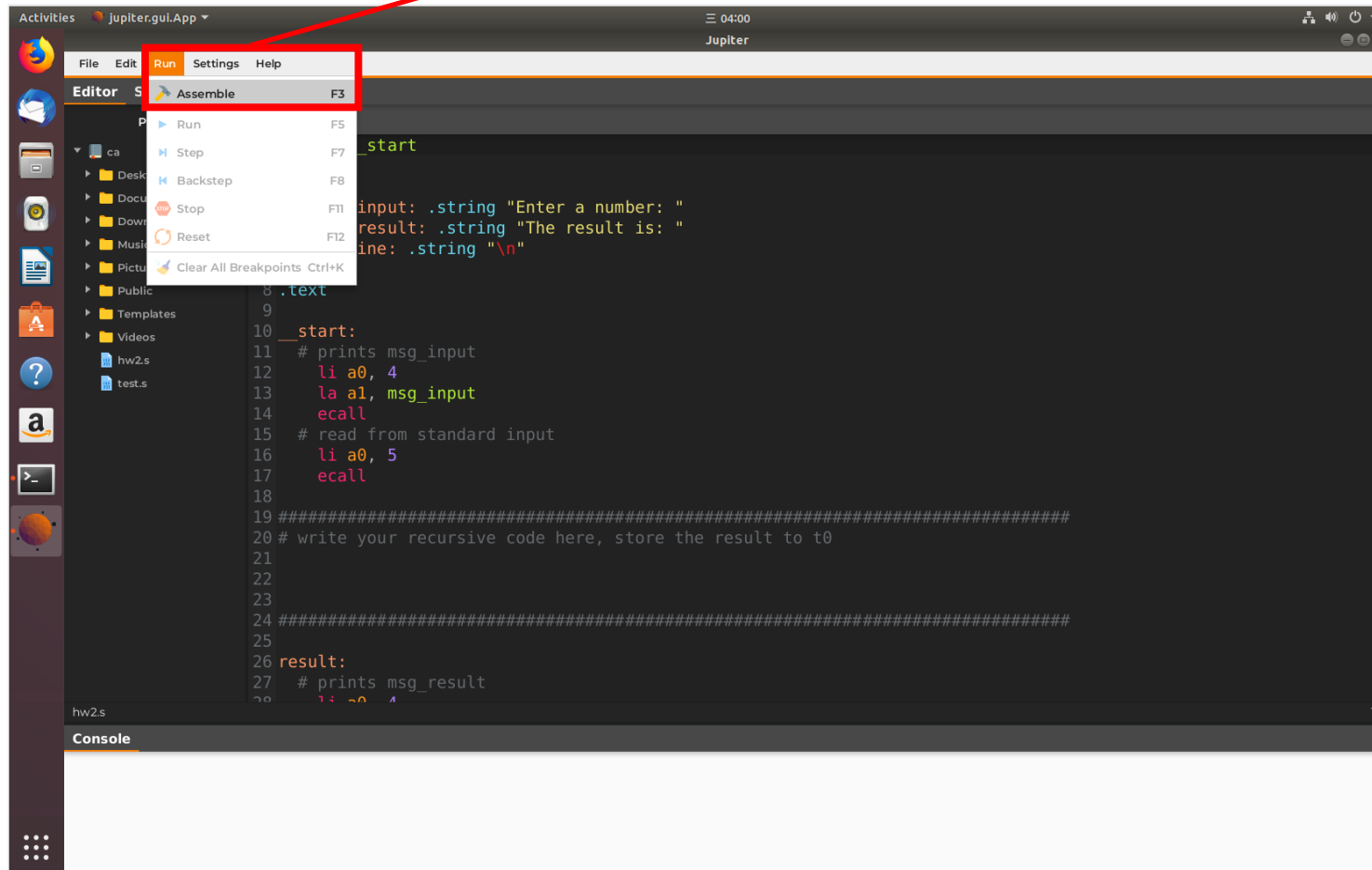
```
hamt5821@enderman:~$ jupiter hw2.s
Enter a number: 12345
The result is: 0

Jupiter: exit(0)
```

RISC-V Simulator

- GUI

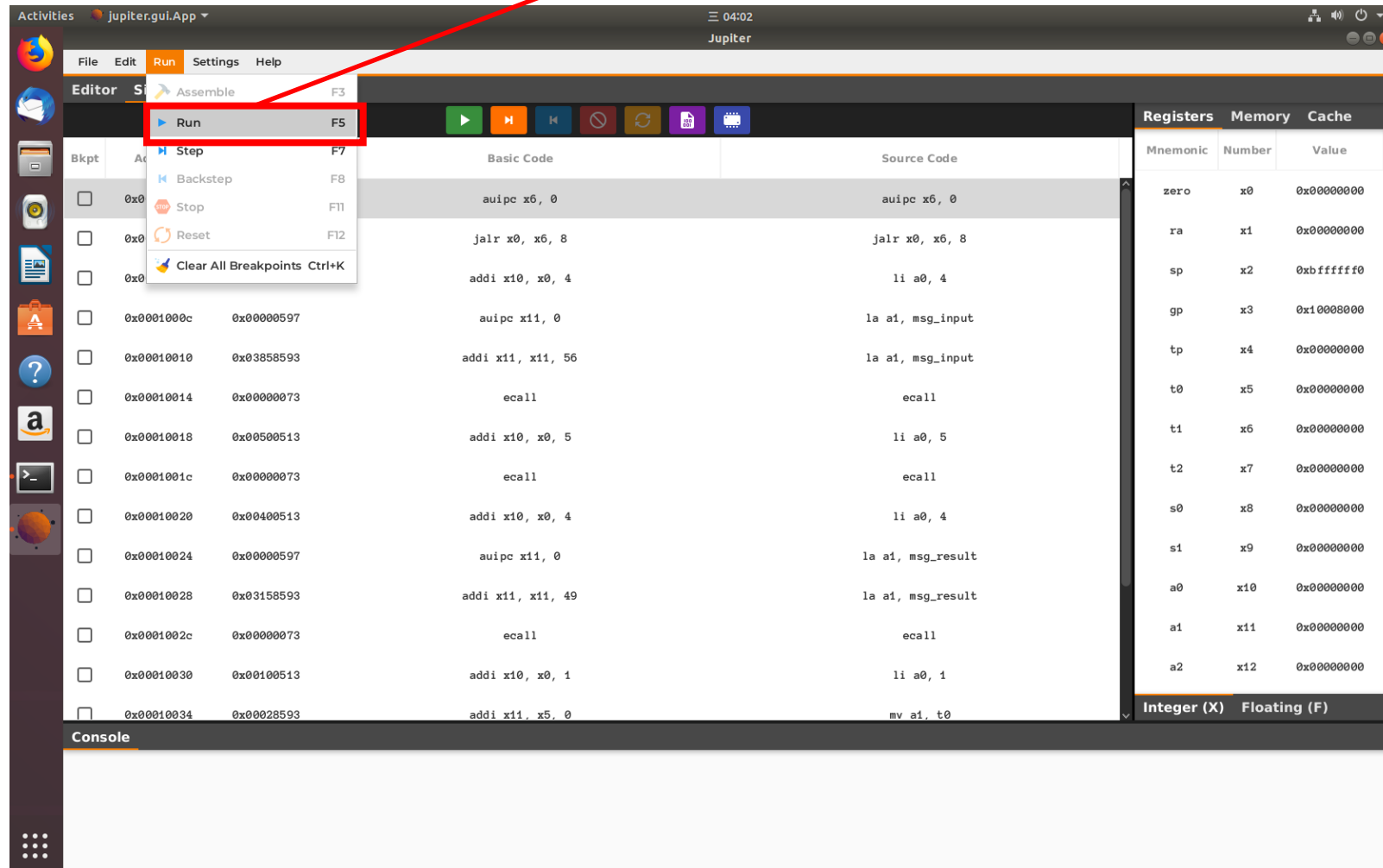
Click Run -> Assemble



RISC-V Simulator

- GUI

Click **Run** to run your code



RISC-V Simulator

- GUI

EditorSimulator

Bkpt	Address	Machine Code	Basic Code	Source Code
<input type="checkbox"/>	0x00010000	0x00000317	auipc x6, 0	auipc x6, 0
<input type="checkbox"/>	0x00010004	0x00830067	jalr x0, x6, 8	jalr 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, msg_input
<input type="checkbox"/>	0x00010010	0x03858593	addi x11, x11, 56	la a1, msg_input
<input type="checkbox"/>	0x00010014	0x00000073	ecall	ecall
<input type="checkbox"/>	0x00010018	0x00500513	addi x10, x0, 5	li a0, 5
<input type="checkbox"/>	0x0001001c	0x00000073	ecall	ecall
<input type="checkbox"/>	0x00010020	0x00400513	addi x10, x0, 4	li a0, 4
<input type="checkbox"/>	0x00010024	0x00000597	auipc x11, 0	la a1, msg_result
<input type="checkbox"/>	0x00010028	0x03158593	addi x11, x11, 49	la a1, msg_result
<input type="checkbox"/>	0x0001002c	0x00000073	ecall	ecall
<input type="checkbox"/>	0x00010030	0x00100513	addi x10, x0, 1	li a0, 1
<input type="checkbox"/>	0x00010034	0x00028593	addi x11, x5, 0	mv a1, t0

RegistersMemoryCache

Mnemonic	Number	Value
zero	x0	0x00000000
ra	x1	0x00000000
sp	x2	0xbfffffff0
gp	x3	0x10008000
tp	x4	0x00000000
t0	x5	0x00000000
t1	x6	0x00010000
t2	x7	0x00000000
s0	x8	0x00000000
s1	x9	0x00000000
a0	x10	0x00000005
a1	x11	0x00010044
a2	x12	0x00000000

Integer (X)Floating (F)

Console

Enter a number: |

Enter a number and check the output

RISC-V Simulator

- GUI

Activities Jupiter.gui.App 04:08 Jupiter

File Edit Run Settings Help

Editor Simulator

Registers Memory Cache

Bkpt	Address	Machine Code	Basic Code	Source Code	Mnemonic	Number	Value
<input type="checkbox"/>	0x0001000c	0x00000597	auipc x11, 0	la a1, msg_input	zero	x0	0x00000000
<input type="checkbox"/>	0x00010010	0x03858593	addi x11, x11, 56	la a1, msg_input	ra	x1	0x00000000
<input type="checkbox"/>	0x00010014	0x00000073	ecall	ecall	sp	x2	0xbfffffff0
<input type="checkbox"/>	0x00010018	0x00500513	addi x10, x0, 5	li a0, 5	gp	x3	0x10008000
<input type="checkbox"/>	0x0001001c	0x00000073	ecall	ecall	tp	x4	0x00000000
<input type="checkbox"/>	0x00010020	0x00400513	addi x10, x0, 4	li a0, 4	t0	x5	0x00000000
<input type="checkbox"/>	0x00010024	0x00000597	auipc x11, 0	la a1, msg_result	t1	x6	0x00010000
<input type="checkbox"/>	0x00010028	0x03158593	addi x11, x11, 49	la a1, msg_result	t2	x7	0x00000000
<input type="checkbox"/>	0x0001002c	0x00000073	ecall	ecall	s0	x8	0x00000000
<input type="checkbox"/>	0x00010030	0x00100513	addi x10, x0, 1	li a0, 1	s1	x9	0x00000000
<input type="checkbox"/>	0x00010034	0x00028593	addi x11, x5, 0	mv a1, t0	a0	x10	0x0000000a
<input type="checkbox"/>	0x00010038	0x00000073	ecall	ecall	a1	x11	0x00000000
<input type="checkbox"/>	0x0001003c	0x00a00513	addi x10, x0, 10	li a0, 10	a2	x12	0x00000000
<input type="checkbox"/>	0x00010040	0x00000073	ecall	ecall			

Integer (X) Floating (F)

Console

Enter a number: 123
The result is: 0

Homework 2 - Recurrence Equation

Integer division



$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + n, & \text{if } n \geq 2 \\ 1, & \text{otherwise} \end{cases}$$

Input and Output

- The sample code will read a number from standard input and print the result to standard output.
- Task
 - The input (n) will be stored in `a0`
 - Calculate $T(n)$ and put it in `t0` **in Integer format**

```
.globl __start

.rodata
msg_input: .string "Enter a number: "
msg_result: .string "The result is: "
newline: .string "\n"

.text

__start:
# prints msg_input
li a0, 4
la a1, msg_input
ecall

# read from standard input
li a0, 5
ecall

#####
# write your recursive code here, store the result to t0
#####

result:
# prints msg_result
li a0, 4
la a1, msg_result
ecall

# prints the result in t0
li a0, 1
mv a1, t0
ecall

# ends the program with status code 0
li a0, 10
ecall
```

Requirement

- Correctness of your program would be judged by output
- Implement in recursive function for full-credit, or your credit would be 40% off

Submission

- Due date: 2019/10/30 afternoon (14:20)
- 10% off per day for late submission
- You should pack the folder in a .zip file
 - Hw2_<student_id>.zip
 1. hw2_<student_id>.s
 2. readme.txt (please write the environment you use)
- Upload to NTU COOL