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

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

```
sudo add-apt-repository ppa:andrescv/jupiter
sudo apt-get update
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

RISC-V Simulator

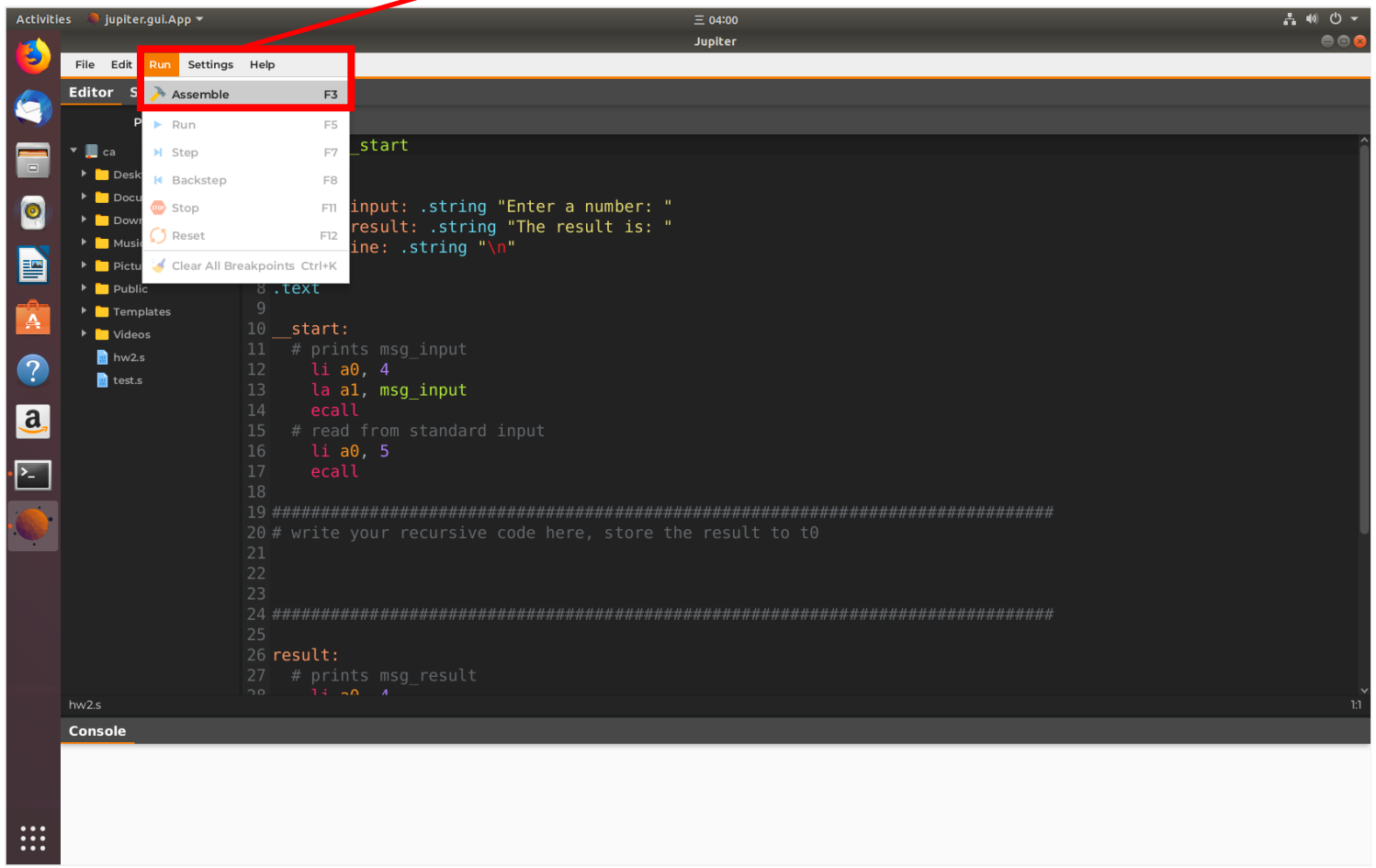
- Command line

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

Jupiter: exit(0)
```

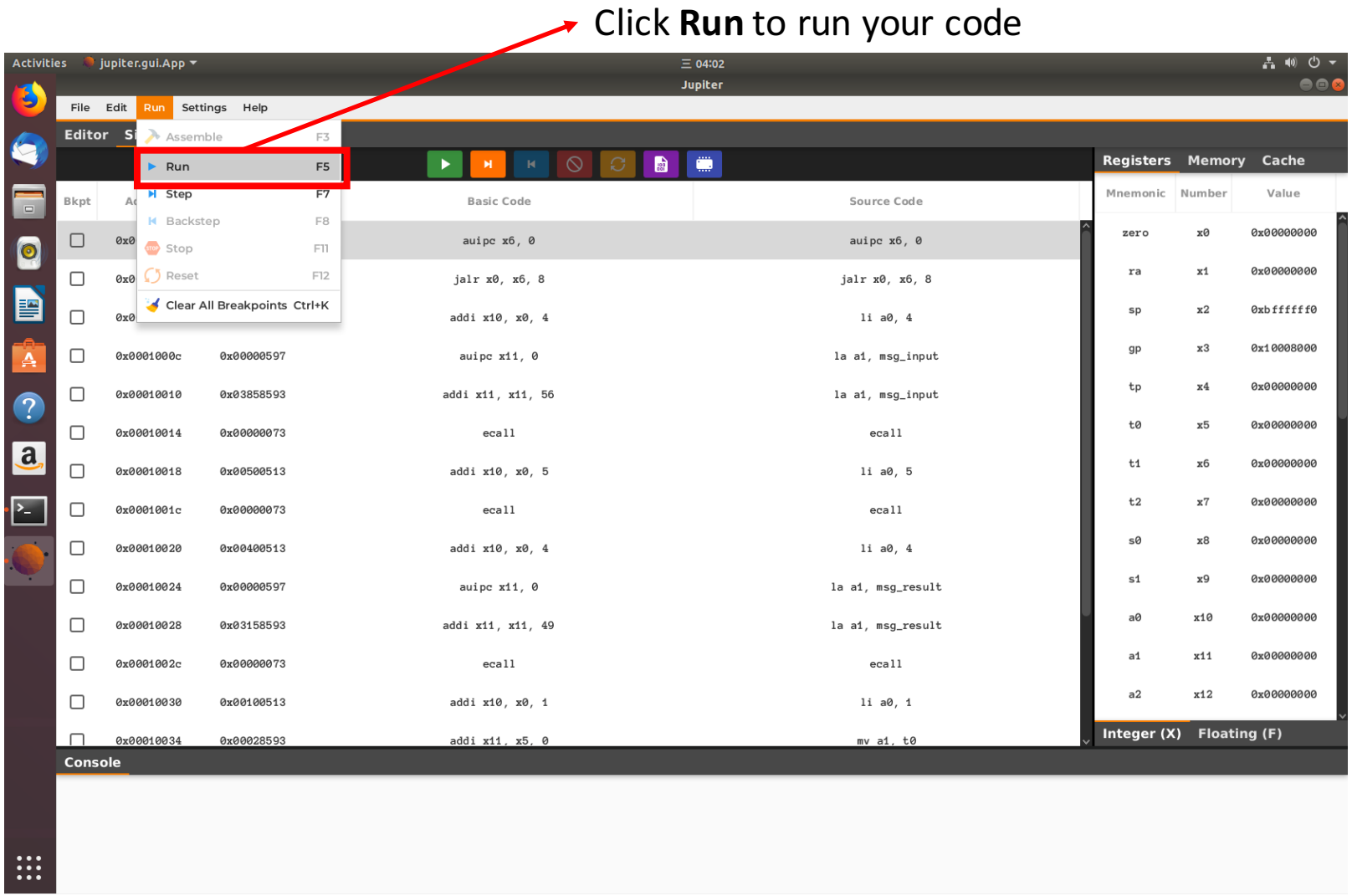
RISC-V Simulator

- GUI



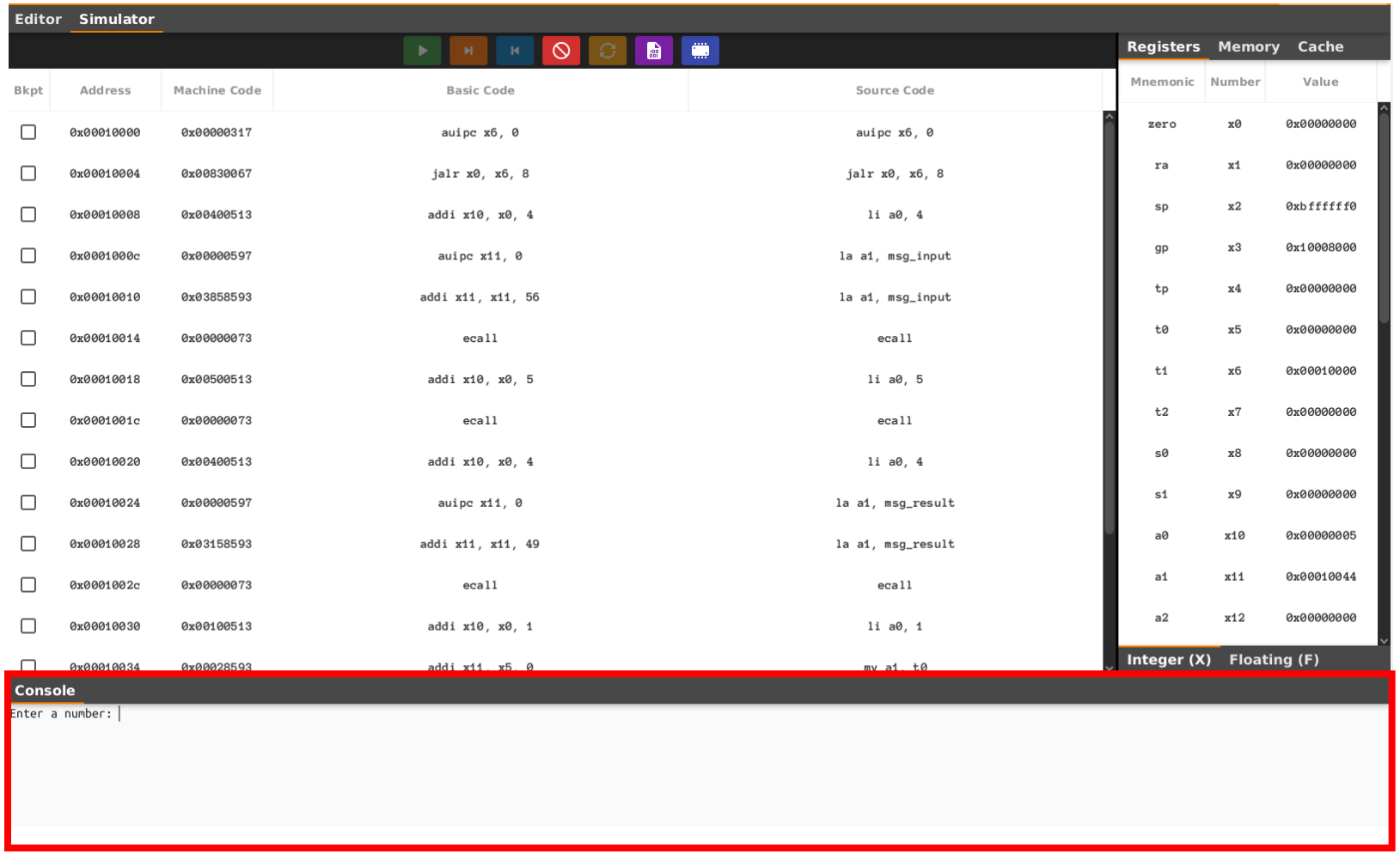
RISC-V Simulator

- GUI



RISC-V Simulator

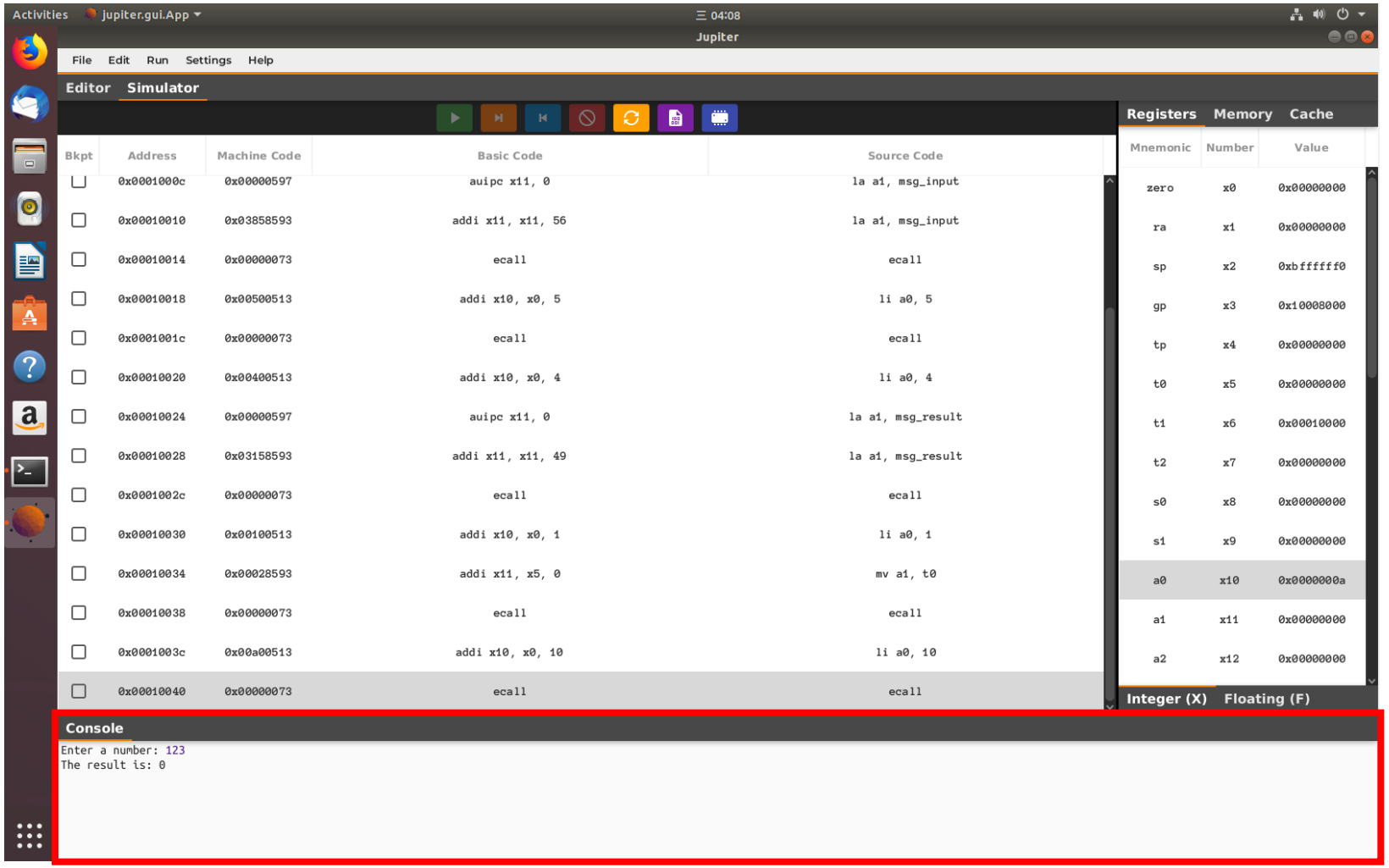
- GUI



Enter a number and check the output

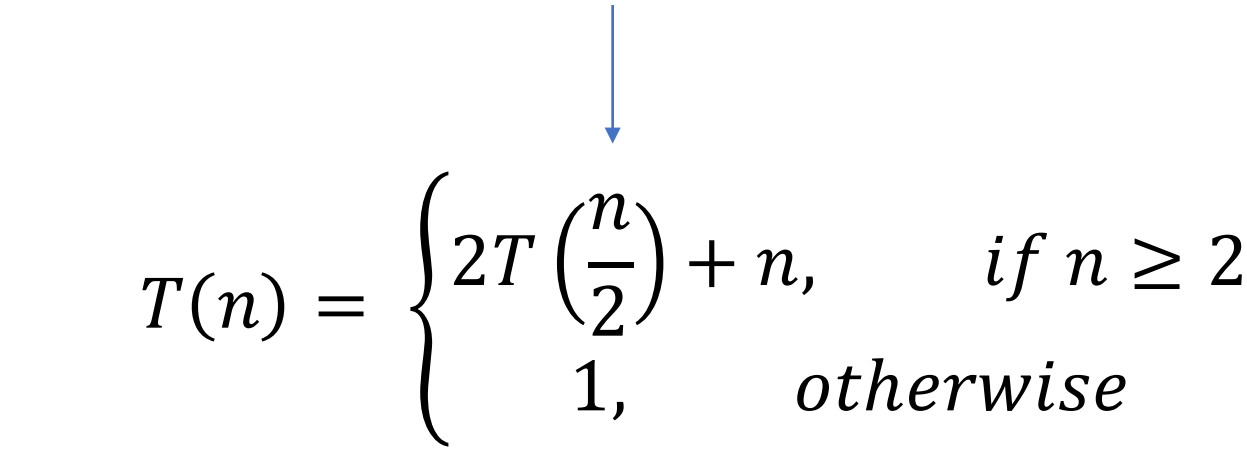
RISC-V Simulator

- GUI



Homework 2 - Recurrence Equation

Integer division



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

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