Computer architecture HW1

Due: 14:20 on Oct 22, 2024

Outline

- Jupiter: RISC-V Simulator
- HW1_1: The extended Euclidean algorithm
- HW1_2: Longest Substring without Repeating Characters
- HW1_3: Linked List Cycle
- Submission
- Rules

Jupiter: RISC-V Simulator

- An open source RISC-V assembler and runtime simulator.
- Download here: https://github.com/andrescv/Jupiter

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

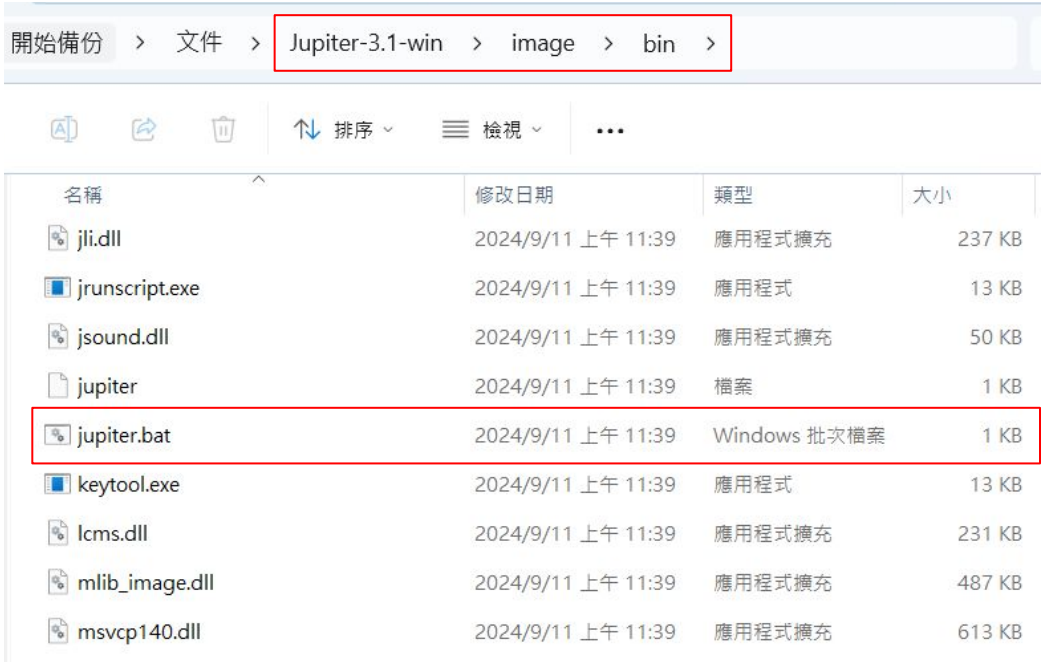
```
./image/bin/jupiter # for GUI mode
./image/bin/jupiter [options] <files> # for CLI mode
```

Running Jupiter on Windows

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

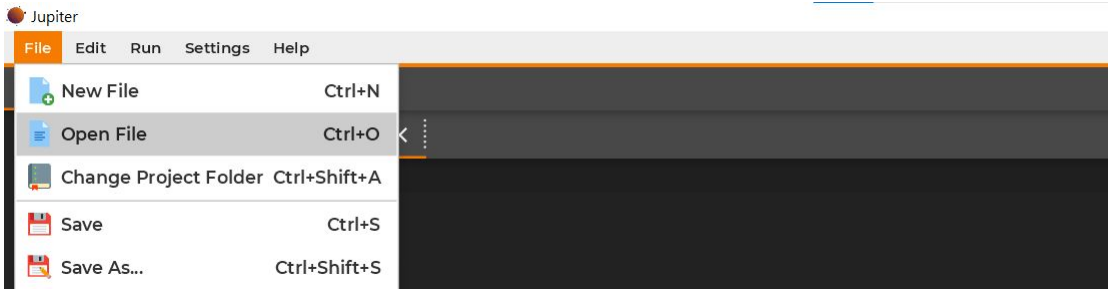
Jupiter: RISC-V Simulator

- Unzip the folder: Jupiter 3.1 win.zip
- Click jupiter.bat to lunch

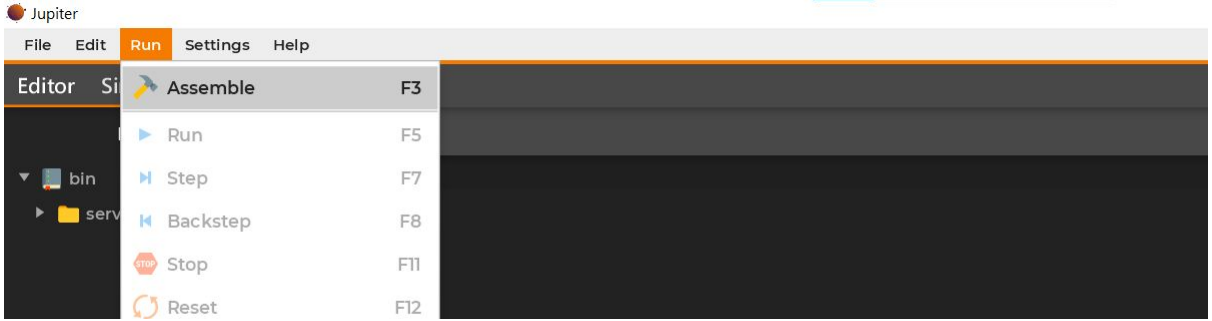


Gui for Jupiter

- Open the file

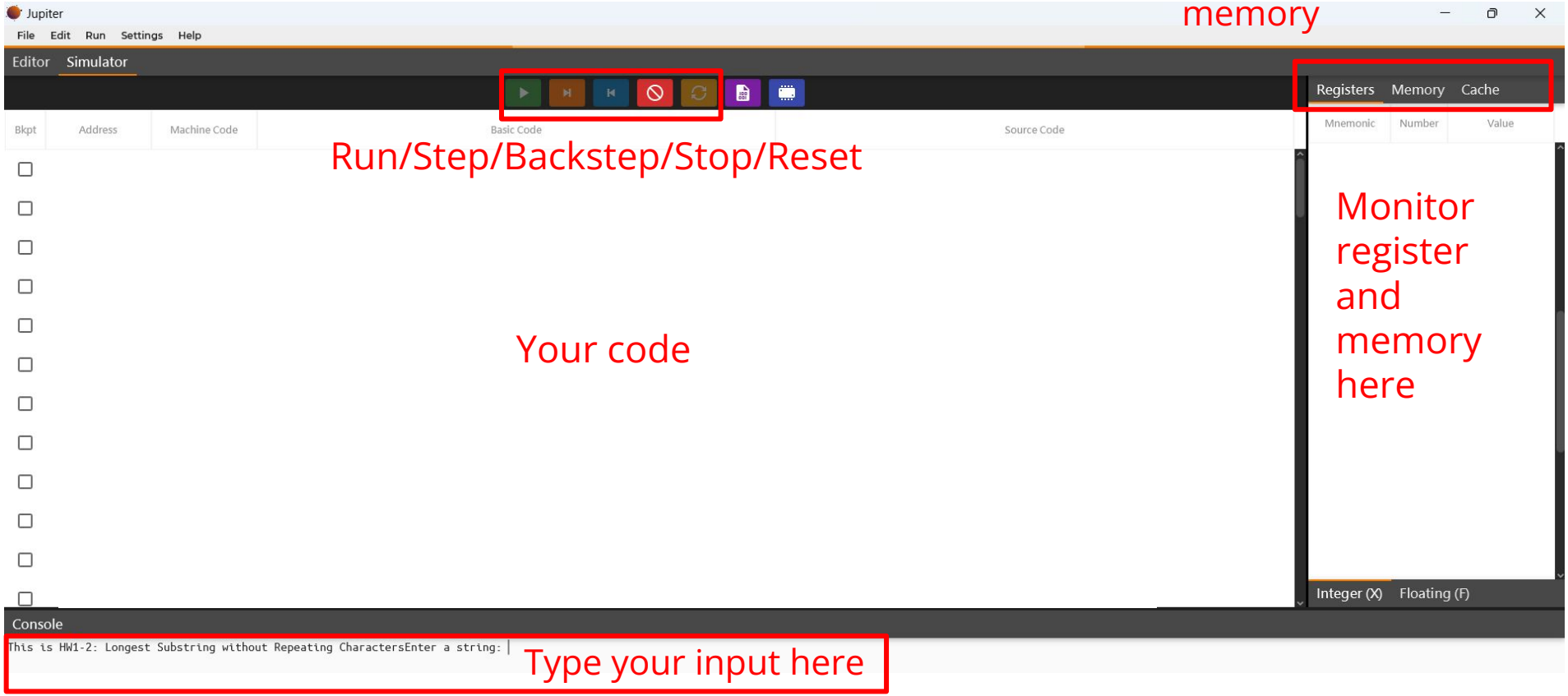


- Run the code
  - Click Run -> assemble



Gui for Jupiter (Cont.)

Select to see the content of register or memory



HW1_1 The extended Euclidean algorithm

- The Euclidean algorithm takes two integers x,y as input and outputs their greatest common divisor.
- The extended Euclidean algorithm further finds a pair of integers a,b such that $ax+by = gcd(x,y)$.

- For inputs = (527,341)

527 = 341*1 + 186
341 = 186*1 + 155
186 = 155*1 + 31
155 = 31*5 + 0

gcd(527,341) = 31
31 = 527*2 + 341*(-3)

HW1_1 The extended Euclidean algorithm

- If gcd(x,y) = 1, there exists a multiplicative inverse of x modulo y.
- z is said to be a multiplicative inverse of x modulo y if *xz mod y = 1*, that is (x*z) % y = 1.
- We also need to find $z \in \{0,1, \ldots ,y-1\}$ if it exists.

gcd(45,23) = 1
So there exists an inverse of [45 modulo 23] ≡ 22
Because (45*22)/23 = 43...1

I/O method

- Functions in “__start” will store your input from the command line into register
- Input
  - Positive integer x and y
  - x would be stored into a0, y would be stored into a1
- Output
  - gcd(x,y), a, b, z(inv(x modulo y))
  - Outputs should be stored into the specified registers
  - If inv(x modulo y) doesn’t exist, please store 0 to s3
  - See the example in the next page
- When your code is finished, jump to “result” part in script, it will print out your result in the command window.

output	register
gcd(x,y)	s0
a	s1
b	s2
z (inv (x mod y))	s3

Examples (public patterns)

```
This is HW1-1: Extended Euclidean Algorithm
Enter a number for input x: 527
Enter a number for input y: 341
The result is:
 GCD: 31
 a: 2
 b: -3
 inv(x modulo y): 0
```

```
This is HW1-1: Extended Euclidean Algorithm
Enter a number for input x: 45
Enter a number for input y: 23
The result is:
 GCD: 1
 a: -1
 b: 2
 inv(x modulo y): 22
```

Hint

- Both (a,b) can be derived by using iterative method, so how can we implement iterative method? (stack...?)
- The DIV(x,y) instruction in RISC-V can only get the quotient of x/y, so how can we get the remainder of the DIV operation? (x=y*q + r)
- Remember to keep inv(x modulo y) in {0,1, . . . ,y−1} before submitting (check it's not negative and is in the range[0,1, . . . ,y−1])

$$ax + by = gcd(x, y)$$
$$a'y + b'(x\%y) = gcd(y, x\%y) = gcd(x, y)$$
$$ax + by = a'y + b'(x - floor(x/y)* y) = b'x + (a' - b'* floor(x/y)) y$$

Iterative equation

$$\begin{cases} a=b' \\ b=a' - floor(x/y)* b' \end{cases}$$

$$a'=1,\ b'=0,\ when\ x\%y = 0$$

HW1_2: Longest Substring without Repeating Characters

- Given a string s as an array of characters, print the longest substring without repeating characters and its length. For example,
  - If s = “ababc”, the output should be “abc”.
  - If s = “aaa”, the output should be “a”.

HW1_2: Longest Substring without Repeating Characters

- Characters are stored as ascii code
- A character is 8 bits (1 byte)

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

I/O method

- Functions in “__start” will store your input from the command line into memory, and the starting address of the string would be in a0 (i.e. x10)
- Input
  - Inputs are lowercase and uppercase alphabets (A~Z and a~z).
  - Maximum input length is 100.
  - “Enter” is the last character, but should not be taken into account.
- Output
  - Output is the longest substring from the input without repeating characters.
  - Store the beginning address of the answer in the register t4.
  - You can decide the address in the register t4, just make sure that the answer can be printed correctly.
- Jump to the provided “result” when your code is finished.

Examples

```
This is HW1-2: Longest Substring without Repeating Characters
Enter a string: ababc
Answer: abc
```

```
This is HW1-2: Longest Substring without Repeating Characters
Enter a string: aaa
Answer: a
```

Public patterns

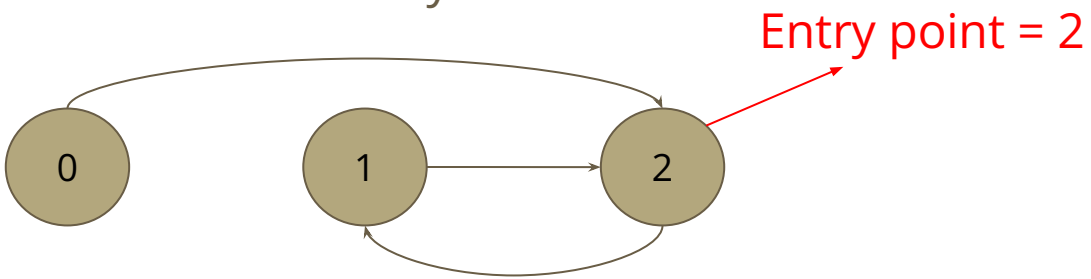
1. ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzaaa
2. pneumonoultramicroscopicsilicovolcanoconiosis
3. ababcbcbcbcbcb

- We will make sure that each pattern only has 1 answer.

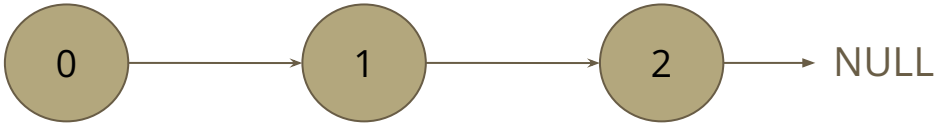
HW1_3: Linked List Cycle

In this question you need to judge whether a linked list contains a cycle, if there is a cycle you should determine where is the entry point.
There will be at most 94 nodes.

Contain cycle



No cycle



Files

```
CA_hw1/
├── b13901001_hw1
│   └── p3
│       ├── hw1_p3.s
│       ├── pattern0.txt
│       ├── pattern1.txt
│       └── pattern2.txt
└── image
```

- The files should be organized as the right figure
- Change the path below to test different pattern
- There will be 3 hidden patterns

```
47    .rodata
48        pattern: .string "../../{student_id}_hw1/p3/pattern0.txt"
49    .text
```

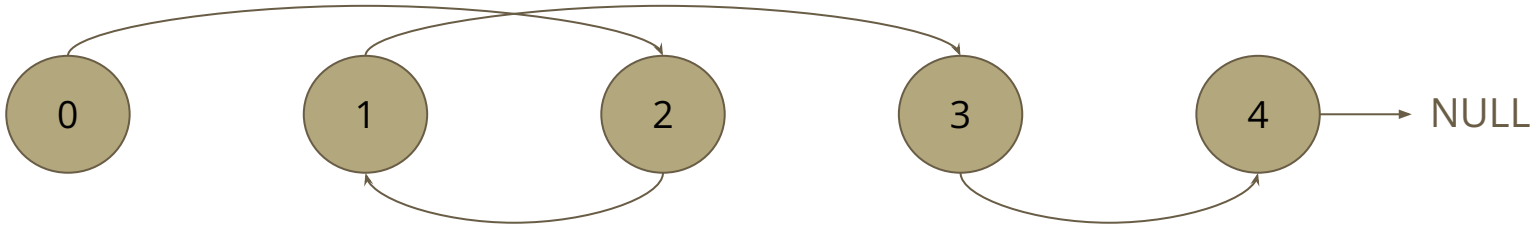
Example

Dec	Hx	Oct	Html	Chr
32	20	040		Space
33	21	041	!	!
34	22	042	"	"
35	23	043	#	#
36	24	044	$	$
37	25	045	%	%

Pattern: “#!$~

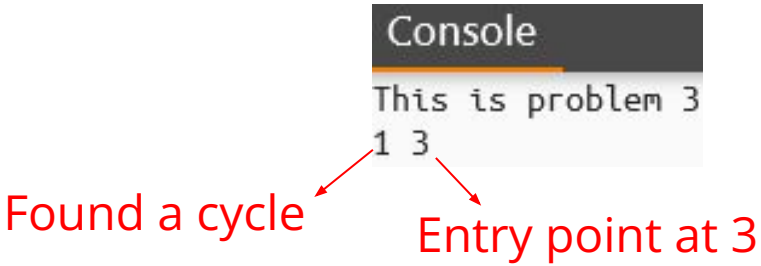
0x00010204	-32	-32	-32	94
0x00010200	4	1	3	2

- The linked list is stored at 0x10200
- The head is always at 0(0x10200)
- Each byte stores one pointer, which points to the next node
- We use 94 to represent null
- The pointer is ASCII encoded in pattern.txt and will be shifted 32 before being stored in the memory (Consider this only if you want to generate your own pattern)



Output

- Store whether you found a cycle in t0 (i.e. x5)
- Store the entry point in t1 (i.e. x6)



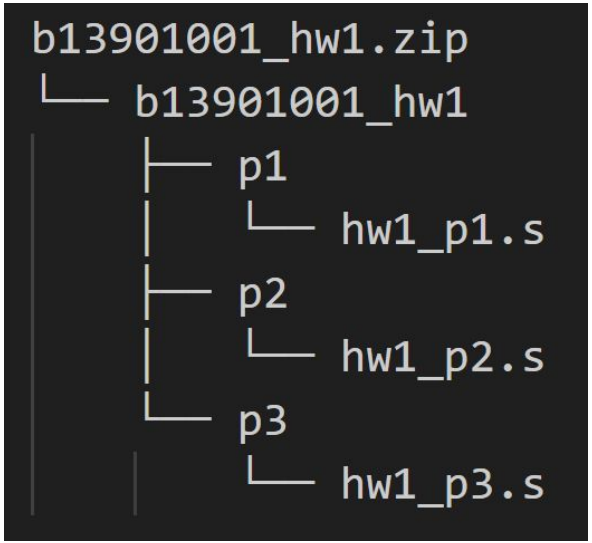
Grading

- HW1 takes up 5 points of your final score (100 points).
- You will get 1 point by passing all public patterns in each question
- You will get 1 point by passing all private patterns in each question.
- You will get at most 6 points if you pass all patterns in all three questions.

Number of patterns	p1	p2	p3
public	2	3	3
private	4	3	3

Submission

- Deadline:
  - 14:20 on Oct 22, 2024
  - Late submission will not be accepted
- Hand in a zip file on NTU COOL
- Your homework should be copied into a folder and packed into a zip file



Rules

- If you have any problems, ask your question through NTU COOL “Discussions”.
- No plagiarism.
- Do NOT modify any provided instructions.