



香港中文大學

The Chinese University of Hong Kong

CENG3420

Lab Overview & Introduction to RARS

Fangzhou Liu

Department of Computer Science & Engineering

Chinese University of Hong Kong

`{fzliu23}@cse.cuhk.edu.hk`

Spring 2026

- 1 Overview of CENG3420 Labs
- 2 RISC-V ISA Simulator – RARS

Overview of CENG3420 Labs

- Assembly language – symbolic
- Machine language – binary
- **Assembler** is a program that
 - turns symbols into machine instructions (e.g., riscv64-unknown-elf-as)
- **Simulator** is a program that
 - mimics the behavior of a processor
 - usually written in high-level language (e.g., spike)

We have 3 labs in total, with 2-3 sub-labs for each lab.

- **Lab1:** RISC-V assembly language programming using **RARS** simulator.
- In lab1, we will practice coding in RISC-V assembly language, and understand how our codes run in a RISC-V CPU.
 - **Lab1-1:** basic operators and system call.
 - **Lab1-2:** function call and simple algorithm implementation.
 - **Lab1-3:** stack data structure, recursive function call, more complex algorithm implementation.

We have 3 labs in total, with 2-3 sub-labs for each lab.

- **Lab2:** build(complete) a C-based RISC-V assembler and simulator.
- Codebase: <https://github.com/MingjunLi99/ceng3420>. We need to implement the assembler and simulator based on the codebase.
 - Lab2-1: implement a RISC-V assembler.
 - Lab2-2: implement a RISC-V ISA simulator with:
 - RISC-V 32 general-purpose registers
 - 32-bit data and address
 - 25+ instructions (including pseudo instructions)

We have 3 labs in total, with 2-3 sub-labs for each lab.

- **Lab3:** build a more complete C-based RISC-V Simulator based on lab2.
 - Lab3-1: control logic in CPU, finite state machine.
 - Lab3-2: execution model, memory interface.
 - Lab3-3: BUS driver, etc.

RISC-V ISA Simulator – RARS

- **RARS is the RISC-V Assembler, Runtime and Simulator for RISC-V assembly language programs.**
- We write codes in RISC-V assembly language, then **RARS** translates them into RISC-V instructions and corresponding machine codes, then execute the codes through simulation, like a RISC-V CPU.
- **RARS** supports RISC-V IMFDN ISA base (riscv32 & riscv64).
- **RARS** supports debugging using breakpoints like *ebreak*.
- **RARS** supports side by side comparison from psuedo-instruction to machine code with intermediate steps.

- RARS tutorial: <https://cass-kul.github.io/tutorials/rars/>
- Install Java environment: <https://java.com/en/download/>
- Download RARS:
<https://github.com/TheThirdOne/rars/releases/tag/continuous>
- Run RARS: run command `java -jar <rars jar path>` in the command window, under the path where you place `rars.jar`

```
cbai@hpc1:/research/dept8/gds/cbai/ta/rars$ java -jar rars.jar
```

- We also provide Java install package and RARS in **RARS.zip** on **Blackboard**.

RARS Overview

The screenshot displays the RARS (RISC Assembler RISC Assembler) interface. The main window is titled "hello3420.asm" and contains assembly code. The code is as follows:

```
1 .globl _start
2
3 .data # global variable declarations follow this line
4 welcome_msg: .ascii "Welcome to CENG3420!\n"
5
6 .text # instructions follow this line
7 _start: # a label, marks a position in the code
8     addi a0, x0, 1 # STDOUT=1
9     la a1, welcome_msg # Load the address of welcome_msg
10    addi a2, x0, 21 # Length of the string
11    addi a7, x0, 64 # Specify the system call number
12    ecall # Raise a system call
13    # End of program, leave a blank line afterwards is preferred
14
```

The interface includes a menu bar (File, Edit, Run, Settings, Tools, Help) and a toolbar with various icons. A status bar at the top right indicates "Run speed at max (no interaction)".

On the right side, the "Registers" panel is visible, showing a table of registers and their values:

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffffe0
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400000

The bottom panel shows the "Messages" tab with the following output:

```
Welcome to CENG3420!
— program is finished running (dropped off bottom) —
Reset: reset completed.
```

A "Clear" button is located next to the messages.

RARS Overview

The screenshot displays the RARS application window, which is divided into several panels:

- Tools panel:** Located at the top right, it contains a toolbar with various icons for file operations, editing, and execution. Below the toolbar is a status bar that reads "Run speed at max (no interaction)".
- Source codes panel:** The central area displays the assembly code for a test program. The code includes comments, directives, and instructions. The line numbers range from 88 to 111. The code is as follows:

```
88 # 12 "isa/rv64ui/rvliw.5" 2
89
90
91 .test
92 .globl _start
93 _start: nop
94
95 #-----
96 # Arithmetic tests
97 #-----
98
99 test_2: li x1, 0xffffffff80000000
100 srlw x14, x1, 0
101 li x7, 0xffffffff80000000
102 li x5, 2
103 bne x14, x7, fail
104
105 test_3: li x1, 0xffffffff80000000
106 srlw x14, x1, 1
107 li x7, 0x000000000400000000
108 li x5, 3
109 bne x14, x7, fail
110
111 test_4: li x1, 0xffffffff80000000
112
```
- Registers panel:** Located on the right side, it displays a table of registers. The table has three columns: Name, Number, and Value. The registers are listed in hexadecimal format. The registers are: x0, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, x14, x15, x16, x17, x18, x19, x20, x21, x22, x23, x24, x25, x26, x27, x28, x29, x30, x31, and x32. The values are all 0x0000000000000000.
- Program information panel:** Located at the bottom, it displays messages and run information. The messages include:

```
Assembly: assembling F:\Research\misc\TA\CBM32420\tools\test.asm
Parsing in F:\Research\misc\TA\CBM32420\tools\test.asm line 312 column 2: RARS does not recognize the .global directive. Ignored.
Parsing in F:\Research\misc\TA\CBM32420\tools\test.asm line 318 column 2: RARS does not recognize the .global directive. Ignored.
Assembly: operation completed successfully.
```

RARS Overview

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

☐ Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x00100513	addi x10, x0, 1	8: addi a0, x0, 1 # STDOUT=1
<input type="checkbox"/>	0x00400004	0x0fc10597	swipe x11, 0x0000fc10	9: la a1, welcome_msg # Load the address of welcome_msg
<input type="checkbox"/>	0x00400008	0xffe58593	addi x11, x11, 0xffffffff	
<input type="checkbox"/>	0x0040000c	0x01500613	addi x12, x0, 21	10: addi a2, x0, 21 # Length of the string
<input type="checkbox"/>	0x00400010	0x04000893	addi x17, x0, 0x00000040	11: addi a7, x0, 64 # Specify the system call number
<input type="checkbox"/>	0x00400014	0x00000073	ecall	12: ecall # Raise a system call

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+10)
0x10010000	0x636e6557	0x20656466	0x43206574	0x3347445	0x21303234	
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	

Registers Floating Point Control and Status

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffffe0
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400000

Messages Run I/O

Welcome to CER03420!

— program is finished running (dropped off bottom) —

Clear

Reset: reset completed.

RARS Overview

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Tools panel

Text segment panel

Dispt	Address	Code	Basic	Source
	0x040000	0x00000013	add r0,r0,0	91 _start: nop
	0x040004	0x00000013	add r1,r1,0xffff0000	99 test_2: li r1, 0xffffffff00000000
	0x040008	0x00000013	add r2,r2,0	
	0x04000c	0x00000013	add r3,r3,0	100: mliw r14, r1, 0
	0x040010	0x00000013	add r4,r4,0	101: li r7, 0xffffffff00000000
	0x040014	0x00000013	add r5,r5,0	
	0x040018	0x00000013	add r6,r6,0	102: li r0, 2
	0x04001c	0x3471c03	lsw r14, r1, 0x00003358	103: lsw r14, r1, fail
	0x040020	0x00000013	add r7,r7,0xffff0000	105 test_3: li r1, 0xffffffff00000000
	0x040024	0x00000013	add r8,r8,0	
	0x040028	0x00000013	add r9,r9,0	106: mliw r14, r1, 1
	0x04002c	0x00000013	add r10,r10,0	107: li r7, 0x0000000040000000
	0x040030	0x00000013	add r11,r11,0	
	0x040034	0x00000013	add r12,r12,0	108: li r0, 3
	0x040038	0x3271c03	lsw r14, r1, 0x00003358	109: lsw r14, r1, fail
	0x04003c	0x00000013	add r13,r13,0	111 test_4: li r1, 0xffffffff00000000

Data segment panel

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x101000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10100c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10101c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101024	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101028	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10102c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101030	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101034	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101038	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10103c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x101040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Registers panel

Registers	Floating Point	Control and Status	Name	Number	Value
r0				0	0x0000000000000000
r1				1	0x0000000000000000
r2				2	0x0000000000000000
r3				3	0x0000000000000000
r4				4	0x0000000000000000
r5				5	0x0000000000000000
r6				6	0x0000000000000000
r7				7	0x0000000000000000
r8				8	0x0000000000000000
r9				9	0x0000000000000000
r10				10	0x0000000000000000
r11				11	0x0000000000000000
r12				12	0x0000000000000000
r13				13	0x0000000000000000
r14				14	0x0000000000000000
r15				15	0x0000000000000000
r16				16	0x0000000000000000
r17				17	0x0000000000000000
r18				18	0x0000000000000000
r19				19	0x0000000000000000
r20				20	0x0000000000000000
r21				21	0x0000000000000000
r22				22	0x0000000000000000
r23				23	0x0000000000000000
r24				24	0x0000000000000000
r25				25	0x0000000000000000
r26				26	0x0000000000000000
r27				27	0x0000000000000000
r28				28	0x0000000000000000
r29				29	0x0000000000000000
r30				30	0x0000000000000000
r31				31	0x0000000000000000

Program information panel

Assembly: assembling F:\Research\misc\TA\CEB03420\tools\test.asm

Warning in F:\Research\misc\TA\CEB03420\tools\test.asm line 212 column 2: RARS does not recognise the global directive. Ignored.

Warning in F:\Research\misc\TA\CEB03420\tools\test.asm line 218 column 2: RARS does not recognise the global directive. Ignored.

Assembly: operation completed successfully.

- Create a new source file: Ctrl + N
- Close the current source file: Ctrl + W
- Assemble the source code: F3
- Execute the current source code: F5
- Step running: F7
- Instructions & System call query: F1

An Example Program

Hello CENG3420

```
.globl _start

.data    # global variable declarations follow this line
welcome_msg: .asciz "Welcome_to_CENG3420!\n"

.text    # instructions follow this line
_start: # a label, marks a position in the code
    addi a0, x0, 1 # STDOUT=1
    la a1, welcome_msg # Load the address of welcome_msg
    addi a2, x0, 21 # Length of the string
    addi a7, x0, 64 # Specify the system call number
    ecall # Raise a system call
# End of program, leave a blank line afterwards is preferred
```


An Example Program

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00109513	addi x10, x0, 1	9: addi a0, x0, 1
	0x00400004	0x0fc10597	auipc x11, 0x0000fc10	11: la a1, welcome_msg
	0x00400008	0xffc58593	addi x11, x11, 0xffff...	
	0x0040000c	0x01500613	addi x12, x0, 21	13: addi a2, x0, 21
	0x00400010	0x04000893	addi x17, x0, 0x00000040	15: addi a7, x0, 64
	0x00400014	0x00000073	ecall	17: ecall

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x00400000	0x00109513	0x0fc10597	0xffc58593	0x01500613	0x04000893	0x00000073	0x00000000	0x00000000
0x00400004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040000c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040001c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400024	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400028	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040002c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400030	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400034	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400038	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040003c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400044	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400048	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040004c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400050	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400054	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400058	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040005c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400064	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400068	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040006c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400070	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400074	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400078	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040007c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400084	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400088	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040008c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400090	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400094	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400098	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x0040009c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000a4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000a8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000ac	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000b0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000b4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000b8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000bc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000c4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000c8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000cc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000d0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000d4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000d8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000dc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000e4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000e8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000ec	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000f0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000f4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000f8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x004000fc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Control and Status

Registers Floating Point

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffffc
gp	3	0x10000000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
s0	10	0x00000015
a1	11	0x10010000
a2	12	0x00000015
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000040
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x0040001c

0x00400000 (.text) [X] Hexadecimal Addresses [X] Hexadecimal Values [] ASCII

Messages Run I/O

Welcome to ENG3420!

-- program is finished running (dropped off bottom) --

Clear

RARS provides a small set of operating system-like services through the system call (`ecall`) instruction. Register contents are not affected by a system call, except for result registers in some instructions.

- Load the service number (or number) in register `a7`.
- Load argument values, if any, in `a0`, `a1`, `a2` ..., as specified.
- Issue `ecall` instruction.
- Retrieve return values, if any, from result registers as specified.

System Calls in RARS

Name	Number	Description	Inputs	Outputs
PrintInt	1	Prints an integer	a0 = integer to print	N/A
PrintFloat	2	Prints a float point number	fa0 = float to print	N/A
PrintString	4	Prints a null-terminated string to the console	a0 = the address of the string	N/A
ReadInt	5	Reads an int from input console	a0 = the int	N/A
ReadFloat	6	Reads a float from input console	fa0 = the float	N/A
ReadString	8	Reads a string from the console	a0 = address of input buffer, a1 = maximum number of characters to read	N/A
Open	1024	Opens a file from a path Only supported flags (a1), read-only (0), write-only (1) and write-append (9)	a0 = Null terminated string for the path, a1 = flags	a0 = the file descriptor or -1 if an error occurred
Read	63	Read from a file descriptor into a buffer	a0 = the file descriptor, a1 = address of the buffer, a2 = maximum length to read	a0 = the length read or -1 if error
Write	64	Write to a file descriptor from a buffer	a0 = the file descriptor, a1 = the buffer address, a2 = the length to write	a0 = the number of characters written
LSeek	62	Seek to a position in a file	a0 = the file descriptor, a1 = the offset for the base, a2 is the beginning of the file (0), the current position (1), or the end of the file (2)}	a0 = the selected position from the beginning of the file or -1 is an error occurred

THANK YOU!