Start guide to CompSys A2

September 25, 2017

We have a *src* folder and a *architecture-tools* folder

First of all, be aware that we have a folder *src*, which is our assignment folder, and the reference simulator folder *architecture-tools*, which is from the public Github repository *compSys-e2017-sim*. At this stage, everyone shall be able to build in both folders. Then we shall have:

In *src* folder: only one executable file: **sim**.

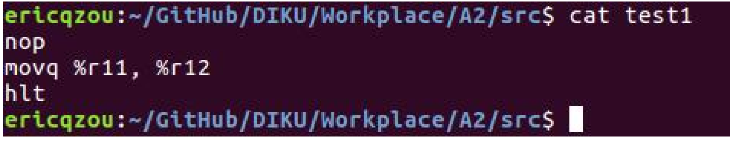
In *architecture-tools* folder: Three executable files: **asm**, **sim** and **trace**.

What is the difference and how they are related to A2

In the assignment text, one is asked to "validate that the handed out simulator is working properly" on the three instructions. Now we will use this as example to walk through the beginning stage.

So, first, we have an assembly file, call "test1". There is exactly three instructions, and very importantly, a newline at the end (which is not visible in the screenshot):

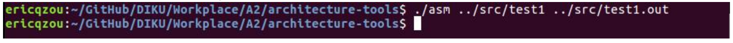
Figure 1: File *test1*



```
ericqzou:~/GitHub/DIKU/Workplace/A2/src$ cat test1
nop
movq %r11, %r12
hlt
ericqzou:~/GitHub/DIKU/Workplace/A2/src$ █
```

After that, this *test1* shall be assembled by the *asm* executable in folder *architecture-tools*, and get a output file, which we will name it to be "*test1.out*":

Figure 2: Generate *test1.out*



```
ericqzou:~/GitHub/DIKU/Workplace/A2/architecture-tools$ ./asm ../src/test1 ../src/test1.out
ericqzou:~/GitHub/DIKU/Workplace/A2/architecture-tools$ █
```

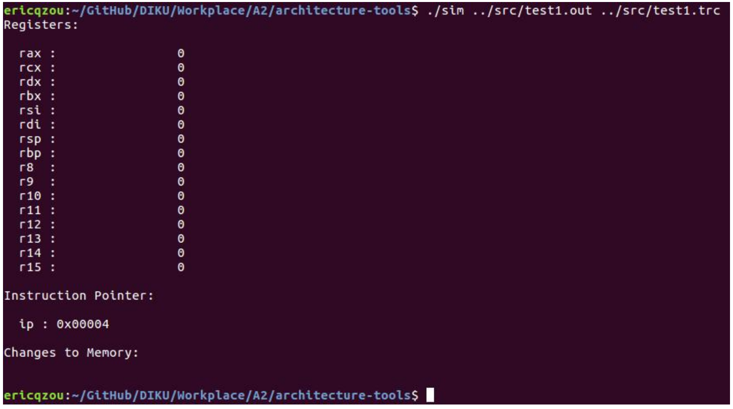
Note: this generated .out file is the one that can be run for both sim executables in both folder. Do not run your sim executable on the original assembly program file.

Now we are ready to generate the trace file by using *sim* **in the *architecture-tools* folder**. The difference between the two *sim* executables are:

- *sim* in *architecture-tools* folder **can generate** a trace file.
- *sim* in *src* folder do not generate, but **can use** the generated trace file for validation purpose.

Generating the trace file from *test1.out* to, say, *test1.trc*, will be like:

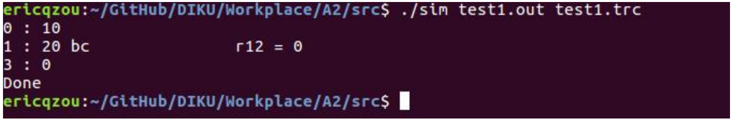
Figure 3: Generate *test1.trc*



```
ericqzou:~/GitHub/DIKU/Workplace/A2/architecture-tools$ ./sim ../src/test1.out ../src/test1.trc
Registers:

  rax :                0
  rcx :                0
  rdx :                0
  rbx :                0
  rsi :                0
  rdi :                0
  rsp :                0
  rbp :                0
  r8  :                0
  r9  :                0
  r10 :                0
  r11 :                0
  r12 :                0
  r13 :                0
  r14 :                0
  r15 :                0

Instruction Pointer:

  ip : 0x00004

Changes to Memory:


ericqzou:~/GitHub/DIKU/Workplace/A2/architecture-tools$ 
```

When the trace file is generated, one can run *sim* in **src folder** on *test1.out* and *test1.trc* to do the validation. If things are correctly implemented, no error message will occur, as the following screenshot:

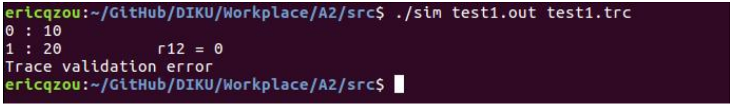
Figure 4: Validate using trace file



```
ericqzou:~/GitHub/DIKU/Workplace/A2/src$ ./sim test1.out test1.trc
0 : 10
1 : 20 bc                r12 = 0
3 : 0
Done
ericqzou:~/GitHub/DIKU/Workplace/A2/src$ 
```

If one's implementation has bugs, you will get a message while running the validation, at the stage where any of the validation flow fails:

Figure 5: Validation failed



```
ericqzou:~/GitHub/DIKU/Workplace/A2/src$ ./sim test1.out test1.trc
0 : 10
1 : 20          r12 = 0
Trace validation error
ericqzou:~/GitHub/DIKU/Workplace/A2/src$ 
```

Note: the validation part is also part of the assignment. It is not fully implemented.