Homework #4

TA in charge: Haejin Nam
E-mail: haejinnam@kaist.ac.kr

I. Goal of this assignment

- ✓ Understand what a computer architecture simulator is
- ✓ Understand what a benchmark is and how benchmarks are different to each other
- ✓ Execute a simulator, 'SimpleScalar' with a benchmark suite, 'SPEC2000'
- ✓ Analyze simulation results

II. Submission and grading

- ✓ You must submit all of followings **(Total 100 points)**:

  A. 6 simulation result files **(20 points)**:
  When the simulation completes, there will be six result files <apsi_result.txt, crafty_result.txt, gzip_result.txt, mcf_result.txt, parser_result.txt, swim_result.txt>
  You must submit all these result files without any modifications.

  B. Analysis report about simulation results **(80 points)**:
  Analyze characteristics of the six benchmarks based on the simulation result. *There is no specific format for this report.* You can use any information if it is in result files. You may also refer to any other literatures about SimpleScalar and SPEC2000. You could include tables or figures to complement your analysis.
  * Be cautious with the exact meaning of each value presented in result files.
  e.g. 'sim_elapsed_time' represents the time elapsed on the system where you are running the simulation; not the time elapsed in the *simulated system*. Hence, it is irrelevant with the performance of the *simulated system*.
  * You don't have to understand all the values presented in the simulation result since many of them are not even covered in this course. But try to analyze the simulation results *with your own words* based on the knowledge covered in this course.

  The report file name must be <Report_StudentID_NAME.pdf>.
  e.g. Report_20201234_JohnSmith.pdf
  You can use either English or Korean in the report as you want.

- ✓ *Compress all these files into a .zip file* and upload it on KLMS.
- ✓ The submitted file name must be <HW4_StudentID_NAME.zip>.

e.g. HW4_20201234_JohnSmith.zip

- ✓ Homework #4 is an individual assignment. You can talk and discuss to each other, but *you cannot simulate benchmarks or write a report with others*.

III. Due date

- ✓ **Nov. 8th (Tue) 23:59:00**
- ✓ **Late** submission due date: Nov. 9th (Wed.) 23:59:00
- ✓ If you miss the due date, there will be 50% deduction from your total score.
- ✓ **You cannot submit after the late submission due date**.

IV. Cheating

- ✓ If there are any cheatings in your submission, you will get 0 points.
- ✓ *Followings will be regarded as cheating*:
  - A. Copying other students' simulation results or reports
  - B. Modifying other students' results and using them as if they were your own
  - C. Using other sources without any references excluding your own simulation results
  - D. All other sorts of inappropriate behaviors

V. Prerequisite

- ✓ Follow the guide <Environment_setup_guide.pdf> to setup the simulation environment for this (and later) homework.

VI. Simulator and benchmark

- ✓ Simulator: SimpleScalar
  'SimpleScalar' is one of the classical simulators used in computer architecture research. With this simulator, you can simulate a CPU based computer system and evaluate/estimate its performance for various program applications.
  If you want to know about 'SimpleScalar' simulator, please check out http://www.simplescalar.com/ .

✓ Benchmark suite: SPEC CPU2000

Basically, benchmarks are a kind of testing programs to quantify relative performance of various computing systems. 'SPEC CPU' is one of the most representative benchmark suites, including integer-incentive workloads and floating-point-incentive workloads. In this homework, you will use 6 benchmarks selected from SPEC CPU2000.

A. apsi
B. crafty
C. gzip
D. mcf
E. parser
F. swim

If you want to know more about 'SPEC CPU2000', please read <SPEC2000.pdf> uploaded on KLMS.

VII. Execution example

✓ After setting up simulation environment with <Environment_setup_guide.pdf>, you should have an access to 'simplesim-3.0' directory with a terminal.

✓ This example assumes you have setup the simulation environment using the *provided VM image*. But the flow should mostly be the same for any other environments.

✓ This example also assumes that you are not familiar with Unix/Linux shell.

✓ As following this execution example, you can simulate benchmarks and get 6 result files.

A. Open a new terminal and change the current directory to 'simplesim-3.0'.
You can do this by *typing the command below*, *next to the dollar sign* ($).

```
cs311@cs311:~$ cd simplesim-3.0
```

'cd' is a command to change the current directory.
'simplesim-3.0' is a directory name, which contains SimpleScalar source code files, tool scripts for the simulator, SPEC CPU2000 benchmark suite directory 'SPEC2000', and a script for this homework 'hw4.sh'.

After typing the command, press 'Enter'. Then you will see a new sentence on the terminal as follows.

```
cs311@cs311:~/simplesim-3.0$
```

*** If you didn't use the provided VM image**, change current directory to the 'simplesim-3.0' directory extracted from the archive file.

B. Build SimpleScalar simulator by *typing the command below.*

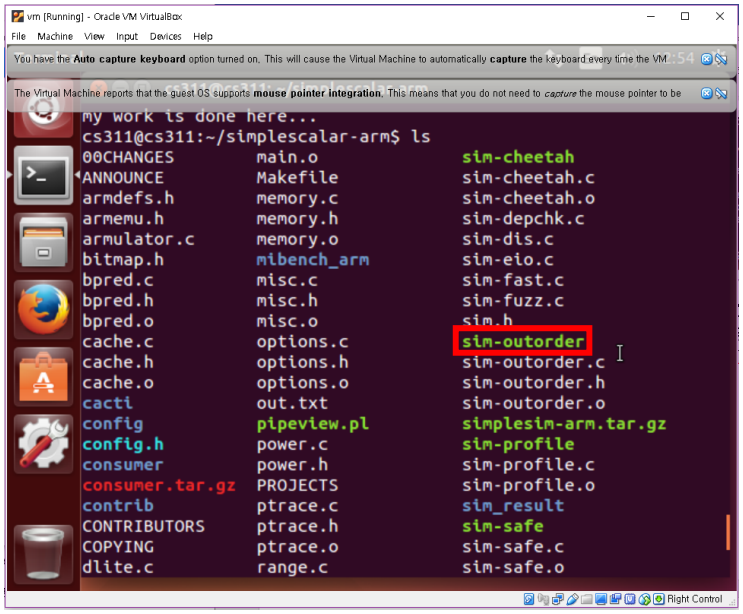
```
cs311@cs311:~/simplesim-3.0$ make all
```

This should not fail if you are using the provided VM image and have correctly followed the guide. This will generate several *executable files*.

C. List current directory by *typing the next command.*

```
cs311@cs311:~/simplesim-3.0$ ls
```

You will see several *executable files* within the listed files. Executable file names are colored in light green.
In HW4, you will use 'sim-outorder' executable file to run simulations.



D. Before starting simulations for HW4, we recommend you a test run of 'sim-outorder'.
Type the command below and press 'Enter' to run a simulation. The command is quite long, so please type carefully to not make typos. Be cautious if you copy and paste, since directly copying from a PDF file can deform the text.

```
cs311@cs311:~/simplesim-3.0 $ ./sim-outorder \
-redir:sim sim_result/swim_result.txt \
-max:inst 10000000 SPEC2000/spec2000binaries/swim00.peak.ev6 \
 < SPEC2000/spec2000args/swim/swim.in > /dev/null
```

Be careful about several dots (.), bars (-), and under-bars (_). There are 7 zeros. With Korean keyboard, you can write back-slash (\) as 'won (₩)' key. *When you press 'Enter' after writing back-slash, the terminal automatically makes inequality sign (>)*. It just means that the command is not finished yet. Ignore it, and write the next command line. Check twice you wrote every single character

including spaces correctly.

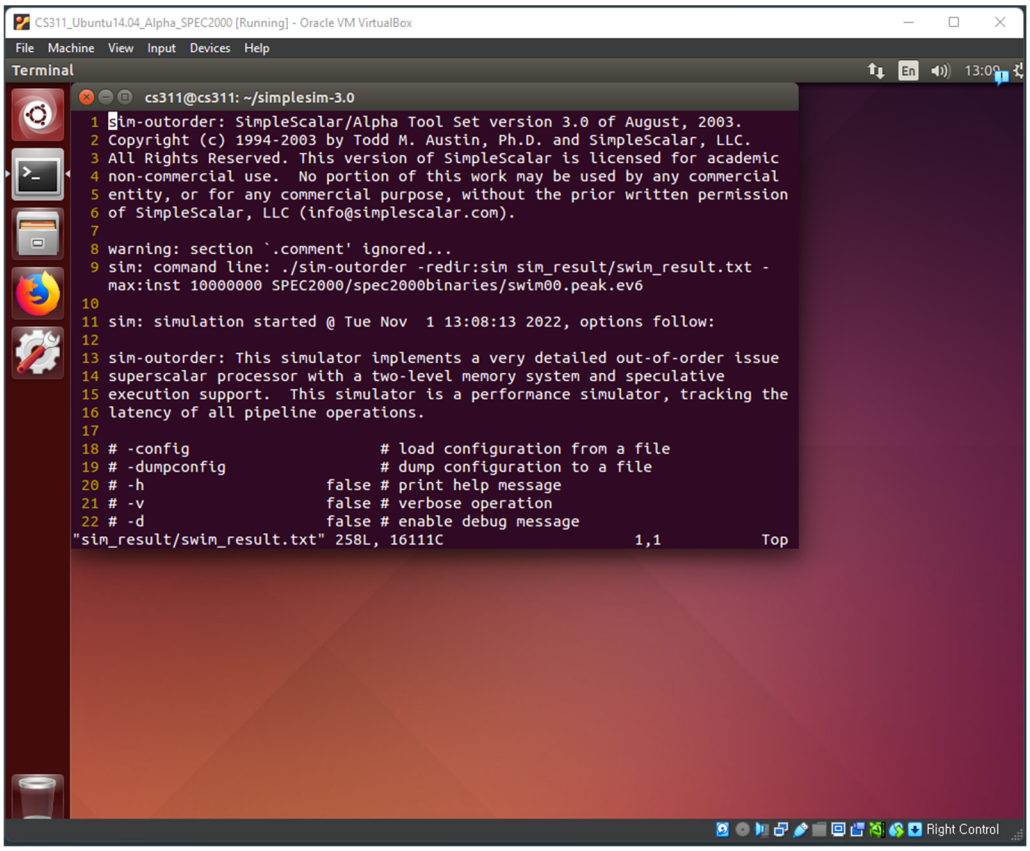
Wait until 'cs311@...' line is shown again.

E. When the simulation completes, *type the command* below to check the simulation result.

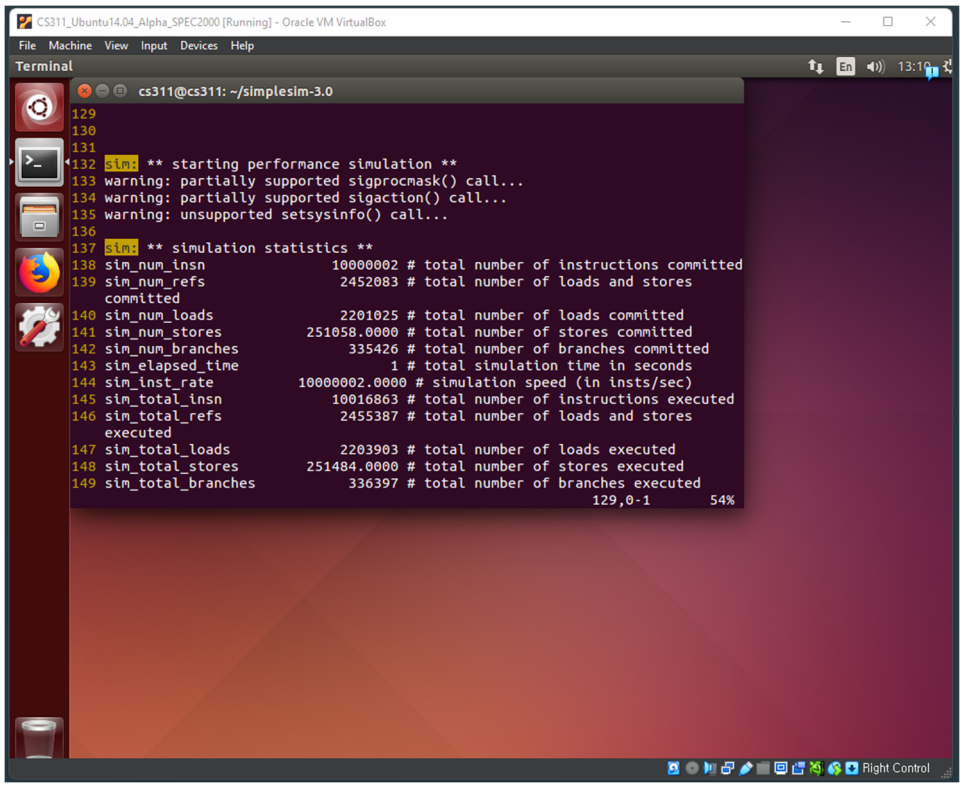
```
cs311@cs311:~/simplesim-3.0$ vim sim_result/swim_result.txt
```

'vim' is one of text editors like 'notepad' in Windows. You can use any other text editors if you prefer.

You can see contents of 'swim_result.txt' text file as follows.



The first several lines are explanation of simulator configuration. You can scroll down and find the sentence, “sim: ** starting performance simulation **”.



After that line, it shows performance simulation results. *Each line is composed of three parts*.

Statistics value name	Measured value	# Meaning of value

F. You need to use these simulation results to write the analysis report. After confirming simulation results, *you can exit ‘vim’ editor with the command*.

```
:q
```

When you type the command, it shows the last line of the terminal. If you press ‘Enter’, ‘vim’ editor will be closed and the terminal will be back to the state before starting ‘vim’ editor.

If you are new to vim, try command ‘**vimtutor’** on the terminal. It will give you a simple, but practical tutorial.

G. If there are no issues with the simulation result, it means there are no problems in the simulator. Now, you can simulate with the selected six benchmarks. *There are two ways to simulate these benchmarks*.

1. Simulate other five benchmark as described earlier

   By using commands similar to the one used earlier, other benchmarks can be run in a similar way. But it could be a little bit bothersome since each benchmark requires different arguments. For example, while *swim* reads

input from the standard input (using '<' operator), *gzip* just take them as arguments. This means you have to examine which benchmark needs which arguments and how to pass them correctly. Furthermore, typing commands one by one may result in fat-finger errors. Thus, we highly recommend the way below:

2. Run a batch script to run simulation for all the six benchmarks at once

   If you want to simulate the six benchmarks at once, you can use the shell script included in the 'simplesim-3.0' directory, 'hw4.sh'.

   A shell script is a file which has a set of shell commands. *You can execute commands in the script by calling the shell script as follows*.

```
cs311@cs311:~/simplesim-3.0$ ./hw4.sh
```

   This way is far less error-prone, but you would need to wait a little longer until all the simulations complete.
   Even though it simulates at once, simulation results are not affected. (In fact, this shell script just sequentially executes the six simulations.)

H. After the end of simulation, the result files will be saved into 'sim_result' directory. *You can go to 'sim_result' directory, and confirm results with 'vim' editor*.
   There should be six simulation results named:
   <apsi_result.txt, crafty_result.txt, gzip_result.txt, mcf_result.txt, parser_result.txt, swim_result.txt>.

```
cs311@cs311:~/simplesim-3.0$ cd sim_result
cs311@cs311:~/simplesim-3.0/sim_result$ vim <benchmark>_result.txt
```

VIII. Tips

- ✓ Please read this guideline carefully if you are not familiar with Linux. Most of you can do HW4 without any difficulties simply following the instructions in this guide.
- ✓ There are some red-highlighted words and *italic sentences* in this guideline. They are very important information for this homework. Do not ignore these highlighted contents.
- ✓ If you have any questions, please post them on *KLMS QnA board.* Both English and Korean are fine.
- ✓ In HW6 you will need to use the same 'SimpleScalar' simulator and the environment. Hence, we recommend you to complete this homework and keep the environment.