

Computer Organization Project #1

TA in charge: Junoh Moon
E-mail: junoh.moon@kaist.ac.kr

I. Goal of this project

- ✓ 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. 5 simulation result files (**20 points**):

After appropriate simulation, there will be five result files
<gcc_result.txt, mcf_result.txt, bzip2_result.txt, swim_result.txt, art_result.txt>.
You must submit all these results files without any modifications.
 - B. Analysis report about simulation results (**80 points**):

With simulation result files, you will analyze characteristics of five benchmarks in terms of the number of instructions, IPC, memory utilization, etc. *There are not any specific formats for the report.* You can use any information if it is in result files. You can include anything such as a table, graph, chart, screen captures, etc.

The report file name must be <Report_StudentID_NAME.pdf>.
ex) Report_20203214_JunohMoon.pdf
You can use either English or Korean as you want.
- ✓ *Compress all these files into a .zip file and upload the compressed file on KLMS.*
- ✓ The submitted file name must be <Project1_StudentID_NAME.zip>.
ex) Project1_20203214_JunohMoon.zip
- ✓ Project #1 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

- ✓ The end of Nov. 12 (Thur.)
- ✓ Late submission due date: the end of Nov. 13 (Fri.)
- ✓ After the due date, there will be **50% penalty on your project #1 score**.
If you submit after the due date, your maximum score will be 50 points.
- ✓ After the late submission due date, you will get **0 point**.

IV. Cheating

- ✓ If there are any cheatings in your submission, you will get **0 point**.
- ✓ *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

- ✓ Since most of students do not use Linux operating system, we will give you a Linux image via KLMS, and you will use 'VirtualBox' to run a Linux image.
- ✓ *You can follow another guideline, <[linux_image_setup_guide.pdf](#)> to install 'VirtualBox', unzip the Linux image, and execute it on 'VirtualBox', but if you are not new to Linux environment, you can copy the workspace from the image into your environment.*
- ✓ TAs have already confirmed that the Linux image works well if you follow the guideline correctly.
- ✓ If there are some problems even though you follow the guideline, please contact to TA.

VI. Simulator and benchmark

- ✓ Simulator: SimpleScalar
'SimpleScalar' is one of the famous simulators used in computer architecture. With this simulator, you can build a modeling application for program performance analysis.
If you want to know about 'SimpleScalar' simulator, please check out <http://www.simplescalar.com/>.

✓ Benchmark suite: SPEC2000

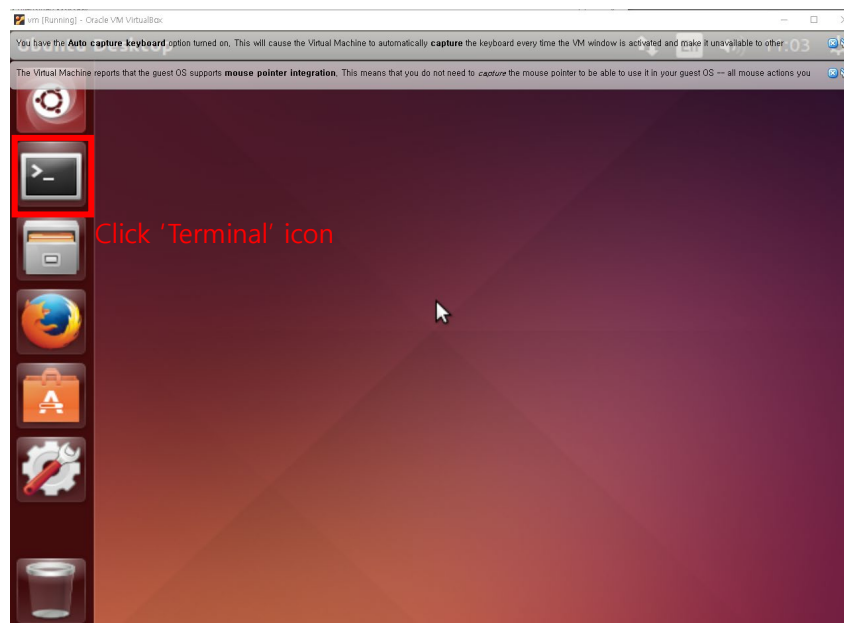
Basically, benchmarks are a kind of testing programs to quantify relative performance of various computing systems. ‘SPEC2000’ is one of the most famous benchmark suites, including integer-intensive jobs and floating-point-intensive jobs. In this project, you will use 5 benchmarks among them.

- A. Gcc
- B. Mcf
- C. Bzip2
- D. Swim
- E. Art

If you want to know more about ‘SPEC2000’, please read <[SPEC2000.pdf](#)> uploaded on KLMS.

VII. Execution example

- ✓ After setting simulation environment with <[linux_image_setup_guide.pdf](#)>, you may be able to use Linux Ubuntu 14.04 operating system on the virtual machine. In the Linux image, *there is pre-installed ‘SimpleScalar’ with ‘SPEC2000’.*
- ✓ As following this execution example, you can simulate benchmarks and get 5 result files.
 - A. If you follow <[linux_image_setup_guide.pdf](#)> appropriately, you may see a screen as below.
 - B. *Click left ‘Terminal’ icon or press ‘Ctrl’ + ‘Alt’ + ‘t’ to open new terminal.*



- C. New terminal will be open. If there are not any problems, it shows the followed

sentence.

```
cs311@cs311:~$
```

It means that your user name is 'cs311' and now you are in a home directory, '~'.

- D. Next to the dollar mark, '\$', write the followed command.

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

'cd' is a command to change the current directory. 'simplesim-3.0' is a directory name, which has pre-installed 'SimpleScalar' simulator.

After writing the command, press 'Enter'. Then, the terminal shows new sentence as follows.

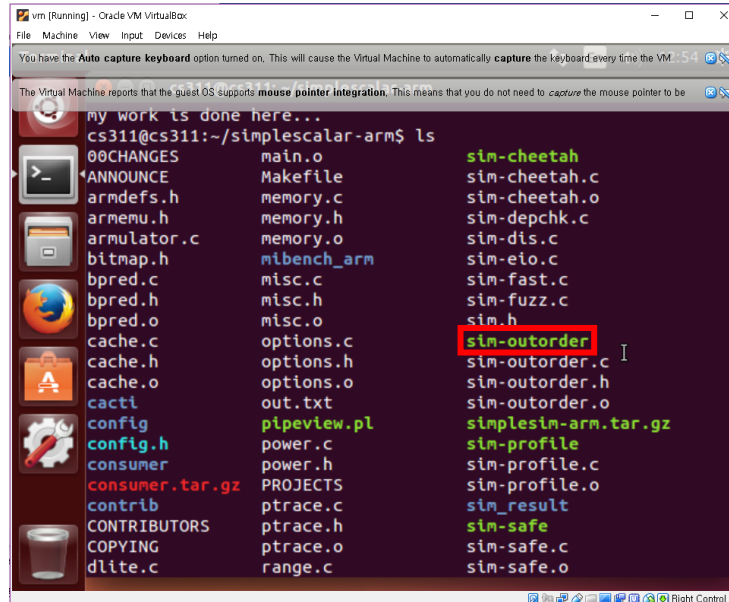
```
cs311@cs311:~/simplesim-3.0$
```

- E. write the command and press 'Enter'.

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

- F. If you use 'ls' command, you will see several executable files. Executable files are shown as light green.

In project #1, you will use 'sim-outorder' executable file for simulation. Please check 'sim-outorder'.



```

cs311@cs311:~/simplesim-3.0$ ls
main.o          Makefile        sim-cheetah
armdefs.h       memory.c        sim-cheetah.o
armemu.h        memory.h        sim-depchk.c
armulator.c     memory.o        sim-dis.c
bitnap.h        nibench_arm     sim-eio.c
bpred.c         misc.c          sim-fast.c
bpred.h         misc.h          sim-fuzz.c
bpred.o         misc.o          sim.h
cache.c         options.c       sim-outorder
cache.h         options.h       sim-outorder.c
cache.o         options.o       sim-outorder.h
cacti          out.txt        sim-outorder.o
config         pipeview.pl    simplesim-arm.tar.gz
config.h       power.c        sim-profile
consumer       power.h        sim-profile.c
consumer.tar.gz PROJECTS       sim-profile.o
contrib        ptrace.c       sim_result
CONTRIBUTORS  ptrace.h       sim-safe
COPYING        ptrace.o       sim-safe.c
dlite.c        range.c        sim-safe.o
  
```

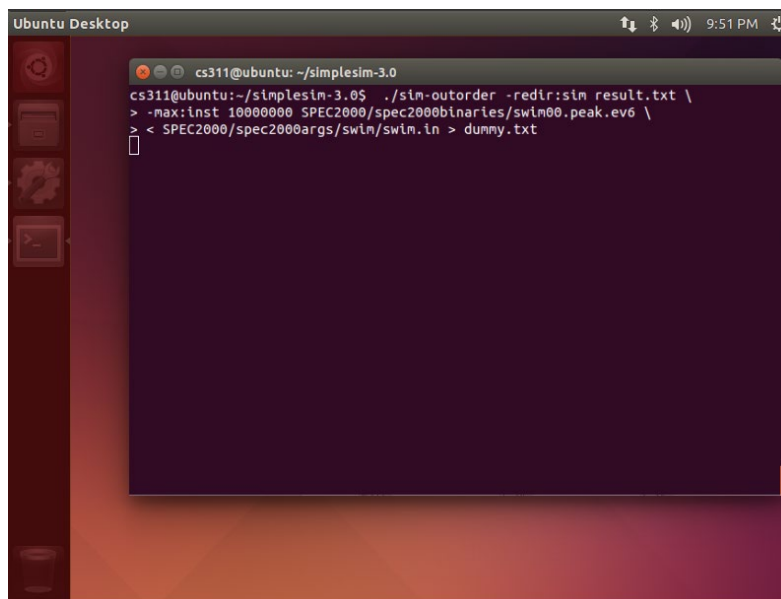
- G. Before starting simulation for project #1, you should test 'sim-outorder' whether it works well or not.

Write the command below, and press 'Enter' to test. The command is quite long, so you can easily make some errors. Please write carefully, and do not just copy and paste. Copying sentences from PDF file can make some duplicate words.

```
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.

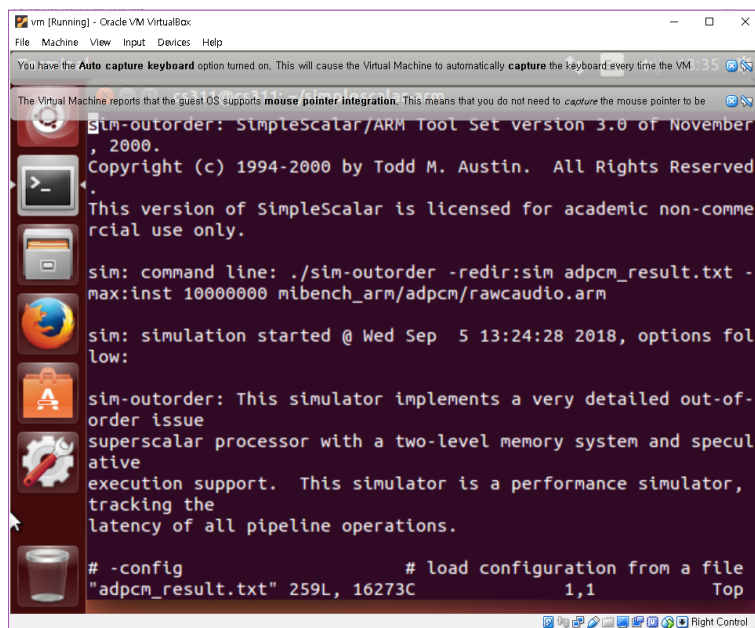
After writing all commands, you should have the following.



- H. If you press 'Enter', the cursor moves the next line and nothing is occurred. *Do not do anything, and wait until 'cs311@...' line is shown again.* During this time, 'SimpleScalar' simulates with 'SPEC2000' based on the command. If you do something during that time, a simulation result will be wrong.
- I. After testing simulation, write the command to check the simulation result.

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

'vim' is one of famous text editors like 'notepad' in Windows. You can see contents of 'swim_result.txt' text file as follows.



```

sim-outorder: SimpleScalar/ARM Tool Set version 3.0 of November
, 2000.
Copyright (c) 1994-2000 by Todd M. Austin. All Rights Reserved
.
This version of SimpleScalar is licensed for academic non-comme
rcial use only.

sim: command line: ./sim-outorder -redir:sim adpcm_result.txt -
max:inst 10000000 mibench_arm/adpcm/rawaudio.arm

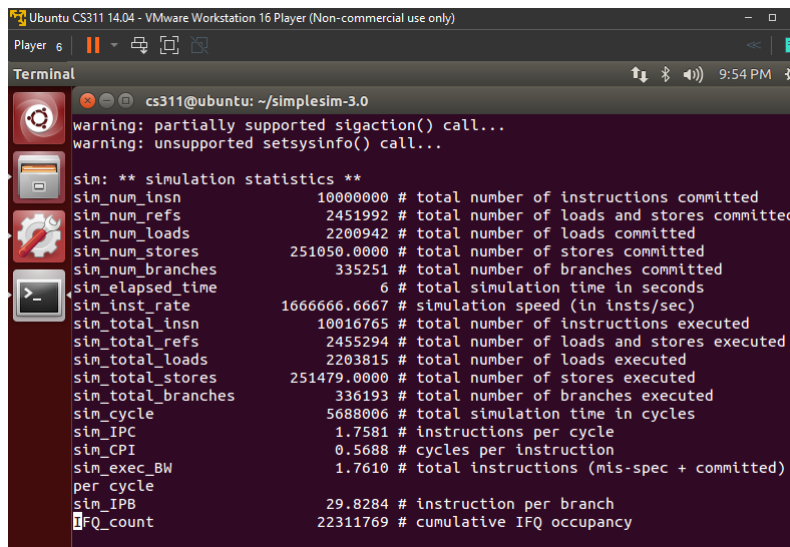
sim: simulation started @ Wed Sep 5 13:24:28 2018, options fol
low:

sim-outorder: This simulator implements a very detailed out-of-
order issue
superscalar processor with a two-level memory system and specul
ative
execution support. This simulator is a performance simulator,
tracking the
latency of all pipeline operations.

# -config                # load configuration from a file
"adpcm_result.txt" 259L, 16273C                1,1        Top

```

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



```

cs311@ubuntu: ~/simplesim-3.0
warning: partially supported sigaction() call...
warning: unsupported setsysinfo() call...

sim: ** simulation statistics **
sim_num_insn      10000000 # total number of instructions committed
sim_num_refs      2451992 # total number of loads and stores committed
sim_num_loads      2200942 # total number of loads committed
sim_num_stores     251050.0000 # total number of stores committed
sim_num_branches   335251 # total number of branches committed
sim_elapsed_time    6 # total simulation time in seconds
sim_inst_rate     1666666.6667 # simulation speed (in insts/sec)
sim_total_insn     10016765 # total number of instructions executed
sim_total_refs     2455294 # total number of loads and stores executed
sim_total_loads     2203815 # total number of loads executed
sim_total_stores    251479.0000 # total number of stores executed
sim_total_branches  336193 # total number of branches executed
sim_cycle          5688006 # total simulation time in cycles
sim_IPC            1.7581 # instructions per cycle
sim_CPI            0.5688 # cycles per instruction
sim_exec_BW        1.7610 # total instructions (mis-spec + committed)
per cycle
sim_IPB            29.8284 # instruction per branch
IFQ_count          22311769 # cumulative IFQ occupancy

```

After that line, it shows performance simulation results. *Results are composed of three parts.*

Simulation parameter	(Blank)	Simulation result	#	Explanation
----------------------	---------	-------------------	---	-------------

- J. You can use these simulation results to analyze and write a report. After confirming simulation results, *you can exit ‘vim’ editor with the command.*

```
:q
```

When you write 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 running ‘**vimtutor**’ on the terminal. It will give you a simple, but practical tutorial.

- K. If you confirm the simulation result, it means there are not any problems on the simulator. Now, you can simulate with selected 5 benchmarks. *There are two ways to simulate these benchmarks.*

※ Real simulations needs much longer time than the testing simulation done before. Please confirm again whether you can simulate now or not.

1. Simulate other four benchmark as described before

Similar to the command mentioned just before, other benchmarks can be run in a similar way. The problem is, however, each benchmark has different arguments. For example, swim reads input from the standard input (< operator), bzip2 just reads arguments in contrary. 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

Simulate five benchmarks at once with a shell script:

If you want to simulate five benchmarks at once, you can use a shell script, ‘proj1.sh’. A shell script is a file which has a set of commands. *You can use same commands with only calling the shell script as follows.*

```
cs311@cs311:~/simplesim-3.0$ ./proj1.sh
```

It is better because there is less possibility to make errors, but you must wait much longer since it simulates five benchmarks at once.

Even though it simulates at once, simulation results are not affected. *There are not any differences in results, so do not worry about using it.*

- L. After the end of simulation, the result files are stored in ‘sim_result’ directory. *You can go to ‘sim_result’ directory, and confirm results with ‘vim’ editor.*

There will be five simulation results named:

<gcc_result.txt, mcf_result.txt, bzip2_result.txt, swim_result.txt, art_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 all this guideline carefully. Most of you can do project #1 without any difficulties if you read it carefully.
- ✓ There are some **red-highlighted words** and *italic sentences* in this guideline. They are very important information for your project #1. Do not ignore these highlighted contents.
- ✓ If you have some questions, please contact to the charge TA. *You can ONLY use KLMS QnA board.* Both English and Korean can be used for a question.
- ✓ Project will continuously use 'SimpleScalar' and 'SPEC2000'. If you fail project #1, it will affect to project #2 and #3. Project #1 is not quite difficult, so please do not give up!