ECE 4100 / ECE 6100 / CS 4290 / CS 6290
Advanced Computer Architecture
Lab 1: Analyzing Benchmark Traces and Computing CPI (5pts)
Due: Friday, August 26th, 2016 (1pm)

This is an individual assignment. You can discuss this assignment with other classmates but you should code your assignment individually. You are NOT allowed to see the code of (or show your code to) other students.

OBJECTIVE: The objective of the first lab assignment is to test the proficiency of students in doing programming-based assignments and to ensure that the students have the basic background in computer architecture to take this graduate level course. The assignment is due before the Phase II registration deadline (Friday), so that the students can make well-informed decisions whether they should continue with the course or should consider taking it after getting the required pre requisites. We will provide an autograder for this assignment, so that you can estimate the score your submission is likely to receive at the time of submitting.

PROBLEM DESCRIPTION: The first lab assignment is aimed at doing analysis of the static and dynamic occurrences of instructions in a given benchmark trace. We will provide a code template and traces from four benchmarks (gcc, mcf, libquantum, and bzip2) selected from the SPEC CPU2006 suite. We will also provide a trace reader for these traces. Your job is to do the following:

Task 1: Quantify the mix of the dynamic instruction stream. The instructions in the trace are classified as five types: ALU, LOAD, STORE, CBR (Conditional Branch), and OTHER. You will modify the provided code to count the number of dynamic instructions for each category and the percentage of these instruction types in the instruction mix.

Task 2: Estimate the overall CPI using a simple CPI model in which the CPI for each category of instructions is provided. We will use the following CPI for each category:
1) ALU: 1
2) Load: 2
3) Store: 2
4) CBR: 3
5) Other: 1

Note: As the instruction mix for each trace is likely to be different, the CPI for each trace is likely to be different also.

Task 3: Estimate the instruction footprint by counting the number of unique PCs in the benchmark trace (ideally, this information should be multiplied by the average bytes per instruction to get the total footprint, however for this lab we will forego this multiplication).

HOW TO GET STARTED:

1. Download the tarball (Lab1.tar.gz) from T-square
2. “tar -xvzf Lab1.tar.gz”
3. “cd Lab1”
4. “ls” -- Lab1 contains four subdirectories: src, scripts, traces, and results
5. “cd src”
6. “ls” (there are three files: makefile, trace.h, sim.cpp)
7. Open sim.cpp in your favorite editor -- we have provided the trace reader and print functions for you already. Your job is to simply write the function: analyze_trace_record() and update the stats variable (stat_*)
8. Once you write the function, type “make” … this should create an executable “sim”
9. “./sim ../trace/bzip2.otr.gz” (to run one trace and see the output)
10. Do sanity check if the output makes sense (or if you need to debug your code)
11. Once your code is ready, go to the scripts directory “cd ../scripts”
12. do “./runall.sh” -- this runs your code for all four traces, stores the result files in the results directory and also generates the “Report.txt” file.
13. The last line of the report file is your approximate score out of 5 points. Note that we will run your source code on a separate set of “hidden” traces, so your graded score may be different if your implementation is not correct.

WHAT TO SUBMIT: Two files ONLY: Your **sim.cpp** and **report.txt**

REFERENCE MACHINE:
ECE STUDENTS: We will use **ecelinsrv7.ece.gatech.edu** as the reference machine for this course. (http://www.ece-help.gatech.edu/labs/unix/names.html).

CS STUDENTS: We will use **shuttle3.cc.gatech.edu** as the reference machine for this course. (https://support.cc.gatech.edu/facilities/general-access-servers)

Before submitting your code ensure that your code compiles on this machine, and generates the desired output (without any extra printf statements). Please follow the submission instructions. If you do not follow the submission file names, you will not receive the full credit.

NOTE: It is impractical for us to support other platforms such as Mac, Windows, Ubuntu etc.

HOW WELL YOU DID?: If you have the right background, this lab should take about 1-2 hours to complete. If it takes you 5-10 hours or more to do this lab assignment, you may not have the right background for this course. Note: Lab2, Lab3, and Lab4 will each likely take about 10x more time and are about 10x harder. So, if you do not have the right background, you may find the assignments for this course to be extraordinarily difficult.

FAQ:

1. **/usr/bin/ld: cannot find -lz collect2: error: ld returned 1 exit status**
   Please remove the '-lz' flag from the Makefile or install zlibc on your system

2. **How do I connect to ecelinsrv7.ece.gatech.edu/shuttle3.cc.gatech.edu?**

   a. You will need Cisco Anyconnect (for VPN access to any machine at Gatech) http://anyc.vpn.gatech.edu
   b. Putty SSH for doing ssh from your machine to ecelinsrv7.ece.gatech.edu/shuttle3.cc.gatech.edu
   c. SCP for file transfers to/from the server. There are several scp GUI tools available or you can directly execute the command from the terminal.