**CSE-443/543: High Performance Computing**

**Lab #03**

Max Points: 42

# Objective:

The objective of this exercise is to:

* Build familiarity with compiling code at the command line and ensuring that code adheres to standards
* Determine if there is a measurable performance difference between two semantically equivalent coding approaches, in this case a switch statement versus the equivalent if-else construct.
* Gain experience reporting on the results of a microbenchmark
* Learn how to get more ‘helpful’ output from the /usr/bin/time command by using appropriate command line options.

Fill in answers to all of the questions. For almost all the questions you can simply copy-paste appropriate text from the shell/output window into this document or the corresponding spreadsheet. You may discuss the questions with your instructor.

# Background

[In Session 05 in class](https://docs.google.com/presentation/d/1cVng9Kc428IPKdsKLc5Whl1rCqD2eNS9pHHfbfzeoQ4/edit?usp=sharing) we talked about performing microbenchmarks to compare specific functionality of a program or system, including the proper ways to conduct and report on such a benchmark. In this lab assignment, you will be studying two ways to implement flow control in a program and will determine which is faster.

Recall that C++ provides two different approaches to implement control flow, using either a series of if-else statements, or a switch statement. In C++, the switch statement can be used only with primitive data types (such as int), which could provide the compiler an opportunity to optimize the switch statement. Often there are questions and discussions about the performance difference between if vs. switch. (Search [stackoverflow.com](https://stackoverflow.com/) if you are interested in seeing these discussions.)

Based on this, we will make the hypothesis that when a switch statement is used for a contiguous range of int values (say 0…9) the compiler can better optimize the switch statement when compared to a semantically equivalent series of if-else statements. Your job is to verify whether this is true and back up your conclusion with timing data and appropriate statistical analyses. The result can depend on compiler versions and optimization levels, so you’ll need to do the experiment to answer this question!

For this purpose, I am giving you code that performs a computation either via if-else or switch statements depending upon the value of a command line argument you supply. One catch with this code – it does not adhere to the CSE coding standards and as part of this lab **you will need to fix it** so that it passes a cpplint.py check with no errors!

You will be using the /usr/bin/time command to determine the execution time of your program. Recall that the output of /usr/bin/time looks similar to this:

0.13user 0.22system 0:15.83elapsed 2%CPU (0avgtext+0avgdata 4180maxresident)k

0inputs+0outputs (0major+304minor)pagefaults 0swaps

While this output is informative, it cannot be easily copied into a spreadsheet for processing. It is possible to change the output format to make this much easier. You should review the [man page for time](https://linux.die.net/man/1/time) to determine how to change the output format of /usr/bin/time so that the user, system and elapsed time are the only things printed and that they are given in seconds (i.e. without any colons). For example:

0.13 0.22 15.83

# Procedure

1. Download the file Lab02.cpp from Canvas to your workstation.
2. Run cpplint.py to identify problems with the code. You should fix these problems, i.e. ensure that your code will pass a cpplint.py test before going further.
3. Study the starter code and note the following key aspects
   1. The useSwitch method implements the logic using switch statements.
   2. The useIf method implements the same functionality using a series of if-else statements.
4. Write a bash shell script called Lab02.bash that does the following:
   1. Invokes /bin/bash as a login shell.
   2. Prints a single line of “model name” information from /proc/cpuinfo
   3. Prints a single line of “MemTotal” information from /proc/meminfo
   4. Runs cpplint.py against the source code (Lab02.cpp)
   5. Compiles the Lab02.cpp file using our standard set of options, creating an output executable called Lab02.
   6. Runs Lab02 using the /usr/bin/time command to determine the user, system and elapsed times. You should use an appropriate command line option so that the output of /usr/bin/time is just these times given in seconds. See the Background section above and the man page for time. You should run the program
      1. Eight times with these command line options: switch 2000000000
      2. Eight times with these command line options: if 2000000000
      3. Note that you may want to precede the execution with appropriate echo commands that remind you what the executable and options were for the run.
5. Execute your shell script using the nohup command and redirecting the output to a file called Lab02.timing.
6. Using the timing results in your Lab02.timing file:
   1. Fill in the Apparatus table below using your results from 4b-c
   2. Fill in the Observations tables in your Google drive spreadsheet
   3. Complete the Analysis tables in your Google drive spreadsheet
7. Using the results from your Analysis, answer the Discussion question below

.

# Apparatus (experimental platform)

The apparatus used for the experiments documented in this report were conducted on the following platform (fill in the two lines of Details using information determined in your shell script):

|  |  |
| --- | --- |
| Component | Details |
| CPU Model | Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz |
| Main Memory (RAM) size | 8056076 kB |

# Discussion

Based on your data, do you accept or reject the hypothesis that switch statements are faster than if-else statements? Correctly support your conclusion with data from the experiment and analysis, including an appropriate measure of the statistical significance. You should type your discussion into the table below.

|  |
| --- |
| **Record your inferences and conclusions below** |
| After running Lab02.cpp with switch statements and then with if statements, we can see an average time of 13.06 user seconds and 13.06 elapsed seconds for switch, and 12.771 user seconds and 12.776 seconds for if. Here we see that the switch statement is on average 1.022 times faster than the if statement, which goes for both user time and elapsed time. There is a difference of 2.26% between user times and a difference of 2.22% between elapsed times. After doing a Student’s T-test, we get a t statistic of 227.2. Based on what we see here, we can say that in this scenario, using an if-else statement is the faster option as opposed to a switch statement. |

# Submit to Canvas

When you complete the lab, download this document from Google Drive as a Microsoft Word (.docx) file with the naming convention Lab02 - MUid.docx (example: Lab02 - ferrenam.docx).  You should save the corresponding Google spreadsheet file as a Microsoft Excel (.xlsx) file with the naming convention Lab02 – Results MUid.xlsx (example: Lab02 - Results - ferrenam.xlsx)

Then, submit the following files to Canvas:

1. The Microsoft Word file you downloaded from Google Drive.
2. The Microsoft Excel file you downloaded from Google Drive.
3. The Lab02.bash shell script you created for this lab.
4. Your Lab02.cpp program with the style issues corrected.
5. Your Lab02.timing file.