

# PERFORMANCE, DATA STRUCTURES AND ALGORITHMS

Exercise 05

Exercise 05

Compare C to Java with Doubles

# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PURPOSE

The purpose of this exercise is to give you practice instrumenting a C program for performance measurement.

For this exercise, you will use both the C and Java implementations of the program that computes the square root of a number using doubles.

Both C and Java provide ways to measure performance in the form of system calls. The method to use in Java is System.nanoTime(). In C, you will use the clock() function. Your program will need to use these twice, once before the code you want to time, and once after. The difference between the two is the time it took to execute that portion of the program.

For information on the C function, refer to the C man page for the clock function by typing “man clock”. In addition to providing information about what the function does, this man page tells you that you need to include the header file for time.h (where clock() and relevant types and constants are defined) in your program by inserting the line “#include <time.h>” at the top of your program.

Note that you can also type “man 3 clock”. The 3 in the second command tells the man command to look in section 3 of the man pages, which contain C language functions, rather than shell commands and programs. It is sometimes necessary to tell the man program which section to refer to when there is a shell command and a C function that share the same name.

If you are using a virtual machine, be sure to obtain performance numbers from both programs running within the virtual machine, as the fact that you are using a virtual machine will impact performance.

ACTIVITIES

Perform each of the following activities. If you have questions, issues, or doubts, please ask for help and do not just guess.

1. Add code to measure the execution time *in milliseconds* of the part of the Java program that computes the square root of a number. Do not include program input and output in the code to be timed.
2. Add corresponding code to C program.
3. Run both programs and compare the execution time.
4. How long did it take to run each program (enter your answer in the space below)?
5. What conclusions can you draw about the performance of computations in Java versus in C based on your results (enter your answer in the space below)?
6. When you are ready to submit your work, first remove all intermediate files from your src directory. This includes exercise05.o and the exercise05 executable. You can use the “make” command to do this by typing “make clean” at the command prompt.
7. Save and archive your work, including this document and your finished C program (you do not need to submit the Java program), and upload it to the LMS.