

CS1XC3 - Computer Science Practice and Experience: Development Basics

Assignment #9 – Profiling and Doxygen

Weight

7% of total course grade

Due date

Tuesday April 6th at 11:59pm

Primary Learning Objectives

- Use gcc and gprof to profile a C program's performance.
- Use Doxygen to document a C program.

Requirements

You are given two starter code files **dotprod.zip** and **course.zip** in the Assignment #9 folder in the Avenue to Learn course shell.

Part A - Profiling Code

When you unzip **dotprod.zip** you will find a C program that conducts a dot product operation 1 billion times using a standard dot product algorithm. Improving the performance of linear algebra operations such as dot product is actually very important in computer graphics (e.g. video games) as operations like these are executed many times per second to generate individual frames of animation.

Use gcc and gprof on the pascal server to profile the performance of the current program. Save the results provided by gprof to a file called `old_prof.txt`.

Improve the performance of the program by creating and using a new dot product function in place of the old dot product function. Right now the dot product function is implemented to carry out the general dot product operation for vectors of a given length, and it uses a loop to manage the number of calculations. This adds overhead to the function's execution, and that

overhead would be unnecessary if we only have dot products of vectors of length 3 (which would not be unusual in a 3D game with coordinates x,y,z for example).

Create a function which assumes the vectors are both length 3. And then use the technique of loop unrolling (see: https://en.wikipedia.org/wiki/Loop_unrolling#Early_complexity) to remove the loop entirely. So your new function should consist of a single return statement that carries out the dot product computation for vectors of length 3.

Use gcc and gprof on the pascal server to profile the performance of the new program. The intention is that the new program is the same as the old program except using a new dot product function you have created instead. That said, if you do something like call both the old and new functions in the loop conducting the dot product operation, or if you copy and paste the loop testing the dot product function and have the 2nd loop test your version of the dot product function, these would both be acceptable ways to test the function too. Save the results provided by gprof to a file called new_prof.txt.

Save the new C program as dotprod.c. Zip the dotprod.c, old_prof.txt and new_prof.txt files together into a file called dotprod.zip.

Part B – Using Doxygen

When you unzip **course.zip** you will find a C program with libraries for creating courses with students. The main.c file is used to demonstrate these libraries.

Document this code using Doxygen comments. Be sure to document all files, functions (parameters, return values), and typedef struct definitions.

Add additional regular comments to the function bodies where appropriate to make the code easier to read. As discussed in-class, not every line should be commented, but brief descriptive comments documenting what a section of code is doing (e.g. a loop) or interesting features (e.g. why are both calloc and realloc being used in a function) is important.

Generate the Doxygen html documentation. Zip both your modified files and the Doxygen html documentation in a file called course.zip.

Submission

As noted above, save your solution files in zip files called dotprod.zip and course.zip and submit them to the Assignment #9 dropbox on Avenue to Learn.

Marking scheme

Component	Description	Marks
Profiling	Before and after profiling results are provided with expected value.	20
Optimization	Optimized version of dot product function.	20
Doxygen	Doxygen is used to comment the course program in the expected ways.	40
Comments	Regular comments are added to the function bodies of the program as appropriate.	20
Total:		100