

COMP 621 Program Analysis and Transformations Assignment #0

Getting Ready

Due: Monday, January 13, 2014 – beginning of class

Overview:

The purpose of this assignment is to get you engaged in the course and ready to do assignment #1. Assignment #0 is not worth any marks, but I do want a report from you handed in, and I would like you take this assignment seriously. For each question, just follow the directions about what to do and what to report. If you had trouble with the question, please describe what the difficulty was.

Question 0: *Join the course google group*

Visit the course web site at <http://www.sable.mcgill.ca/~hendren/621/index.html> and join the google group. Do this right away. Bookmark this web site as information will be added over the term.

Question 1: *Background - Compilers*

If you have already taken an introductory compiler course, then the answer for this question is just the name, and a short description of the course.

If you have not taken an introductory compiler course, then you can fill in the background you need for this course by doing some reading from any standard compiler textbook. To answer this question say what reading you did to prepare.

Question 2: *Background - Object-Oriented Programming*

In this course you will be expected to be able to understand large frameworks written in Java, and if you plan to do a project with the VM implementation, you should be comfortable with C++. To answer this question, please summarize your experience with Java and C++.

If you don't have significant experience, especially with Java, then please do some background reading on Java, and write some practice programs in Java. If this is your situation, please summarize how you prepared for Java.

Question 3: *Background in MATLAB*

MATLAB is a very popular language that is challenging for compiler optimizations. MATLAB was made to be easy for Scientists/Engineers to use, but it is actually very hard to analyze and optimize. In this course we will use the McLAB analysis framework for MATLAB to practice with designing new compiler analyses. However, before you can understand how to build the analyses you need to understand the foundations of the MATLAB language.

Both Mathworks' MATLAB and an open source version, Octave, are installed on the SOCS machines in Trottier. You may also find them on the research machines in your graduate labs (if they are not there, please ask your research groups to install them, McGill has an institutional license to install on all university-owned computers).

If you are not familiar with MATLAB, you can refer to the following documents:

- <http://www.mathworks.com/help/matlab/>, for the documentation on Matlab basics.
- http://ta.twi.tudelft.nl/nw/users/domenico/intro_fem/afternoon_lab_sessions/Matlab.manual.pdf
- <http://code.google.com/p/yagtom/>
- <http://www.gnu.org/software/octave/> for documentation for Octave.

To experiment with MATLAB and to start to understand performance issues with MATLAB, you should write a short numerical program, one version using MATLAB vector notation and one version using explicit loops. For the explicit loops version, be sure to "pre-allocate" any arrays you are defining in the body of the loop using the built-in `zeros` function. This is to prevent MATLAB from growing the array on each loop iteration.

You can use code you find online or in a book, but make sure you say where you got the code from, and make sure you understand the code thoroughly. You can also write your own code.

Check that your vector version and your loop version compute the same results.

Time the two versions for various problem sizes. Hint: you can use the commands `tic` and `toc`.

To answer this question, provide the two versions of your program, and briefly describe the results of your timing, and your interpretation of those results. You might have fun searching the web for various advice about how to use vectors to speed up MATLAB, and seeing if that advice is true or not. One relevant research paper can be found at:

<http://dl.acm.org/citation.cfm?id=331972>.¹

Hint for this question: MATLAB has a "JIT-accelerator" now, whereas before it was only an interpreter.

¹Note that any paper in the ACM digital library can be freely download from a McGill IP address. If you are working from home, use a McGill VPN.

If you feel even more ambitious, try running MATLAB in interpreter-only mode and repeat the experiment. You can turn the JIT on and off using the MATLAB commands `feature accel on` and `feature accel off`. It is on by default. What did you find?

To be even more ambitious, try it also on Octave, which is an open source interpreter. What did you find?

Question 4: *Your research interests*

Briefly describe your research interests and how COMP 621 fits with those interests.