

# 15-410 Self-assessment Exercise

## About the self-assessment

The prerequisite for 15-410, Operating Systems Design & Implementation, is 15-213, Introduction to Computer Systems. The latter is a sort of “Computer Architecture for Programmers” class, covering topics such as disks, caches, virtual memory, threads, and assembly language. It forms a solid basis for courses such as OS and Computer Architecture.

This self-assessment exercise, based on questions from a recent 15-213 final exam, is designed to provide prospective students with an understanding of the core 213 skills which will be used in 410. Projects 0 and 1 will be written mostly in C, with small scraps of x86 assembly language. Project 2, a thread library, will require a firmer understanding of assembly language and the relationship between a sequence of C code and the corresponding assembly language. In addition, Project 2 will naturally require (and develop) understanding of concurrency. These skills will then be used in Project 3 to write a small operating system. Again, this will be mainly in C, but critical parts will be in assembly language.

Thus, the questions in this exercise focus on decoding simple assembly language sequences, C memory layout and pointer arithmetic, and concurrency. This format is artificial in the sense that we will not ask you to demonstrate mastery of this material in an exam setting on the first day of class. *However, by approximately three weeks into the semester we will be relying on all of these skills to the extent that completing these three questions should take well under an hour.* For example, both the thread library and the operating system kernel are responsible for *building* a standard C runtime environment from underlying primitives. In order to debug sequences of C code which are correct but *execute* incorrectly you will need to translate between C and assembly language.

## “Taking the test”

If you had 15-213, this is a good opportunity to dust off your textbook and review parts of it, primarily Chapter 3. You might want to also briefly look over your

**Problem 1. (9 points):**

In this problem you will be given the task of reconstructing C code based on the declarations of C structures and unions provided below and the assembly code on the next page generated from the C code by a Linux/IA32 C compiler.

For each IA32 assembly code sequence below on the left, fill in the missing portion of the corresponding C source line on the right.

A