Homework 2

25% of homework grade, 2.5% of overall grade
out: 10/10 in class; in: 10/26 at the beginning of class

Decompiling Intel Assembly Language

In this homework, you will examine assembler output from gcc in order to determine what the original C code was.

Log into a TLAB machine and copy /home1/pdinda/HANDOUT/hw2.tar to a working directory. Untar the file (tar xvf hw2.tar). You will find the following files:

1. code-unopt.s (produced by gcc –Wall –S code.c –o code-unopt.s
2. code-unopt.o (produced by gcc –Wall –c code.c –o code-unopt.o
3. code-opt.s (produced by gcc –Wall –O –S code.c –o code-opt.s
4. code-opt.o (produced by gcc –Wall –O –c code.c –o code-opt.o
5. code.h
6. test.c
7. code-handin.c
8. Makefile
9. hw2.pdf (this document)

Your goal is to figure out what C code is in code.c and to replicate it in code-handin.c. The function definitions in code-handin.c are currently empty. Your will write them. It will probably easiest to do so by studying the contents of code-unopt.s and code.h and playing with the compiled code using test.c. The purpose of giving you code-opt.s and code-opt.o is give you an idea of what a compiler will do differently when optimizing. These files are not needed to complete the homework.

When you run make, you will generate code-handin.s, code-handin.o, test-with-handin, and test-with-handout. Code-handin.s and code-handin.o are the assembly and object code for code-handin.c – ie, the code that you've written. Test-with-handin is an executable of test.c that's linked with your code-handin.o. Test-with-handout is an executable of test.c that's linked with my code.o. You might also find it useful to compare your code-handin.s with my code-unopt.s.

Print your code-handin.c and hand it in.