Lab 8
Mon., 26 October 2015
Due Mon., 2 November, 8 a.m.

Purpose: NASM—Netwide Assembler; preparation for final project.

Details: By now you have a rough idea of how MIPS assembly language works, but there are many other machine architectures and instruction sets in the world. Today you will get to write some very simple assembly language programs using NASM—the Netwide Assembler. It is intended to be a portable assembly language for a wide range of Intel platforms, and it is installed on our lab machines in Alden. With NASM you can write assembly language routines that are callable from C (and other) programs.

[Incidentally, a recent Allegheny College graduate, Jim Kukunas, is helping out with the NASM project—see http://www.nasm.us]

We are going to be writing 32-bit NASM programs today (for similarity comparisons with MIPS). The lab machines are actually 64-bit machines, so we’ll need to add some information when we assemble and load our programs to indicate that we are using 32-bit mode.

See the sheet “Quick NASM” (on the course web site on the “Labs” page and also in the share repository).

1. Please fill in the CMPSC 210 survey that was emailed to you earlier this afternoon.

2. If you are stuck on lab 7, see me during lab today or make an appointment to see me tomorrow.

3. Download programs nasm1.asm, nasm2.asm, and nasm3.asm. Follow the directions in the comments to assemble, load, and execute them.

4. Create a NASM program that imitates the form of the three sample programs to perform computations identical to the following C program. Do not algebraically re-arrange the equation—perform all operations as indicated.

```c
#include <stdio.h>
int main() {
    int r = 119;
    int s = -32;
    int t = 7;
    int u = (r-(s-t)) & (s+t-r);
    printf("r=%d,s=%d,t=%d,u=%d\n",r,s,t,u);
}
```
5. [Optional—bonus question.] Download program nasm4.asm, read the comments, and run it. Then create a NASM program, imitating the style of nasm4.asm, that performs computations identical to the following C program:

```c
#include <stdio.h>
int main() {
    int i = 36;
    int j = 15;
    int k = i*i + j*j;
    printf("i=%d,j=%d,k=%d\n",i,j,k);
}
```

6. I am going to allow individuals in the class to choose which final project to do. There are four choices:

(a) Write a C program that takes, as input, one or more assignment statements involving integer arithmetic expressions (including multiplication and division), and prints out a correct MIPS program to carry out the computations and print the results. (Many details omitted from this description—among other things, you will have to process strings to figure out the expressions!)

(b) Write a collection of MIPS functions to carry out some set of tasks (to be specified later, but for example: functions to add two arrays, subtract two arrays, fill an array with some constant value, compare two arrays for equality, etc.). C code for these should be provided as well (for comparison).

(c) Learn enough NASM assembly language to create a set of functions (similar to the previous question), callable from a C program.

(d) Self-designed project (must involve programming in C and either MIPS or NASM; must be specified in sufficient detail that I can determine the level of difficulty; must be approved in advance by me).

In a text document, explain which of these projects has the most appeal to you and why (e.g., you want to learn more C programming, or you feel more comfortable with MIPS than with C, or you have an idea for a project in mind, or ... whatever your reason). If you have no preference, or there are just one or two that you are interested in, state this as well. The purpose of this exercise is to help me gauge interests so that I can tailor the project to the interests of as many individuals in the class as possible.

Submit the fully commented NASM programs and your preliminary project reflections by 8 a.m. on November 2.