Due: Wednesday, Sep 8, 2010. 11:59pm (no extensions).

What to submit: This exercise is intended to reinforce the content of lecture 2-4.

The command gcc is a compiler driver. When run with the flag ’-v’, it shows which programs it is actually executing. Invoke gcc with the flag ’-v’ on some test files and observe the output, then answer the following questions.

1. Which program is the actual C compiler? Give the full Unix filesystem path.

2. Does the C preprocessor appear to be a separate program or does it appear to be integrated into the C compiler?

3. How does gcc handle multiple files (e.g., gcc a.c b.c)?

4. By default, gcc will delete any temporary files it creates. Find out which command line switch can be used to save the temporary files.

5. In Lecture 2, the assembly code for a sample function ‘sum’ is shown:

```c
int sum(int x, int y)
{
    int t = x + y;
    return t;
}
```

This assembly code was obtained with an older version of gcc. Use gcc 4.1.2 (installed on the lab machine) and examine the assembly code it produces for this example. Use -O. You’ll find a minor difference. Explain it.

6. Consider this example from lecture:

```c
int arith
    (int x, int y, int z)
{
    int t1 = x+y;
    int t2 = z+t1;
    int t3 = x+4;
    int t4 = y * 48;
    int t5 = t3 + t4;
    int rval = t2 * t5;
    return rval;
}
```

We mentioned in class that converting the integer multiplication \( x \times 48 \) into a combination of addition and shifts: \((x + 2 \times x) \ll 4\) is also known as strength reduction. The decision whether to apply it rests on assumptions the compiler makes about the implementation of the underlying architecture it targets.

You can tell gcc to tune the code for specific implementations of a particular architecture using the -mtune flag. Tell gcc to tune for a) the Intel Core2 CPU (-mtune=core2)
and b) for the AMD Opteron implementation (-mtune=opteron).

For each of these cases, find out if gcc applies strength reduction to avoid the integer multiply or not. What do you conclude from your observation? Show your work.