ENGR 3410: HW #2
Assembly Languae Programming: Recursion

Due: 19 October 2010

The purpose of this homework is to develop your awareness of instruction set architecture support for procedure calls.

Honor Code Policy

You are to do this homework primarily alone. If you get stuck, you may consult anyone you like ***after*** *you have given the problem some serious effort. You must annotate, per problem, with any collaborators you had for that problem. Your aide should* ***not*** *simply provide you an answer to work backwards from: they are a resource, not an answer key. If you have any questions regarding resources, ask early, ask often.*

The Problem

Write a recursive MIPS subroutine that computes the greatest common divisor of two integers using Euclid's algorithm:

$$\begin{aligned} \gcd(a, 0) &= a \\ \gcd(a, b) &= \gcd(b, a \bmod b) \end{aligned}$$

Figure 1 shows how this can be written in C. You will, of course, be coding it in MIPS assembly language by hand (do *not* get `gcc` or some other program to translate a higher-level formulation into assembly language).

Your routine must:

```
/*
 * Euclid's method of computing the greatest common divisor of two integers,
 * a and b.
 */
int gcd(int a, int b)
{
        if (b == 0)
                return a;
        else
                return gcd(b, a % b);
}
```

Figure 1: Euclid's GCD algorithm in C

1. Be in a file called `gcd.asm` in your `hw2` directory, which you'll commit to the SVN repository.

2. Have its entry point labelled `gcd`.

3. Take its arguments in registers `$a0` and `$a1`.

4. Return its result in register `$v0`.

5. Call itself recursively (do *not* perform tail-call optimization and convert the code to a loop).

6. Because of the recursive calls, you're going to have to be concerned with saving and restoring registers. Save the arguments and return address.

You will want to test your code, of course. Please include some test code. Do NOT create a label called `main` or `_main`.