Assembler for Lab Computer

1 zassemble

The assembler for the lab computer is written in the C language. You should download the code, named zassemble.c from Canvas. Compile the assembler using your favorite C compiler to generate the *zassemble* executable. If you use gcc:

```
gcc zassemble.c -o zassemble
```

To run zassemble from a command prompt: **`./zassemble mycode.asm`**

You can append your call with the name of your output file:

```
./zassemble mycode.asm mycode.out
```

The default output file name, if one is not specified, is **`machine_code.coe`**

1.1 Assembly code conventions

zassemble is the assembler for the instruction set available in the instruction set document. In most cases *zassemble* follows the assembly coding conventions used by MIPS. For R-type instructions, the destination register is listed first. The order of registers in the generated machine code should run correctly on the datapath you have been building in this class.

Load word and store word instructions follow the following format:

```
lw $rt, offs($rs)
sw $rt, offs($rs)
```

where **`rs`** is the base address and **`rt`** is the destination register for a load instruction and the source register for a store instruction.

1.2 Current known issues

Use numbers for your registers, such as **`$1`**. Any other format may cause a segmentation fault.

Make sure your source file does not have non-printing characters in it. These may cause a segmentation fault.

Make sure using a space following the comma to separate arguments.