CPE 315
Spring 2015
Retz
Lab # 3

Objectives: to build a MIPS simulator front-end assembler/parser

Description

For this lab, you will write a two-pass assembler in C. This assembler will load in MIPS assembly files and output the corresponding machine code to the screen. The input to the assembler will be MIPS assembly files with comments, labels, and whitepace (spaces and tabs0> Assume the folloing regarding the input:

1. Comments – Comments start with '#'. Any content on a line after the comment character will be ignored.

2. Labels – Labels will contain alphameric characters and will be case-sensitive. A colon character will appear at the end of a label. There may be whitespace before a label and the label may be followed by an instruction. You will implement a symbol table for storing the label names and their associated values.

3. Blank lines may contain any mixture of whitespace characters.

4. There may be some whitespace between operands.

5. Immediate values may be in decimal, or if prefixed with "0x" will be in hexadecimal. Immediate values may be negative, as indicated by a preceding minus sign.

6. Your assembler must support the following instructions:

and, or, add, addu, addiu, sll, srl, sra, sub, sltu, sltiu, beq, bne, lw, sw, j, jr, and jal.

7. Your assembler should match mneumonic names against instruction opcodes in a case-insensitive fashion, and should identify invalid opcodes. Your assembler will be used in a future lab to generate object code for the assembled instructions.

8. This assembler is a two-pass assembler because it defines symbols during the first pass and generates object code in the second pass.

9. For this lab, your assembler will output machine code to the screen. In future labs, the machine code will be written to an object file that can be processed by a simulator.