Project 1 Computer Science 3050

1. About

This program is executing lines of MIPS code and return their respective machine code in 3.2 assembler while maintaining the allocated memory and simulate the instructions sets in 3.3 simulator.

1. Big Picture

This file is divided into 2 main part; assembler and simulator. In assembler, the programme idea is to convert lines of MIPS code into machine code. Firstly, the code will remove all unnecessary information and comments, then split the MIPS code into tokens (array). Next step is to check the value for every token by using dictionary (map) and store them with their corresponding address. Then split the instruction set into their type (R, I, J) and inserted the formula for each type. Then return their machine code for each MIPS. As of the Simulator, I started with inputting the static data (.data) to allocated memory. In put the machine code from assembler to the text segment. After that the machine code should robust to their instructions from each instruction functions.

1. Functions, Maps, and Class

* class Labelling

This class function is to store data\_type, content, name, and their address

* map<string, string> R

Dictionary (map) to keep all value of R type instruction set operators

* map<string, string> I

Dictionary (map) to keep all value of I type instruction set operators

* map<string, string> J

Dictionary (map) to keep all value of J type instruction set operators

* map<string, int> registers

Dictionary (map) to keep all value of all 31 registers

* int32\_t findLabels

Function to find labels and extract them

* string convertRegister

Convert register into 5 bits binary

* string R\_type

Function that include R-Instruction type formula to turn from OP, Rs, Rd, Rt, Shamt, Funct to the machine code

* string I\_type

Function that include I -Instruction type formula to turn from OP, Rs, Rt, Immediate to the machine code

* string J\_type

Function that include J-Instruction type formula to turn from OP, Immediate to the machine code

* string removeComments

Remove all the comments starting with #

* string removeTabOrSpace

Remove all tabs and space in front or end of each lines

* string final\_data

Return the finalized .text

* void parseLine

Split the finalized text into tokens

* string addressing

Save the address of each token from parseLine

* void machineCode

Generate machine code from according to their instructions

* int data\_input

Function to input the static data into allocated memory

* int text\_input

Function to input machine code to memory

* void view

Function to check if the token is separated and runs smoothly

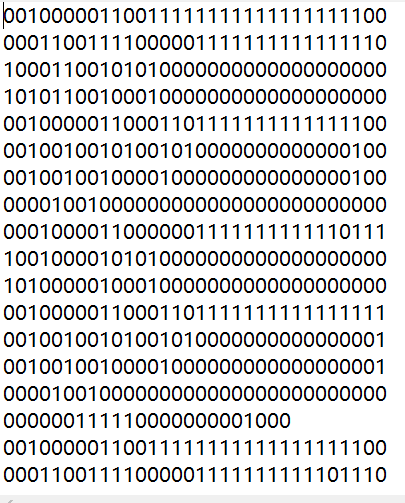
* void readfile

Read input file and return output file

* int main

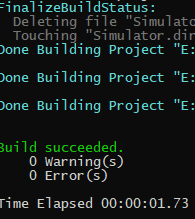
Call the whole programme

Output



Generate Machine Code for Assembler Segment

Problem: My code have some issue when I try to run on cmake/ linux however it can run smoothly on vsc.



Build Succeeded