

COL216 Assignment-3

Nishant Kumar- 2019CS50586

Mihir Meena- 2019CS10370

Aim: Develop an interpreter for a subset of MIPS assembly language instructions.

Input and Output:

A text file which contains the MIPS assembly language program is taken as the input. A C++ program reads the MIPS assembly language program and stores it in an internal data structure. Output is displayed after the execution of each instruction depicting the contents of all 32 registers. Finally after the completion of all the instructions the no. of clock cycles and no. of times each instruction was executed is printed on the terminal.

Design:

Two internal data structures are maintained for representing memory and registers files. Map is used for memory and array is used for registers files. Memory consists of two parts, $0-(2^{19}-1)$ bytes for instructions and $2^{19} - 2^{20}$ for data. If lw/sw tries to access instructions segment an error is raised. Also if no. of instructions goes out of instructions segment an error is raised. 32 registers are : \$r0 to \$r30 int registers and \$r31 is the register containing zero and its contents cannot be changed doing so will result in error. Also the input format must be delimited by just commas and not comma followed by space. If an input is in wrong format then corresponding error is raised and execution stops.

Test Cases:

We tested our code against the following types of test cases:

1. Proper input having mips instructions (Some basic assembly language programs).
2. Inputs having instruction where lw/sw try to access memory locations from instructions segments.
3. Inputs having instructions which try to change contents of \$r31 (zero register).

Some test cases against which we have tested the code.

Test case 1

Input:

```
addi $r2,$r1,6
j 4
sub $r2,$r2,$r2
add $r2,$r1,$r0
addi $r2,$r0,2
lw $r0,524288
lw $r1,524292
add $r2,$r1,$r0
sw $r2,524294
```

Output: Program executes normally

Test case 2

```
addi $r2,$r1,6
lw $r31,524292
add $r2,$r1,$r0
```

Output: Error: Cannot change zero register's value

Test case 3

Input: j 5

Output: Error: Memory location out of instructions memory segment.

Test case 4

```
Input: addi $r2,$r31,6
      addi $r3,$r31,6
      beq $r2,$r3,2
```

Output: Program executes normally

Test case 5

```
Input: addi $r2,$r31,5
      addi $r1,$r31,1
      bne $r2,$r1,2
```

Output: Program executes normally

Test case 6

Input: addi \$r0,\$r31,3
 addi \$r1,\$r31,5
 mul \$r2,\$r0,\$r1
 addi \$r3,\$r31,1
 beq \$r3,\$r2,4
 sw \$r3,524292
 lw \$r1,524292

Output: Program executes normally

Test case 7

Input: addi \$r0,\$r31,10
 sw \$r31,524288
 lw \$r0,524288
 sw \$r31,524284

Output: Error: Cannot access instructions memory segment.

Test case 8

Input: addi \$r0,\$r31,5
 addi \$r1,\$r31,6
 slt \$r2,\$r0,\$r1
 addi \$r0,\$r0,1
 slt \$r2,\$r0,\$r1

Output: Program executes normally