Akash Tadwai
Department of Engineering Science
Indian Institute of Technology Hyderabad
Email: es18btech11019@iith.ac.in

Toy Cool Programs

Akash Tadwai

May 19, 2020

Incorrect Cool Programs

Program - I:

Cause : Type Identifier should begin with capital letter
Error: "incorrect1.cl", line 3: syntax error at or near OBJECTID = string
Compilation halted due to lex and parse errors

Program - II:

Cause : Non-escaped characters shouldnot be present
Error: "incorrect2.cl", line 3: syntax error at or near OBJECTID = string
Compilation halted due to lex and parse errors

Program - III:

Cause : Right Star is missing in comment
Error: "incorrect3.cl", line 7: syntax error at or near ERROR = EOF in comment
Compilation halted due to lex and parse errors

Program - IV:

Cause : case sensitive keyword
Error: "incorrect4.cl", line 2: syntax error at or near TYPEID = False
Compilation halted due to lex and parse errors

Program - V:

Cause : Zero Width Space is not recognized by cool
Error:"incorrect5.cl", line 5: syntax error at or near ERROR = \342
Compilation halted due to lex and parse errors

Akash Tadwai
Department of Engineering Science
Indian Institute of Technology Hyderabad
Email: es18btech11019@iith.ac.in

Correct Cool Programs

The following are some common MIPS instructions which are used in MIPS assembly language.

- ***addiu*** → Add immediate unsigned (No overflow)
- ***sw*** → Store Word
- ***move*** → Move value stored at address to registers
- ***bne*** → Branch on not equal
- ***la*** → Load address
- ***li*** → Load immediate
- ***jal*** → Jump and link
- ***$so - $s7*** → Saved values representing final computed results
- ***$to - $t9*** → Temporary variables
- ***$ao - $a3*** → Arguments for a subroutine
- ***$ra*** → Return address
- **class_nameTab** section contains information about the classes like str_const parts
- **str_const(I)** sections contains all the string literals in our code section.
- **Function calls** corresponds to using branch instructions in the assembly

The MIPS generated codes for the correct programs are as follows:

Program - I : Exponent

```
Main.exponent:
  addiu $sp $sp -12
  sw  $fp 12($sp)
  sw  $s0 8($sp)
  sw  $ra 4($sp)
  addiu $fp $sp 4
  move  $s0 $a0
  la  $a0 str_const0
  sw  $a0 0($sp)
  addiu $sp $sp -4
  move  $a0 $s0
  bne $a0 $zero label6
  la  $a0 str_const4
  li  $t1 1
  jal _dispatch_abort
```

Listing 1: *exponent.s*

Akash Tadwai
Department of Engineering Science
Indian Institute of Technology Hyderabad
Email: es18btech11019@iith.ac.in

Program - II: LCM

```
Main.LCM:
  addiu $sp $sp -16
  sw  $fp 16($sp)
  sw  $s0 12($sp)
  sw  $ra 8($sp)
  addiu $fp $sp 4
  move  $s0 $a0
  la  $a0 str_const0
  sw  $a0 0($sp)
  addiu $sp $sp -4
  move  $a0 $s0
  bne $a0 $zero label10
  la  $a0 str_const4
  li  $t1 1
  jal _dispatch_abort
```

Listing 2: *LCM.s*

Program - III: Parity Check

```
Main.parity:
  addiu $sp $sp -12
  sw  $fp 12($sp)
  sw  $s0 8($sp)
  sw  $ra 4($sp)
  addiu $fp $sp 4
  move  $s0 $a0
  la  $a0 str_const0
  sw  $a0 0($sp)
  addiu $sp $sp -4
  move  $a0 $s0
  bne $a0 $zero label11
  la  $a0 str_const4
  li  $t1 1
  jal _dispatch_abort
```

Listing 3: *parity.s*

Akash Tadwai
Department of Engineering Science
Indian Institute of Technology Hyderabad
Email: es18btech11019@iith.ac.in

Program - IV: Pascal Traingle

```
Main.Pascal_traingle:
  addiu $sp $sp -20
  sw  $fp 20($sp)
  sw  $s0 16($sp)
  sw  $ra 12($sp)
  addiu $fp $sp 4
  move  $s0 $a0
  la  $a0 str_const0
  sw  $a0 0($sp)
  addiu $sp $sp -4
  move  $a0 $s0
  bne $a0 $zero label0
  la  $a0 str_const4
  li  $t1 1
  jal _dispatch_abort
```

Listing 4: *Pascal_triangle.s*

Program - V: Subsequence Check

```
Main.subsequence:
Main.main:
  addiu $sp $sp -12
  sw  $fp 12($sp)
  sw  $s0 8($sp)
  sw  $ra 4($sp)
  addiu $fp $sp 4
  move  $s0 $a0
  la  $a0 str_const0
  sw  $a0 0($sp)
  addiu $sp $sp -4
  move  $a0 $s0
  bne $a0 $zero label13
  la  $a0 str_const5
  li  $t1 1
  jal _dispatch_abort
```

Listing 5: *subsequence.s*

LaTeX generated document