Assignment 0
Toy Cool Programs

Vinta Reethu
ES18BTECH11028

May 9, 2020

Correct programs -

MIPS instruction:

- addiu – Add immediate unsigned (no overflow)
- sw – Store word
- move – Move values stored at addresses to registers
- lw – Load word
- la – Load address
- move – Copies a value from one register to another
- beq – Branches on equal
- jal – Jump and link
- add – Arithmetic instruction to add
- mul – Multiply two registers
- div – Signed division
- jr – Jumps to register
- bne – Branch on not equal
- li – Loads on immediate value into register
- General purpose registers (GPRs) are indicated with a dollar sign ($).
- $so - $s7 – Saved value represent final computed results

- $t0 - $t9 – Temporary variable
- $a0 - $a9 – Arguments for subroutine
- class_nameTab – Class Names Table includes pointers to all the names of the classes.
- str_const(x) – Contains all string literals of our code.
- class_objTab – Objects Table contains pointers to the object prototype and init method of each class.
- x_dispTab – Dispatch Tables contain the pointers to the different functions defined for each class : those inherited from its parent and those defined in the class itself.
- x_protObj – Object prototypes are used as an empty or dummy version of an object of each class.
- Main_init – Code that initializes an object of class Main passed in $a0
- Main.main – The main method for class Main $a0 contains the initial Main object

Program 1 - Fibonacci Numbers :

```
    addiu   $sp $sp -20
    sw      $fp 20($sp)
    sw      $s0 16($sp)
    sw      $ra 12($sp)
    addiu   $fp $sp 4
    move    $s0 $a0
    lw      $s1 20($fp)
    la      $t2 int_const1
    move    $t1 $s1
    la      $a0 bool_const1
    beq     $t1 $t2 label2
    la      $a1 bool_const0
    jal     equality_test
```

Listing 1: Main.fibo – Find fibonacci number.

Program 2 - GCD of two numbers :

```
    addiu   $sp $sp -16
    sw      $fp 16($sp)
    sw      $s0 12($sp)
    sw      $ra 8($sp)
    addiu   $fp $sp 4
    move    $s0 $a0
    lw      $s1 20($fp)
    la      $t2 int_const0
    move    $t1 $s1
    la      $a0 bool_const1
    beq     $t1 $t2 label2
    la      $a1 bool_const0
    jal     equality_test
```

Listing 2: Main.gcd – GCD of two numbers

Program 3 - Sum of sqaures of first n numbers :

```
    addiu   $sp $sp -20
    sw      $fp 20($sp)
    sw      $s0 16($sp)
    sw      $ra 12($sp)
    addiu   $fp $sp 4
    move    $s0 $a0
    lw      $s1 20($fp)
    lw      $s2 20($fp)
    la      $a0 int_const0
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s2)
    add     $t1 $t1 $t2
    sw      $t1 12($a0)
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s1)
    mul     $t1 $t1 $t2
    sw      $t1 12($a0)
    move    $s1 $a0
    la      $s2 int_const1
    lw      $a0 20($fp)
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s2)
    mul     $t1 $t1 $t2
    sw      $t1 12($a0)
    move    $s2 $a0
    la      $a0 int_const0
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s2)
    add     $t1 $t1 $t2
    sw      $t1 12($a0)
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s1)
    mul     $t1 $t1 $t2
    sw      $t1 12($a0)
    move    $s1 $a0
    la      $a0 int_const2
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s1)
    div     $t1 $t1 $t2
    sw      $t1 12($a0)
    sw      $a0 20($s0)
    lw      $fp 20($sp)
    lw      $s0 16($sp)
    lw      $ra 12($sp)
    addiu   $sp $sp 24
    jr      $ra
```

Listing 3: Main.sum_of_squares - Sum of sqaures of first n numbers

Program 4 - Swapping of two numbers :

```
    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 label0
    la      $a0 str_const7
    li      $t1 1
    jal     _dispatch_abort
```

Listing 4: Main.main – Swapping of two numbers

Program 5 - Even number or not :

```
    addiu   $sp $sp -16
    sw      $fp 16($sp)
    sw      $s0 12($sp)
    sw      $ra 8($sp)
    addiu   $fp $sp 4
    move    $s0 $a0
    lw      $a0 16($fp)
    sw      $a0 0($sp)
    addiu   $sp $sp -4
    la      $a0 int_const0
    sw      $a0 0($sp)
    addiu   $sp $sp -4
    move    $a0 $s0
    bne     $a0 $zero label3
    la      $a0 str_const5
    li      $t1 1
    jal     _dispatch_abort
```

Listing 5: Main.check - Even number or not

InCorrect programs -

Incorrect 1 :

Identifier begins with capital letter, here it begins with ’Number’ instead of ’number’.
Identifier should always begin with small letter.
Error Shown -
”incorrect1.cl”, line 3: syntax error at or near TYPEID = Number
”incorrect1.cl”, line 8: syntax error at or near TYPEID = Number

Incorrect 2 :

Undetermined string constant.Can be avoided using backslash.
A non-escaped newline character may not appear in a string.
Error Shown -
”incorrect2.cl”, line 4: syntax error at or near ERROR = Unterminated string constant

Incorrect 3 :

Comments cannot cross file boundaries.
Error Shown -
”incorrect3.cl”, line 10: syntax error at or near ERROR = EOF in comment

Incorrect 4 :

”true” and ”false” are case sensitive word in COOL.
Starting letters in true and false should be in small letters.
Error Shown -
”incorrect4.cl”, line 3: syntax error at or near TYPEID = True

Incorrect 5 :

Zero space is not recognised in COOL.
Error Shown -
”incorrect5.cl”, line 7: syntax error at or near ERROR = \342

Non trivial Programs -

Non trivial 1:

Finds whether given number is *Armstrong number* or not.
A number is said to be Armstrong number if the sum of cubes of individual digits of number is equal to the number.
Example : $n = 370$
Sum of cubes of digits = $3^3 + 7^3 + 0^3$ = 27 + 343 + 0 = 370

Non trivial 2:

Finds out the *square root* of a number and also *prime factors* of that number.
Example : $n = 15$
Square root = 3(Rounded of to floor value)
Prime factors = 3 , 5.

Thank You!