Compilers-1 - CS3510

Toy Cool Programs

Correct COOL Programs

Please see below some of the MIPS assembly language instructions

1) **li** : Load immediate value instruction
2) **sw** : store word
3) **move** : move the value stored in the address to registers
4) **addiu** : add the immediate unsigned without any overflow
5) **bne** : branch on equal
6) **str_const(l)** it contains all the string literals in out code sections
7) Function calls use branch instructions which are in assembly language.
8) **la** : stand for local address
9) **$sp** is stack pointer
10) **$fp** is frame pointer
11) Labels are set of instructions to be executed which are in assembly language when we were directed to go there

Program1:
Average of N numbers rounded off

```
Main.main:
   addiu   $sp $sp -32     #space on stack have been reserved and moving stack pointer
   sw   $fp 32($sp)          #save the frame pointer onto stack
   sw   $s0 28($sp)          #preserve $s0
   sw   $ra 24($sp)          #preserve the Returned address pointer
   addiu   $fp $sp 4
   move        $s0 $a0
   la    $a0 str_const0      #str_const0 is string represented by StringSymbol
   sw   $a0 0($sp)           #store word in register a0 to 0($sp) address
   addiu   $sp $sp -4
   move        $a0 $s0
   bne         $a0 $zero label0     #branch to label0 if a0 not equal to zero
   la    $a0 str_const5              #these statements get executed after label0 returns
   li    $t1 1
```

```
    jal   _dispatch_abort       #end program execution
```

Program 2:
Factorial of a number N

```
Main.fact:                              #for the factorial function assembly code
        addiu   $sp $sp -20             #reserving space on stack
        sw      $fp 20($sp)             #preserve the frame pointer onto stack
        sw      $s0 16($sp)             #preserve $s0
        sw      $ra 12($sp)             #preserve the Returned address pointer
        addiu   $fp $sp 4               #moving frame pointer by 4 bits
        move    $s0 $a0
        lw      $s1 20($fp)             #restore the value
        la      $t2 int_const0          #the integer constant in local address
        move    $t1 $s1                 #move the number read into $t1
        la      $a0 bool_const1         #checking with boolean constant
        beq     $t1 $t2 label2          #branch on equal to label2 if t1 = t2
        la      $a1 bool_const0
        jal     equality_test           #jump to label
```

Program 3
GCD of 2 numbers:

```
Main.gcd:
        addiu   $sp $sp -16             #this method was called in the main method
        sw      $fp 16($sp)         #preserve the frame pointer onto stack
        sw      $s0 12($sp)
        sw      $ra 8($sp)
        addiu   $fp $sp 4           #moving frame pointer by 4 bits
        move    $s0 $a0
Main.main:
        addiu   $sp $sp -12         #reserving space on stack
        sw      $fp 12($sp)         #preserve the frame pointer onto stack
        sw      $s0 8($sp)
        sw      $ra 4($sp)
        addiu   $fp $sp 4
```

```
    move  $s0 $a0
    la    $a0 str_const0        #string constant in the local address
    sw    $a0 0($sp)
    addiu  $sp $sp -4           #reserving space on stack
    move  $a0 $s0
    bne   $a0 $zero label7   #branch on not equal to label7 and execute instructions
    la    $a0 str_const4
    li    $t1 1
    jal   _dispatch_abort         #jump to label and exit execution and abort is given
```

Program 4

Pattern:

```
Main.pattern:
    addiu  $sp $sp -28          #reserving space on stack
    sw    $fp 28($sp)
    sw    $s0 24($sp)
    sw    $ra 20($sp)
    addiu  $fp $sp 4            #moving frame pointer by 4 bits
    move  $s0 $a0
    la    $s1 int_const0        #initialising two integers
    la    $s2 int_const0     # labels follow after this which execute the instructions

Main.main:
    addiu  $sp $sp -16          #reserving space on stack
    sw    $fp 16($sp)
    sw    $s0 12($sp)
    sw    $ra 8($sp)
    addiu  $fp $sp 4            #moving frame pointer by 4 bits
    move  $s0 $a0
    la    $a0 str_const2              #string constant in the local address
    sw    $a0 0($sp)
    addiu  $sp $sp -4
    move  $a0 $s0                     #this main method calls pattern method
    bne   $a0 $zero label8
    la    $a0 str_const4
    li    $t1 1
    jal   _dispatch_abort
```

Program 5
String Reverse:

```
Main.main:
        addiu   $sp $sp -28                 #reserving space on stack
        sw      $fp 28($sp)
        sw      $s0 24($sp)
        sw      $ra 20($sp)
        addiu   $fp $sp 4
        move    $s0 $a0
        la      $a0 str_const0              #string constant in the local address
        sw      $a0 0($sp)
        addiu   $sp $sp -4
        move    $a0 $s0                     #labels are present in the code which tell
        bne     $a0 $zero label0      #which operation to perform according to the code
        la      $a0 str_const2
        li      $t1 1
        jal     _dispatch_abort       #end program execution
```

The instructions have different labels to be executed in their program flow the labels and the instructions there will be evaluated accordingly.There will be different branches happening due to conditional statements depending on their boolean values we need to execute certain statements so labels represent these different branches which happens and certain statements in the code will be executed on the branch we go on due to the conditional statement boolean value.Please see that the explanations of some of the instructions have been mentioned in the points above.

Incorrect Programs
Program 1

The identifier for Int starts with a capital letter X, where the identifiers should start with small letters.

Error:

```
"incorrect1.cl", line 3: syntax error at or near TYPEID = X
"incorrect1.cl", line 5: syntax error at or near TYPEID = X
Compilation halted due to lex and parse errors
```

Program 2

Unterminated string constant

Error:

"incorrect2.cl", line 3: syntax error at or near ERROR = Unterminated string constant
Compilation halted due to lex and parse errors

Program 3

Error due to dash, improper comment.

Error:

"incorrect3.cl", line 8: syntax error at or near '-'
Compilation halted due to lex and parse errors

Program 4

Here **'fi'** is a key word in the language

Error:

"incorrect4.cl", line 3: syntax error at or near FI
"incorrect4.cl", line 6: syntax error at or near FI
Compilation halted due to lex and parse errors

Program 5

White space zero length is not allowed by the COOL language

Error:

"incorrect5.cl", line 4: syntax error at or near ERROR = \342
Compilation halted due to lex and parse errors