COE 301/ICS 233, Term 171

Computer Architecture & Assembly Language
Quiz# 2 Solution

Date: Tuesday, Oct. 31, 2017

Q1. Determine the output produced by the following program given that the program inputs are 7 and 4. Needed syscall services are given below.

Service	$v0	Arguments / Result
Print Integer	1	$a0 = integer value to print
Read Integer	5	$v0 = integer read
Exit Program	10	

```
            .text
            .globl main
main:
            li $v0, 5
            syscall
            move $t0, $v0
            li $v0, 5
            syscall
            move $a1, 5
            move $a0, 7
            jal Proc1
            move $a0, $v0
            li $v0, 1
            syscall
            li $v0, 10
            syscall
Proc1:
            bne $a0, $a1, Skip
            move $v0, $a0
            jr $ra
Skip:
            addi $sp, $sp, -8
            sw $a0, ($sp)
            sw $ra, 4($sp)
            addi $a0, $a0, -1
            jal Proc1
            lw $t0, ($sp)
```

```
        lw $ra, 4($sp)
        addi $sp, $sp, 8
        add $v0, $v0, $t0
        jr $ra
```

The program computes 7+6+5+4=22 and then displays it.

Q2. Assuming that functions F and G receive two integer arguments in $a0 and $a1 and return their results in $v0, implement the function F given below saving needed registers on the stack. Save changed registers according to the assumed programming convention:

```
        int F(int a, int b) {
             return  G(2a,b)+G(a,2b);
        }
```

```
F:      addiu  $sp, $sp, -16   # frame = 16 bytes
        sw     $ra, 0($sp)      # save $ra
        sw     $a0, 4($sp)      # save argument a
        sw     $a1, 8($sp)      # save argument b
        sll    $a0, $a0, 1      # $a0 = 2*$a0
        jal    G                # call G(2a,b)
        sw     $v0, 12($sp)     # save result of call G(2a,b)
        lw     $a0, 4($sp)      # $a0 = a
        lw     $a1, 8($sp)      # $a0 = b
        sll    $a1, $a1, 1      #$a1 = 2*$a1
        jal    G                # call G(a,2b)
        lw     $v1, 12($sp)     # restore result of call G(2a,b)
        addu   $v0, $v0, $v1    # $v0 = G(2a,b)+G(a,2b)
        lw     $ra, 0($sp)      # restore $ra
        addiu  $sp, $sp, 16     # free stack frame
        jr     $ra
```