姓名：張碩恩

學號：404410053

系級：資工三

請同學以PDF之檔案格式上傳你的答案

Exercise 2.4 (課本p.165)

For the MIPS assembly instructions below, what is the corresponding C statement? Assume that the variables f, g, h, i, and j are assigned to registers$s0, $s1, $s3, and $s4, respectively. Assume that the base address of the arrays A and B in registers $s6 and $s7, respectively.

sll $t0, $s0, 2 # $t0 = f \* 4

add $t0, $s6, $t0 # $t0 = &A[f]

sll $t1, $s1, 2 # $t1 = g \* 4

add $t1, $s7, $t1 # $t1 = &B[g]

lw $s0, 0($t0) # f = A[f]

addi $$t2, $t0, 4 # $t2 = &A[f+1] // f here is original f, not A[f]

lw $t0, 0($t2) # $t0 = A[f+1] // f here is original f, not A[f]

add $t0, $t0, $s0 # $t0 = A[f+1] + A[f] // f here is original f, not A[f]

sw $t0, 0($t1) # B[g] = A[f+1] + A[f] // f here is original f, not A[f]

Overall:

B[g] = A[f+1] + A[f]

f = A[f]

(We cannot swap the order.)

Exercise 2.23 (課本p.169)

Assume $t0 holds the value 0x00101000. What is the value of $t2 after the following instructions?

slt $t2, $0, $t0 # 0<0x00101000, so $t2 = 1 // $t2 = 1

bne $t2, $0, ELSE # 1 not equal to 0, so goto ELSE // $t2 = 1

j DONE

ELSE: addi $t2, $t2, 2 # $t2 = $t2+2, so $t2 = 3 // $t2 = 3

DONE:

$t2 = 3

Exercise 2.31 (課本p.171)

Implement the following C code in MIPS assembly. What is the total number of MIPS instructions needed to execute the function?

int fib( int n ) {

If ( n==0 )

return 0;

else if ( n==1 )

return 1;

else

return fib(n-1) + fib(n-2) ;

fib:

addi $sp, $sp, -12 # allocate stack frame of 12 bytes

sw $a0, 8($sp) # store arguement

sw $ra, 4($sp) # store return address

sw $s0, 0($sp) # store $s0

slti $t0, $a0, 2 # if $a0<2, $t0=1 // for fib(0) or fib(1)

beq $t0, $0, else # if $t0=0, goto else // for argument >= 2

add $v0, $a0, $0 # $v0=$a0 // fib(0)=0, fib(1)=1

j exit # finish fib

else:

addi $a0, $a0, -1 # fib(n-1)

jal fib # recursive call

add $s0, $v0, $0

addi $a0, $a0, -1 # fib(n-2)

jal fib # recursive call

add $v0, $v0, $s0

exit:

lw $a0, 8($sp) # restore arguement

lw $ra, 4($sp) # restore return address

lw $s0, 0($sp) # restore $s0

addi $sp, $sp, 12 # free stack

jr $ra # return

The total number of MIPS instructions needed to execute the function will depend on the argument. And because it is recursive call, it may be very large.