Patrick Austin

CS 219 Homework 3

28 February 2016

1. MARS snapshot of successful run attached on next page.

.text

addi $a0, $a0, 4 # example n = 4 - function should end with value a in register v0

# and sp at its original value, 0x7fffeffc

# and a0 equal to its original value, 4

sum:

addi $sp, $sp, -8 # adjust stack for 2 items (argument, return address)

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

sw $a0, 0($sp) # save argument

add $t0, $a0, $zero # set t0 to a0. t0 will contain the current value of n

add $t1, $t1, $a0 # add a0 to t1. t1 will contain the sum so far

bne $t0, $zero, L1 # test if n=0

add $v0, $t1, $zero # if so, save final sum to v0

addi $sp, $sp, 8 # and pop the 2 items from the stack

jr $ra # and return

L1:

addi $a0, $a0, -1 # otherwise, decrement n

jal sum # recursively call sum

lw $a0, 0($sp) # restore original sum

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

addi $sp, $sp, 8 # pop 2 items from the stack

jr $ra # and return



2. MARS snapshot of successful run attached on next page.

.data

IntArray: .word 0x5,0x9,0x4,0xFF,0xFE,0xFA,0x7,0x2,0x8,0xFD # initial values of integer array

size: .word 10 # number of integers in IntArray

result: .space 4 # declare 4 bytes to hold integer result

.text

la $t0, IntArray # t0 will hold current address of int array, starting at index 0

addi $t1, $zero, 0 # t1 will hold current index of array (initially 0)

addi $t2, $zero, 9 # t2 will hold max size of array (9)

lw $t3, 0($t0) # load int at index 0 of the array- t3 will hold current max int

loop:

beq $t1, $t2, end # if current index equals max index, end looping and go to save final result

addi $t0, $t0, 4 # otherwise, move to next location in the array

addi $t1, $t1, 1 # increment current index by 1

lw $t4, 0($t0) # load current index int to t4

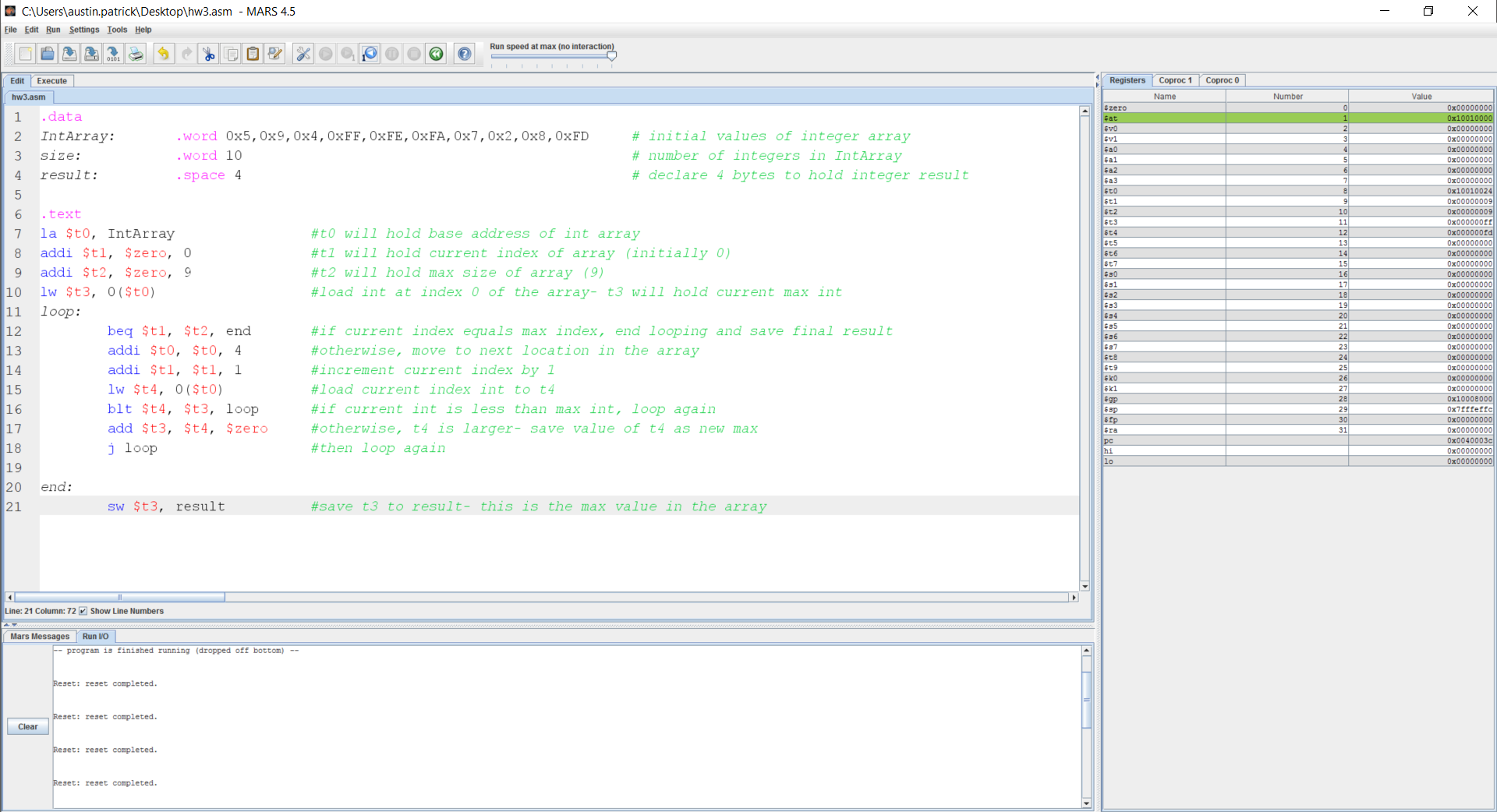
blt $t4, $t3, loop # if current int is less than max int, loop again

add $t3, $t4, $zero # otherwise, t4 is larger- save value of t4 in t3 as new max

j loop # then loop again

end:

sw $t3, result # save t3 to result- this is the max value in the array



3. MARS snapshot of successful run attached on next page.

.data

g: .word 14 # initial value of variable g

h: .word 15 # initial value of variable h

i: .word 1 # initial value of variable i

j: .word 21 # initial value of variable j

f: .space 4 # declare 4 bytes of storage to hold integer result

.text

lw $a0, g # load g from memory into a0

lw $a1, h # load h from memory into a1

lw $a2, i # load i from memory into a2

lw $a3, j # load j from memory into a3

jal leaf # jump to leaf procedure

j end # jump to end of program

leaf: # enter leaf procedure

addi $sp, $sp, -4 # make space to save s0 on stack

sw $s0, 0($sp) # save s0 on stack

add $t0, $a0, $a1 # save result of g + h in t0

add $t1, $a2, $a3 # save result of i + j in t1

sub $s0, $t0, $t1 # save result of (g + h) - (i + j) in s0

add $v0, $s0, $zero # save result to v0 from s0

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

addi $sp, $sp, 4 # restore value of sp

jr $ra # return

end:

sw $v0, f # save value from v0 in memory at f, operation complete

