McClane Howland

4/19/17

CS010

Lab 1

Introduction to MARS and MIPS

**Which registers have a non-zero value at this point?**

Register number 29, $sp, has a non-zero value at this point.

pc also has a non-zero value at this point.

**Explain what the value of the pc indicates:**

The value pc represents the program counter, and is a register that always holds the memory address of the next instruction to be executed.

**What does the 0x notation mean which precedes the 8-digit numbers you see displayed in these panels?**

The 0x notation means that the 8-digit numbers displayed in the value column of the

register panels are in hexadecimal.

**5. The Text Segment contains the code from the .text section of the program (the program instructions). Explain what you think each column in this panel is used for:**

**Bkpt:**

Bkpt stands for breakpoint. This is where you can insert a stop to test

individual parts of your program.

**Address:**

I think the address column shows the memory address of the instruction that has

been executed.

**Code:**

The code column holds the binary representations of the instructions.

**Basic:**

The basic column shows the code that is really being executed, instead of the

pseudo functions like 'move.'

**Source:**

The source column shows the instruction line in the text file that has been

executed.

**6. What is the starting address of the program?**

The starting address of the program is 0x00400000

**7. The Data Segment contains the code from the .data section of the program (the variables and constants defined in the program). What is the starting address of the Data Segment?**

The starting address of the Data Segment is 0x10010000

**8. Each row in the Data Segment lists the contents of 8 words in memory, each of which contains 32 bits, or 4 bytes, of data. Notice that the first 7 words in the Data Segment contain non-zero values. Why are these non-zero for this program?**

Because 7 values are loaded into the data segment in the base address of this program

during execution

**The value of $t3 is 0 at this point. After the breakpoint the value of $t3 is 5, the sum of $t1 and $t2.**

**Code For Exercise 2:**

#prompt for name

li $v0,4 #load service number in $v0

la $a0,prompt\_name #load address of string to be printed in $a0

syscall

#get the name

li $v0,8 #load serice number in $v0

la $a0,name

lw $a1,namelength

syscall

#prompt for age

li $v0,4 #load service number in $v0

la $a0,prompt\_age #load address of string to be printed in $a0

syscall

li $v0,5 #load in the age

syscall

move $t0,$v0 #store the age in $t0

#greet

li $v0,4 #load service number in $v0

la $a0,greeting\_string #load address of string to be printed in $a0

syscall

li $v0,4 #load service number in $v0

la $a0,name

syscall

#add four to the age and store it in $t2

li $t1,4 #load the constant 4 into $t1

add $t2,$t0,$t1 #add age and 4

#print the first part of the age string

li $v0,4 #load service number in $v0

la $a0,result\_stringpt1 #load address of string to be printed in $a0

syscall

#print the age

li $v0,1 #load service number in $v0

move $a0,$t2 #move address in $t2 to $a0 to be printed

syscall

#print the second part of the age string

li $v0,4 #load service number in $v0

la $a0,result\_stringpt2 #load address of string to be printed in $a0

syscall

# the null-terminated string must be defined in data segment

.data

prompt\_name: .asciiz "What is your name? "

name: .space 50

namelength: .word 50

prompt\_age: .asciiz "What is your age? "

age: .space 50

greeting\_string: .asciiz "Hello, "

result\_stringpt1: .asciiz "You will be "

result\_stringpt2: .asciiz " in four years"

##

**Code for Exercise 3:**

la $t3, note\_array # put address of list into $t3

li $a1,100 #duration

li $a2,0 #instrument

li $a3,127 #volume

li $v0,33 #set syscall to type 33, which is "MIDI out synchronous"

li $t0,20 #for loop limit,

li $t1,0 #for loop increment variable

loop:

beq $t1,$t0,end #if $t0 == $t1 end the loop

#loop body

la $t2,($t1) #copy the counter into the index

add $t2, $t2, $t2

add $t2, $t2, $t2 #index now a multiple of 4, bc added together twice

add $t4,$t2,$t3 #add index and array

lw $t5,0($t4) #retrieve the value from the array

move $a0,$t5 #move the retrieved value into the output register

syscall

#increment $t1

addi $t1,$t1,1 #increment $t1

j loop #jump back to the top

end:

#data segment

.data

note\_array: .word 60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80