**练习: 熟悉MARS**

## 编一个简短的MIPS程序

编写MIPS代码完成：在给定$s0 和 $s1的值的前提下，将下列值放到 $t? 寄存器中（其中？表示任意0-7之间的数）:

$t0 = $s0

$t1 = $s1

$t2 = $t0 + $t1

$t3 = $t1 + $t2

...

$t7 = $t5 + $t6

换言之, 对$t2 到 $t7的每个寄存器，都存储其前两个$t? 寄存器的值. 寄存器$s0 和 $s1 中包含初始值.

不要在代码中设置$s0 和 $s1 的值. 取而代之, 学会如何在MARS中手动设置它们的值.

**练习 2**

假定你想编写一个MIPS程序foo，该程序使用5个字的数组，数组元素初始化为整数1, ..., 5.

.data

foo: .word 1,2,3,4,5

你用程序来把数组foo中的每个数加2再写回数组foo

\_start:

addi $t0, $0, foo

lw $t1,0($t0)

addiu $t1, $t1,2

sw $t1, 0($t0)

.....

你也可以使用循环来完成。

回答下列关于MARS的问题.

1. .data, .word, .text 指示器（directives）的含义是什么(即, 在每段中放入什么内容)?
2. 在MARS中如何设置断点breakpoint?
3. 在程序运行到断点处停止时，如何继续执行? 如何单步调试代码?
4. 如何知道某个寄存器register的值是多少? 如何修改寄存器的值.

将你的代码存储到文件lab\_ex.s 中，然后给老师检查.

系统调用：

自己定义字符串并显示。

#---------------------------------

.data 0x0

startString: .asciiz "Converting pixels to grayscale:\n"

finishString: .asciiz "Finished."

newline: .asciiz "\n"

.text 0x3000

main:

ori $v0, $0, 4 #System call code 4 for printing a string

la $a0, startString #address of startString is in $a0

syscall #print the string

#------- INSERT YOUR CODE HERE -------

#

#(You may delete the comment here when you insert your code)

#

#------------ END CODE ---------------

exit:

ori $v0, $0, 4 #System call code 4 for printing a string

la $a0, finishString #address of finishString is in $a0; we computed this

#simply by counting the number of chars in startString,

#including the \n and the terminating \0

syscall #print the string

ori $v0, $0, 10 #System call code 10 for exit

syscall #exit the program

**# Read a string**

.data

m1: .asciiz "Read next line\n"

.text

.globl main

main:

li $v0 4 # syscall 4 (print\_str)

la $a0 m1

syscall

li $v0 8 # syscall 8 (read\_string)

syscall

li $v0 4 # syscall 4 (print\_str)

syscall

jr $ra