

CS120 Lab 06 — MIPS Assembly language coding

We'll use the MIPS simulator; code for ARM is similar. There are a few built-in functions that print letters and numbers — that happens with “sys call” — more details on how that works later, for right now, just go with it.

Use a text editor, and create files for these. Look closely at the machine code in the simulator. When you're running your code, it's ONLY THE MACHINE CODE that matters!

1) Hello World. Enter the following, and run it on the simulator

```
.data
hello: .asciiz "Hello World\n"
```

```
.text
main:
    la $a0, hello
    li $v0, 4
    syscall

    li $v0, 10
    syscall
```

2) Add the numbers in an array, and print the total.

```
.data
data:  .word 33
       .word 44
       .word 66
       .word 0

.text
main:  la $s0, data
       li $s1, 0

loop:  lw $s2, 0($s0)
       beq $s2, $0, loop_end

       add $s1, $s1, $s2
       addi $s0, $s0, 4
       j loop

loop_end:
       move $a0, $s1
       li $v0, 1
       syscall

       li $v0, 10
       syscall
```

3) Here's a variation where we add up the ASCII values of the letters in a string. Replace Bill Gates with your name. What do you add up to?

```
.data
name:  .asciiz "Bill Gates"

.text
main:  la $s0, name
       li $s1, 0

loop:  lb $s2, 0($s0)
       beq $s2, $0, loop_end

       add $s1, $s1, $s2
       addi $s0, $s0, 1
       j loop

loop_end:
       move $a0, $s1
       li $v0, 1
       syscall

       li $v0, 10
       syscall
```

Some more detail on the sys call stuff — the MIPS simulator has some built in subroutines that handle output. You put a number into register \$v0, and other information into register \$a0, and then do a sys call instruction. In the hello world example, you're using the system to print out the message.

A system call with \$v0 = 10 halts the system.

4) Hack the code to print out "hello world" ten times....
and make sure that it would be easy to change it so that it prints out the message 100 times. In other words, use a loop.... a little bit of thinking, and the way to do it should be obvious.