CS2630 Computer Organization

Lab 3: Syscalls, I/O, and structs in MIPS

Learning objectives
- Predict what a program with a syscall does
- Write an assembly program using a syscall
- Predict what a program that accesses the data segment does
- Write an assembly program that uses the data segment
- Translate a program that uses a struct
- Interpret an assembly program that uses a linked structure

Submission
Submit only:
- Answers to two discussion posts in Part 3
- the following files on ICON to the dropbox for "Lab 3: Syscalls, I/O, structs in MIPS"
  - part2b1.s
  - part2b2.s

Part 1: Syscalls

System calls, or syscalls, are procedures provided by the operating system to do tasks like process input/output and allocate memory.

  a. Predicting the behavior of syscalls

Read about the syscalls in MARS (https://courses.missouristate.edu/KenVollmar/mars/Help/SyscallHelp.html) or in MARS under Help | MIPS | Syscalls), specifically the introduction, how to use..., Example, and understanding what the Table of Available Services means.

Using the documentation, for each of the following programs, write what you expect the behavior will be. You should not run the program yet.

1.
li $v0, 1
li $a0, 5
syscall

2.
li $v0, 5
syscall
addu $a0, $v0, $zero
li $v0, 1
syscall
3. $v0, 11
$la, 65
syscall
$v0, 11
$la, 66
syscall
$v0, 11
$la, 67
syscall

b. Testing your hypotheses

1. Run the 1st program in MARS and observe the behavior. If your prediction differs, say why.

2. Run the 2nd program in MARS and observe the behavior. If your prediction differs, say why.

3. Run the 3rd program in MARS and observe the behavior. If your prediction differs, say why.

Part 2: Incorporating the data segment

a. Defining and manipulating data in the data segment

i. Run the program in MARS and observe its behavior. Based on your observations alone, what does the la instruction do and what does the .word directive do? You may single-step through the program and look at memory/register values however you like, but do not read documentation about la and .word to compose your answer.

ii. Read documentation about la pseudo-instruction and .word directive. Does it match your conclusion?

.data
stuff: .word 10 11 12

.text
li $v0, 1
la $s0, stuff
lw $a0, 0($s0)
system

addiu $s0, $s0, 8
li $v0, 1
lw $a0, 0($s0)
system
```assembly
addiu $s0, $s0, -4
li $v0, 1
lw $a0, 0($s0)
syscall

(Outside of class: for another explanation of using labels, you may watch MARS Lesson 3: Assembler and labels https://uiowa.instructure.com/courses/65872/modules/items/1510969)

b. Using the data segment and syscalls

1. Complete this program (save it as part2b1.s) so that it prints the elementwise sum of the things and others arrays. E.g., it would print

50 70
(HINT: you need to print a space)

.data
things: .word 10 20
others: .word 40 50

.text
la $s0, things
la $s1, others

2. Write a program (save it as part2b2.s), that reads an integer (i.e., "read integer" syscall) and compares it to a secret integer. The program prints the letter "N" if the guess was wrong and "Y" if the guess was right. The secret integer should be kept in the data segment of the program.

c. Fun with data

Predict: what will be the result in register $s0 when this program is finished running?

.text
   la $t2, wow
   lw $t1, 0($t2)
   addiu $t1, $t1, 0xFED
   sw $t1, 0($t2)
wow: addiu $s0, $zero, 1

Once you make a prediction, see https://piazza.com/class/j6l8nniwzot5ys?cid=63 to discuss the problem.
Part 3: Representing structs (objects)

Read slides 1-9 from today's Notes to learn about how objects can be represented in memory using the case study of a linked list.

a) Answer to "Peer instruction" on slide 8? Post your answer on https://piazza.com/class/j6l8nniwt5ys?cid=61

Refer to the provided code in listnode.asm
(http://homepage.cs.uiowa.edu/~bdmyers/cs2630_fa17/public/lectures/lecture-011/listnode/) for the remaining questions.

Put your answers to b-f in a post on https://piazza.com/class/j6l8nniwt5ys?cid=62

b) What syscall does it use to allocate a new ListNode object?

c) How big is a ListNode object (in number of bytes)? At what offset from the base address of a ListNode object is the data field? At what offset is the next field?

d) Why is an $s_\_register written to in this line of the append method? Be more specific than "the value must be saved".

```
addu $s0, $zero, $a0       # s0 = this
```

e) After running the program, what at what address is the ListNode 2, ListNode 6, ListNode 3, and ListNode 0 at? (HINT: the answer is not 0x10010000, 0x10010004, 0x10010008, 0x1001000C).

f) Write one question you have about the listnode.asm code.