Ho Chi Minh City University of Technology

Faculty of Computer Science and Engineering
Course: Computer Architecture Lab (CO2008)

Lab 3

Advanced instructions

Ho Chi Minh City, November 2024

Contents

1 Introduction

- **The main purpose of this session is to get familiar with floating-point instructions and read/write file.**
- Students must submit their answers to the BKeL system no later than the last period of the lab section. Then, the instructor will evaluate all students' work during the lab section's final period.

2 Exercises

2.1 Exercise 1

Write a MIPS program to find the solution to the equation:

$$ax^2 + bx + c = 0 \tag{1}$$

where a, b, and c are floating-points number inserted by the user. The result should be in 1 of these 3 cases:

- There are 2 solutions to x: print those results to the screen after the text: "x1 = " and "x2 = ".
- There is 1 solution to x: print the result after the text: "There is one solution, x = ".
- There is no real solution: print "There is no real solution".

Note that the solutions printed must be in floating-point numbers. The user will input the floating-points numbers after the prompts: "Please insert a: ", "Please insert b: ", and "Please insert c: ".

For example, if the user inserted **a=1.2, b=9.6, c=1** then the result should be: **x1 = -7.8944** and **x2 = -0.1056**

In another example, if the user inserted **a=2.4, b=-3.2, c=3.1** then the result should be: **There is no real solution**

2.2 Exercise 2

Write a MIPS program to calculate the following integral:

$$f(x) = \int_v^u \frac{ax^6 + bx^5 + cx}{d^4 + e^3} \tag{2}$$

where u, v, a, b, c, d, and e are floating-point numbers chosen by the user. The user will input the floating-point numbers after the prompts: "Please insert a: ", "Please insert b: ", etc. For example, user inserted **u=1, v=2, a=3, b=4, c=5, d=6, e=7** then the result should be: **-0.0634**

2.3 Exercise 2

To allocate memory, please refer to the following syscall:

```
li $v0, 9 # system call code for dynamic allocation
li $a0, 24 # $a0 contains number of bytes to allocate
```

After the above system call, $v0 contains the first address in heap memory that is allocated. Then, accessing the allocated memory can be done by lw/sw, for example:

```
# Trying to write to allocated space
addi $t0, $zero, 2021
sw $t0, 0($v0)
```

The followings are instructions used to access a file (open/close/read/write):

```
# Sample MIPS program that writes to a new file.
# by Kenneth Vollmar and Pete Sanderson
.data
fout: .asciiz "testout.txt" # filename for output
msg1: .asciiz "Before read: "
msg2: .asciiz "After read: "
buffer_write: .asciiz "The quick brown fox jumps over the
lazy dog.\n"
buffer_read: .asciiz "----------------------------------------\n"
.text
############################################################
# Open (for writing) a file that does not exist
li $v0, 13 # system call for open file
la $a0, fout # output file name
li $a1, 1 # Open for writing (flags are 0: read, 1: write)
li $a2, 0 # mode is ignored
syscall # open a file (file descriptor returned in $v0)
move $s6, $v0 # save the file descriptor
############################################################
# Write to file just opened
li $v0, 15 # system call for write to file
move $a0, $s6 # file descriptor
la $a1, buffer_write # address of buffer from which to write
li $a2, 44 # hardcoded buffer length
syscall # write to file
############################################################
# Close the file
```

```
    li $v0, 16 # system call for close file
    move $a0, $s6 # file descriptor to close
    syscall # close file
    ##############################################################
    ##############################################################
    # Open (for reading) a file
    li $v0, 13 # system call for open file
    la $a0, fout # input file name
    li $a1, 0 # Open for reading (flags are 0: read, 1: write)
    li $a2, 0 # mode is ignored
    syscall # open a file (file descriptor returned in $v0)
    move $s6, $v0 # save the file descriptor
    ##############################################################
    # Read from file
    li $v0, 14 # system call for read
    move $a0, $s6 # file descriptor
    la $a1, buffer_read # address of buffer read
    li $a2, 44 # hardcoded buffer length
    syscall # read file
```

Please do the followings:

1. Manually create a line of text in format <name>,<id>,<address>,<age>,<religion>in a text file by using a text editor. The file must be named: **raw_input.txt**.
   An example text should be: **Davy Jones,2251234,168 Ly Thuong Kiet St District 10 Ward 14 HCMC,69,None**

2. Open the file to read the text you inserted.

3. Declare a string in the heap memory with dynamically allocated memory. The size of the string must be large enough to store the text in the text file.

4. Copy the line from the text file to the string in the memory.

5. Print the string to the terminal in the following format:

   (a) ——Student personal information——

   (b) Name: <name>

   (c) ID: <ID>

   (d) Address: <address>

   (e) Age: <age>

   (f) Religion: <religion>

The output in the terminal for the former example input should be:
——**Student personal information**——
Name: Davy Jones
ID: 2251234
Address: 168 Ly Thuong Kiet St District 10 Ward 14 HCMC
Age: 69
Religion: None

6. The printed result should also be written in a file named **formatted_result.txt**

2.4 Exercise 3

Show that when randomly taking a point in a square with side 1, the probability that that point lies in the circle inscribed in the square is π /4 (Area of the circle divided by area of the square). Write a MARS MIPS program using the function set seed (syscall 40) by time (syscall 30) and random number generation functions to generate real number coordinates (x, y) ($0 < x < 1$, $0 < y < 1$) of 50,000 points is used to determine the PI number as suggested above. Save the results of running the program to the PI.TXT file including the following information:
Number of points within the circle: ddddd/50000
Calculated PI number: f.ffffff