

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	2
2	Exercises	2
2.1	Exercise 1	2
2.2	Exercise 2	2
2.3	Exercise 3	3



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:

$$ax^2 + bx + c = 0 \quad (1)$$

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:

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



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 3

To allocate memory, please refer to the following syscall:

```
1  li $v0, 9 # system call code for dynamic allocation
2  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:

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

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

```
1  # Sample MIPS program that writes to a new file .
2  # by Kenneth Vollmar and Pete Sanderson
3  .data
4  fout: .asciiz "testout.txt" # filename for output
5  msg1: .asciiz "Before read: "
6  msg2: .asciiz "After read: "
7  buffer_write: .asciiz "The quick brown fox jumps over the
8  lazy dog.\n"
9  buffer_read: .asciiz "_____\n"
10 .text
11 #####
12 # Open (for writing) a file that does not exist
13 li $v0, 13 # system call for open file
14 la $a0, fout # output file name
15 li $a1, 1 # Open for writing (flags are 0: read, 1: write)
16 li $a2, 0 # mode is ignored
17 syscall # open a file (file descriptor returned in $v0)
18 move $s6, $v0 # save the file descriptor
19 #####
20 # Write to file just opened
21 li $v0, 15 # system call for write to file
22 move $a0, $s6 # file descriptor
23 la $a1, buffer_write # address of buffer from which to write
24 li $a2, 44 # hardcoded buffer length
25 syscall # write to file
26 #####
27 # Close the file
```



```
27  li $v0, 16 # system call for close file
28  move $a0, $s6 # file descriptor to close
29  syscall # close file
30  #####
31  #####
32  # Open (for reading) a file
33  li $v0, 13 # system call for open file
34  la $a0, fout # input file name
35  li $a1, 0 # Open for reading (flags are 0: read, 1: write)
36  li $a2, 0 # mode is ignored
37  syscall # open a file (file descriptor returned in $v0)
38  move $s6, $v0 # save the file descriptor
39  #####
40  # Read from file
41  li $v0, 14 # system call for read
42  move $a0, $s6 # file descriptor
43  la $a1, buffer_read # address of buffer read
44  li $a2, 44 # hardcoded buffer length
45  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**