**LAB REPORT**

**IT3280E– 152049– Assembly Language and Computer Architecture Lab**

**Lab 05: Character string with ECALL function**

# **Assignment 1:**

*Create a new project to implement the program in Home Assignment 1. Compile and*

*upload to simulator. Run and observe the result. Go to Data Segment, check how test*

*string are stored and packed in memory.*

A screenshot of a computer

Description automatically generated

* Analysis:

+ Intitialize:

* The program defines string *“Hello World”* in the **.data** section using directives **.asciz**.
* Load the service number *4* into register **a7** to print the string. The adress of the string (**test**) is loaded into **a0** to specific which string to print.

+ Running:

* The program is executed and prints out the string *“Hello World”* to the console log.
* The program then exits.
* The **result** is:

A screenshot of a computer

Description automatically generated

* Data Segment:

A screenshot of a computer

Description automatically generated

**+** In memory, each character of the string is stored in consecutive bytes.

**+** After the *“d”,* a null character *(\0)* is automatically added to mark the end of the string (stores null-terminated string)

**Assignment 2:**

*Create a new project to print the sum of two register $s0 and $s1 according to this*

*format:*

*“The sum of (s0) and (s1) is (result)”*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with green text

Description automatically generated

* The result will be: *(for s0 = 3, s1 = 4)*

A screenshot of a computer

Description automatically generated

# **Assignment 3:**

*Create a new project to implement the program in Home Assignment 2. Add more*

*instructions to assign a test string for y variable, and implement strcpy function. Compile and upload to simulator. Run and observe the result*

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

* Execution and Output:

+ Copied string: *Hello hmp*

+ The program successfully copies the string *"Hello hmp"* from y to x using the **strcpy** function.

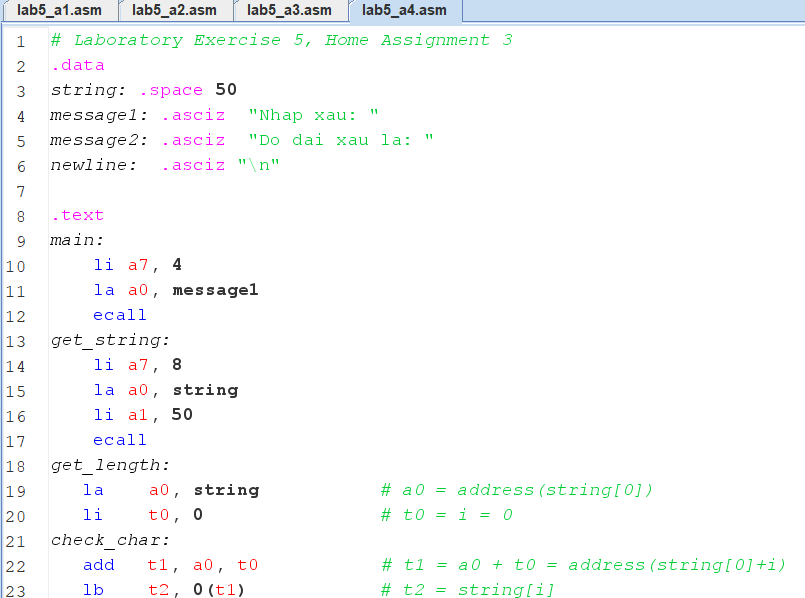
+ After copying, it prints the message followed by the copied string.

A screenshot of a computer

Description automatically generated

# **Assignment 4:**

*Accomplish the Home Assignment 3 with ecall function to get a string from dialog, and show the length to message dialog*



A screenshot of a computer code

Description automatically generated

* The **output** is

A screenshot of a computer

Description automatically generated

# **Assignment 5:**

*Write a program that let user input a string. Input process will be terminated when user press Enter or then length of the string exceed 20 characters. Print the reverse string.*

|  |  |
| --- | --- |
| # Laboratory Exercise 5, Assignment 5  .data  string: .space 21 *# buffer for up to 20 characters plus null terminator*  message1: .asciz "Enter a string (max 20 characters): " *# input prompt*  newline: .asciz "\n" *# newline for formatting*  msg\_reverse: .asciz "Reversed string: " *# reversed string message*  .text  .globl main  main:  *#print input prompt*  li a7, 4  la a0, message1  ecall  *#get input string*  jal ra, input\_string  *#reverse string*  jal ra, reverse\_string  *#print reversed string message*  li a7, 4  la a0, msg\_reverse  ecall  *#print reversed string*  li a7, 4  la a0, string  ecall  *# print newline*  li a7, 4  la a0, newline  ecall  *# exit program*  li a7, 10  ecall  input\_string:  li t0, 0  *#initialize index* | input\_loop:  li a7, 12 *#read character*  ecall  mv t1, a0  li t2, 10 *#enter key*  beq t1, t2, input\_end *#stop if enter*  la t3, string  add t3, t3, t0  sb t1, 0(t3) *#store character*  addi t0, t0, 1 *#increment index*  li t4, 20 *#max length check*  bge t0, t4, input\_end  j input\_loop  input\_end:  la t3, string  add t3, t3, t0  sb zero, 0(t3) *# store null terminator*  jr ra  reverse\_string:  li t1, 0 *#start index*  la t5, string  li t0, 0 *#string length*  mv t2, t5  length\_loop:  lb t3, 0(t2)  beq t3, zero, length\_done  addi t2, t2, 1  addi t0, t0, 1  j length\_loop  length\_done:  addi t4, t0, -1  reverse\_loop:  bge t1, t4, reverse\_end  add t2, t5, t1  add t6, t5, t4  lb t3, 0(t2)  lb a0, 0(t6)  sb a0, 0(t2)  sb t3, 0(t6)  addi t1, t1, 1  addi t4, t4, -1  j reverse\_loop  reverse\_end:  jr ra |

* The **output** is

A screenshot of a computer

Description automatically generated

* Main Function:

+ Displays a prompt for input.

+ Calls functions to read the input string and reverse it.

+ Prints the reversed string.

* Input String Function:

+ Reads characters until Enter is pressed or 20 characters are entered.

+ Adds a null terminator at the end.

* Reverse String Function:

+ Finds the length of the string.

+ Reverses the string in place by swapping characters from start to end.