Bài thực hành số 5

Lớp: 139365 – Học phần: Thực Hành Kiến Trúc Máy Tính

Họ và Tên: Nguyễn Anh Thứ MSSV: 20215144

Bài 1:

Graphical user interface, text, application, email

Description automatically generated

.data: định nghĩa phần dữ liệu của chương trình.

test: .asciiz "Hello World": định nghĩa một chuỗi ký tự được lưu trữ trong bộ nhớ, với tên biến là "test" và giá trị là "Hello World". Trong đó, ".asciiz" là một từ khóa để định nghĩa chuỗi ký tự, và "z" cuối cùng đảm bảo rằng chuỗi kết thúc bằng ký tự null.

.text: định nghĩa phần mã của chương trình.

li $v0, 4: lệnh này gán giá trị 4 cho thanh ghi $v0, xác định loại hệ thống gọi, hoặc "syscall", sẽ được gọi. . Giá trị được gán cho thanh ghi $v0 (giá trị 4) xác định rằng chương trình sẽ gọi hệ thống gọi in chuỗi



la $a0, test: lệnh này lấy địa chỉ của biến "test" (chuỗi "Hello World") và lưu nó vào thanh ghi $a0. Thanh ghi $a0 là thanh ghi đặc biệt trong MIPS, được sử dụng để truyền đối số cho các hệ thống gọi.





syscall: lệnh này gọi hệ thống gọi để hiển thị chuỗi "Hello World" trên màn hình console. Nội dung chuỗi cần được in được truyền vào hệ thống gọi thông qua thanh ghi $a0.

Graphical user interface, text, application

Description automatically generated

Bài 2:

Code của chương trình:

.data

result: .asciiz "The sum of "

s0: .asciiz "(s0) and "

s1: .asciiz "(s1) is "

newline: .asciiz "\n"

.text

main:

# Nhập giá trị cho $s0

li $v0, 5

syscall

move $s0, $v0 # Lưu giá trị nhập vào vào $s0

# Nhập giá trị cho $s1

li $v0, 5

syscall

move $s1, $v0 # Lưu giá trị nhập vào vào $s1

# Tính tổng của $s0 và $s1

add $t0, $s0, $s1

# In chuỗi "The sum of "

li $v0, 4

la $a0, result

syscall

# In giá trị của $s0

li $v0, 1

move $a0, $s0

syscall

# In chuỗi " and "

li $v0, 4

la $a0, s0

syscall

# In giá trị của $s1

li $v0, 1

move $a0, $s1

syscall

# In chuỗi " is "

li $v0, 4

la $a0, s1

syscall

# In kết quả của phép cộng

li $v0, 1

move $a0, $t0

syscall

# In dấu xuống dòng

li $v0, 4

la $a0, newline

syscall

# Kết thúc chương trình

li $v0, 10

syscall

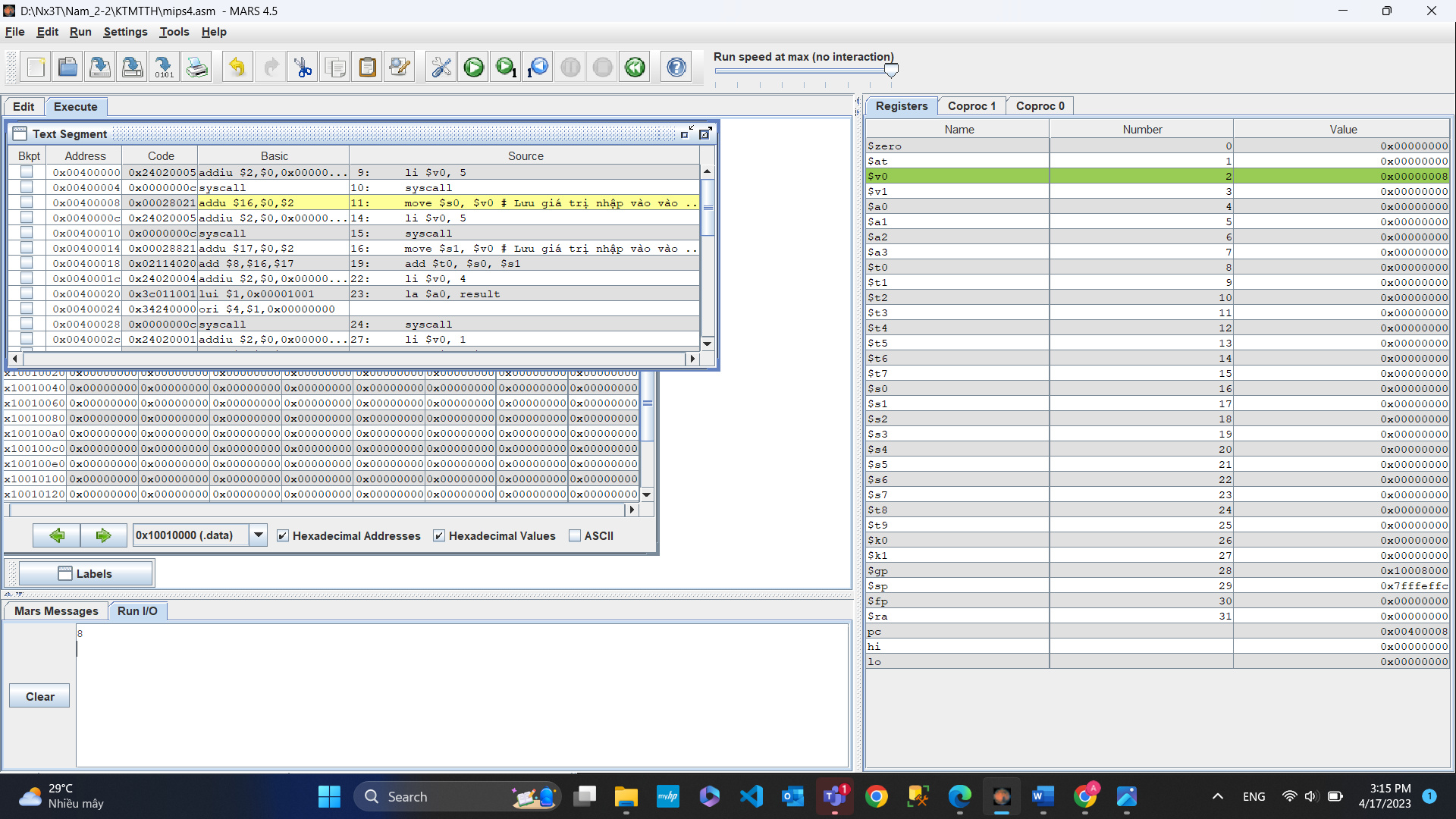
Giải thích:

Khai báo các chuỗi chứa thông điệp cần in ra màn hình và các biến để lưu giá trị đầu vào và kết quả tính toán.

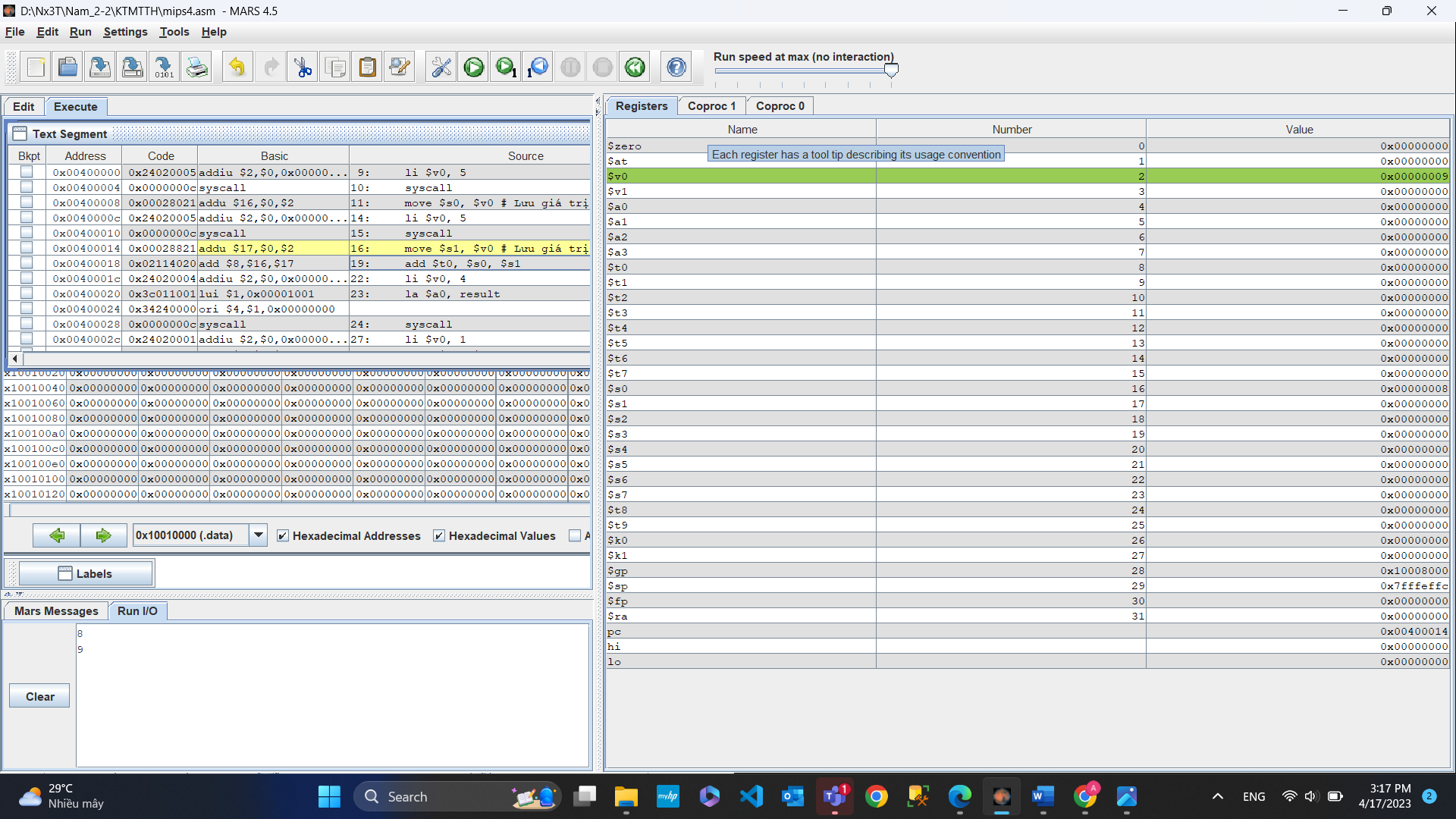
Text

Description automatically generated

Nhập giá trị cho biến $s0 bằng lệnh syscall và lưu giá trị vào $v0.



Nhập giá trị cho biến $s1 bằng lệnh syscall và lưu giá trị vào $v0.



Tính tổng của $s0 và $s1 bằng lệnh add và lưu vào biến $t0.



In ra màn hình chuỗi "The sum of " bằng lệnh syscall và đưa địa chỉ của chuỗi vào thanh ghi $a0 bằng lệnh la.





In ra màn hình giá trị của $s0 bằng lệnh syscall và đưa giá trị của $s0 vào thanh ghi $a0 bằng lệnh move.





In ra màn hình chuỗi " and " bằng lệnh syscall và đưa địa chỉ của chuỗi vào thanh ghi $a0 bằng lệnh la.



A picture containing logo

Description automatically generated

In ra màn hình giá trị của $s1 bằng lệnh syscall và đưa giá trị của $s1 vào thanh ghi $a0 bằng lệnh move.

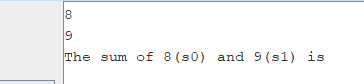


A picture containing graphical user interface

Description automatically generated

In ra màn hình chuỗi " is " bằng lệnh syscall và đưa địa chỉ của chuỗi vào thanh ghi $a0 bằng lệnh la.





In ra màn hình giá trị của $t0 (tổng của $s0 và $s1) bằng lệnh syscall và đưa giá trị của $t0 vào thanh ghi $a0 bằng lệnh move.



Graphical user interface, text

Description automatically generated with medium confidence

In ra màn hình dấu xuống dòng bằng lệnh syscall và đưa địa chỉ của chuỗi vào thanh ghi $a0 bằng lệnh la.



A picture containing text

Description automatically generated

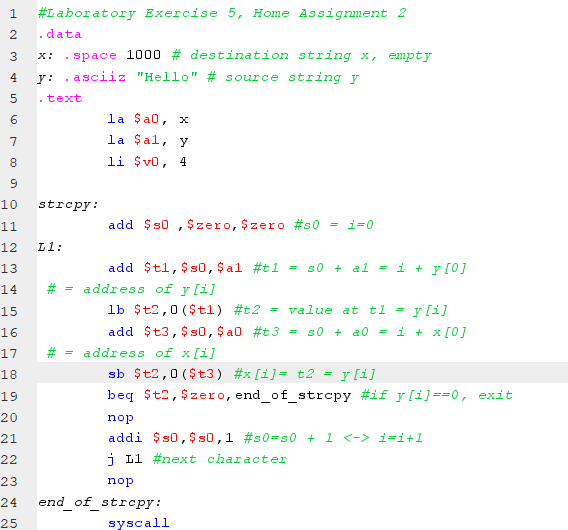
Kết thúc chương trình bằng lệnh syscall.



Text

Description automatically generated

Bài 3:



Giải thích code:

Dòng đầu tiên đặt thanh ghi $a0 trỏ đến vị trí đầu tiên của xâu đích x.

Dòng thứ hai đặt thanh ghi $a1 trỏ đến vị trí đầu tiên của xâu nguồn y.

Dòng thứ ba sử dụng lệnh li (load immediate) để đặt giá trị 4 vào thanh ghi $v0, chứa mã syscall cho lệnh in chuỗi kí tự (syscall 4).

Tiếp theo, chương trình bắt đầu quá trình sao chép bằng cách thiết lập giá trị ban đầu cho biến $s0, tương đương với chỉ số của ký tự đầu tiên trong xâu đích.

Trong vòng lặp L1, chương trình sử dụng địa chỉ của ký tự thứ i trong xâu nguồn (y) để sao chép ký tự tương ứng vào vị trí tương ứng trong xâu đích (x).

Sau đó, biến $s0 được tăng lên một đơn vị để chỉ đến ký tự tiếp theo trong xâu đích, và chương trình tiếp tục vòng lặp cho đến khi gặp ký tự NULL kết thúc chuỗi (được biểu diễn bằng giá trị 0).

Cuối cùng, chương trình gọi lệnh syscall để in ra chuỗi kí tự xâu đích đã được sao chép (syscall 4).

Bài 4:

Code:

Text

Description automatically generated

Text

Description automatically generated

Giải thích:

Hàm get\_string:

Chương trình trên cho phép người dùng nhập vào một xâu và tính toán độ dài của xâu đó. Ban đầu, chương trình in ra chuỗi "Nhap xau:" để yêu cầu người dùng nhập xâu. Sau đó, chương trình sử dụng lệnh syscall với mã số 8 để đọc xâu người dùng nhập vào và lưu vào vùng nhớ có tên là "string" với giới hạn tối đa là 50 ký tự.

Tiếp theo, chương trình gọi hàm get\_length để tính toán độ dài của xâu. Hàm get\_length sử dụng một vòng lặp để kiểm tra từng ký tự trong xâu. Ban đầu, giá trị độ dài của xâu (length) được khởi tạo bằng 0 và biến i cũng được khởi tạo bằng 0. Sau đó, vòng lặp kiểm tra từng ký tự trong xâu bằng cách sử dụng lệnh lb để đọc ký tự tại vị trí thứ i của xâu và so sánh với ký tự null. Nếu ký tự null được tìm thấy, chương trình kết thúc vòng lặp và trả về giá trị độ dài của xâu bằng cách sử dụng lệnh syscall với mã số 56 để in ra chuỗi "Do dai la " kèm theo giá trị độ dài của xâu được lưu trong thanh ghi $v0.

Bài 5:

Code:

Text

Description automatically generated with low confidence

Text

Description automatically generated with low confidence

Giải thích code:

Ban đầu, chương trình sẽ yêu cầu người dùng nhập vào một chuỗi bằng cách sử dụng hàm syscall với mã số 8. Chuỗi này sẽ được lưu trữ trong vùng nhớ có tên là string và có độ dài tối đa là 50 ký tự. Sau khi nhập chuỗi xong, chương trình tiếp tục thực hiện hàm get\_length để tính độ dài của chuỗi vừa nhập.

Hàm get\_length bắt đầu bằng việc khởi tạo giá trị độ dài chuỗi là 0, rồi kiểm tra từng ký tự trong chuỗi bằng cách sử dụng vòng lặp while. Nếu ký tự hiện tại là null character (kết thúc chuỗi), vòng lặp sẽ dừng lại. Ngược lại, chương trình sẽ tăng biến đếm độ dài của chuỗi lên 1 và tiếp tục kiểm tra ký tự tiếp theo.

Sau khi tính được độ dài của chuỗi, chương trình sẽ bắt đầu in chuỗi đó theo thứ tự ngược lại. Ban đầu, chương trình tạo một vùng nhớ mới có tên là rstring để lưu trữ chuỗi đảo ngược. Sau đó, chương trình sẽ duyệt chuỗi ban đầu từ cuối về đầu, mỗi lần đọc một ký tự, ký tự đó sẽ được ghi vào vùng nhớ mới được tạo ở bước trước. Cuối cùng, chương trình sẽ in chuỗi đảo ngược ra màn hình bằng cách sử dụng hàm syscall với mã số 59.