



09/04/2024

STRING COPY

Prepared For :
Dr.lamiaa

Problem Description:

The problem requires the development of a program in MIPS assembly language that facilitates the input, copying, and output of strings. Specifically, the program should prompt the user to input a string (with a maximum length of 100 characters), copy this input string into another string, and then print the copied string. The program also includes standard messages for input and output prompts.

SAMPLE INPUT:

```
ENTER STRING BUT ITS LENGTH LESS THAN OR EQUAL 100  
HELLO, THIS IS A TEST STRING!
```

SAMPLE OUTPUT:

```
THE OUTPUT IS  
HELLO, THIS IS A TEST STRING!
```

Updates Made to the Example:

1. Added Input and Output Messages:

- To enhance user interaction and provide clear instructions, input and output messages were incorporated. These messages prompt the user to enter a string and inform them of the output.

2. Defined a Newline Character:

- A newline character was defined for better formatting and readability of the output.

3. Implemented String Copy Procedure (`strcpy`):

- A procedure named `strcpy` was introduced to facilitate string copying. This procedure takes two arguments: the source string (`string1`) and the destination string (`string2`). It iterates through each character of the source string and copies it to the destination string until it encounters the null terminator, indicating the end of the string.

4. Updated Main Function:

- After prompting the user for input, the program calls the `strcpy` procedure to copy the input string into another string (`string2`). Subsequently, the copied string (`string2`) is printed, and the program exits.

1. Input Prompt:

- The program begins by prompting the user to input a string using the `.asciiz` directive. The message instructs the user to enter a string with a length of 100 characters or less.

2. String Input:

- Upon receiving the input, the program stores it in the memory location reserved for `string1` using the appropriate system call (`syscall`).

3. Newline Character Definition:

- A newline character (`\n`) is defined to facilitate formatting when printing messages or strings.

4. String Copy Procedure (`strcpy`):

- The `strcpy` procedure is implemented to copy one string into another. It utilizes a loop to iterate through each character of the source string (`string1`) and copy it into the destination string (`string2`). The loop continues until the null terminator is encountered, indicating the end of the string.

5. Main Function Execution:

In the main function, after receiving the input string, the program prints a newline character for formatting and then prompts the user with an output message using `syscall`.

The `strcpy` procedure is called with appropriate arguments to copy `string1` into `string2`.

The copied string (`string2`) is printed using `syscall`.

6. Program Termination:

Finally, the program exits gracefully using the appropriate system call (`syscall`).

The screenshot displays the MARS 4.5 IDE interface for the file `C:\Users\MF\Desktop\assignment (2)\string copy\stringcopy.asm`. The main window shows the assembly code with the following instructions:

```
4194304 0x24020004: addiu $2,$0,4      11: li $v0,4
4194308 0x0c011001: lui $1,4097                     12: la $a0,input
4194312 0x34240000: ori $4,$1,0
4194316 0x0000000c: syscall                          13: syscall
4194320 0x0c011001: lui $1,4097                     14: la $a0,string1
4194324 0x34240004: ori $4,$1,49
4194328 0x24050004: addiu $5,$0,100                17: li $a1,100
4194332 0x24020004: addiu $2,$0,8                  18: li $v0,8
4194336 0x0000000c: syscall                          19: syscall
4194340 0x24020004: addiu $2,$0,4                  21: li $v0,4
4194344 0x0c011001: lui $1,4097                     22: la $a0,$b1
4194348 0x34240004: ori $4,$1,67
4194352 0x0000000c: syscall                          23: syscall
4194356 0x24020004: addiu $2,$0,4                  25: li $v0,4
4194360 0x0c011001: lui $1,4097                     26: la $a0,output
4194364 0x34240003: ori $4,$1,55
4194368 0x0000000c: syscall                          27: syscall
```

The Registers window on the right shows the state of the registers, with `$a0` (address 4194304) and `$a1` (address 4194316) pointing to the `stringcopy.asm` file. The `$v0` register contains the value 10.

The Data Segment window at the bottom shows the memory layout, with the `string1` and `string2` buffers located at addresses 268500992 and 268501040 respectively.

The Mars Messages window at the bottom displays the program's output:

```
enter string but its length leesthan or equal 100
Hello, this is a test string!

the output is
Hello, this is a test string!
-- program is finished running --
```