**20184270\_Phi Quoc Hung**

**Lab Session 5 Report**

***Assignment 1:***

- Assembly code:

#Lab 5, sample code & exercise 1

.data

test: .asciiz "Hello World" #Save string “Hello World” in variable test

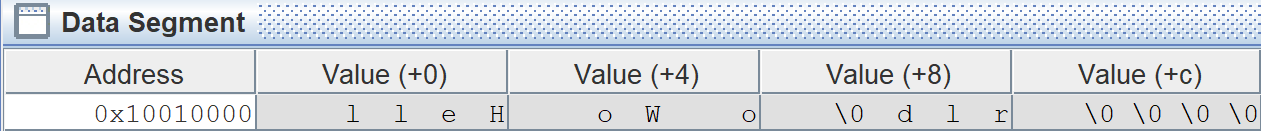
.text

li $v0,4 #Load immediate value 4 to register $v0

la $a0,test #Load string test to register $a0

syscall

- Explain the result:



Each letter is considered as a word. Each address can contains 4 words, and in MIPS, the MSB is at the rightmost part of a address value. So the string is stored from right to left with 4 letter in one address

***Assignment 2:***

- Assembly code

.data

msg1: .asciiz "The sum of "

msg2: .asciiz " and "

msg3: .asciiz " is "

.text

li $s0, 10

li $s1, 20

add $s2,$s1,$s0

#Print msg1

li $v0,4

la $a0, msg1

syscall

#Print $s0

li $v0,1

move $a0, $s0

syscall

#Print msg2

li $v0,4

la $a0, msg2

syscall

#Print $s1

li $v0,1

move $a0, $s1

syscall

#Print msg 3

li $v0, 4

la $a0, msg3

syscall

#Print $s2

li $v0,1

move $a0, $s2

syscall

***Assignment 3:***

- Assembly Code

#Laboratory Exercise 5, Sample Code2

.data

x: .space 1000 # destination string x, empty

y: .asciiz "Ex3 is so hard" # source string y

.text

strcpy:

add $s0, $zero,$zero #s0 = i=0

L1:

la $a1, y

add $t1 ,$s0,$a1 #t1 = s0 + a1 = i + y[0]

# = address of y[i]

lb $t2, 0($t1) #t2 = value at t1 = y[i]

la $a0, x

add $t3, $s0, $a0 #t3 = s0 + a0 = i + x[0]

# = address of x[i]

sb $t2, 0($t3) #x[i]= t2 = y[i]

beq $t2, $zero, end\_of\_strcpy #if y[i]==0, exit

nop

addi $s0,$s0,1 #s0=s0 + 1 <-> i=i+1

j L1 #next character

nop

syscall

end\_of\_strcpy:

- Explain:

The code copy value of string y to string x and print it to the data segment memory

***Assignment 4:***

- Assembly code:

#Laboratory Exercise 5, Sample Code3

.data

string: .space 50

Message1: .asciiz "Nhap xau"

Message2: .asciiz "Do dai la "

.text

main:

get\_string: # TODO

li $v0, 54

la $a0, Message1

la $a1, string

la $a2, 40

syscall

get\_length:

la $a0, string # a0 = Address(string[0])

xor $v0, $zero, $zero # v0 = length = 0

xor $t0, $zero, $zero # t0 = i = 0

check\_char:

add $t1, $a0, $t0 # t1 = a0 + t0

#= Address(string[0]+i)

lb $t2, 0($t1) # t2 = string[i]

beq $t2, $zero, end\_of\_str # Is null char?

addi $v0, $v0, 1 # v0=v0+1->length=length+1

addi $t0, $t0, 1 # t0=t0+1->i = i + 1

j check\_char

end\_of\_str:

end\_of\_get\_length:

sub $t0,$t0,1

print\_length: # TODO

la $a0,Message2

li $v0,4

syscall

li $v0,1

move $a0, $t0

syscall

***Assignment 5:***

- Assembly code:

.data

string: .space 21

reverse: .space 21

Message1: .asciiz "Nhap xau:"

Message2: .asciiz " Xau dao la:"

.text

main:

la $a0, Message1

li $v0, 4

syscall

get\_string:

li $v0, 8

la $a0, string

la $a1, 21

syscall

get\_length:

la $a0, string # a0 = Address(string[0])

xor $v0, $zero, $zero # v0 = length = 0

xor $t0, $zero, $zero # t0 = i = 0

check\_char:

add $t1, $a0, $t0 # t1 = a0 + t0

#= Address(string[0]+i)

lb $t2, 0($t1) # t2 = string[i]

beq $t2,$zero,end\_of\_str # Is null char?

addi $v0, $v0, 1 # v0=v0+1->length=length+1

addi $t0, $t0, 1 # t0=t0+1->i = i + 1

j check\_char

end\_of\_str:

end\_of\_get\_length:

sub $t4,$v0,1 #length of string

li $t5,0 # counter i=0

reverse\_string:

la $a1,string

add $t1,$t4,$a1

lb $t2,0($t1)

la $a0,reverse

add $t3,$t5,$a0

sb $t2,0($t3)

beq $t4,0,end\_reverse

nop

addi $t5,$t5,1

sub $t4,$t4,1

j reverse\_string

end\_reverse:

print\_length:

la $a0,Message2

li $v0, 4

syscall

la $a0,reverse

li $v0, 4

syscall

\*Quizzes:

1. What the difference between the string in C and Java?

Answer: In C, string is an array, that can be pointed to by a pointer and end with the NULL terminator (\0) . In Java, string is an instance (or object) and can not be treated as an array

2. In C, how many characters can be stored within 8 bytes?.

Answer: 2^8^8 characters, because with 1 byte we canIn C, how many characters can bestoredwithin 8 bytes? store 1 character

3. In Java, how many characters can be stored within 8 bytes?

2^8^4 = 256^4 characters