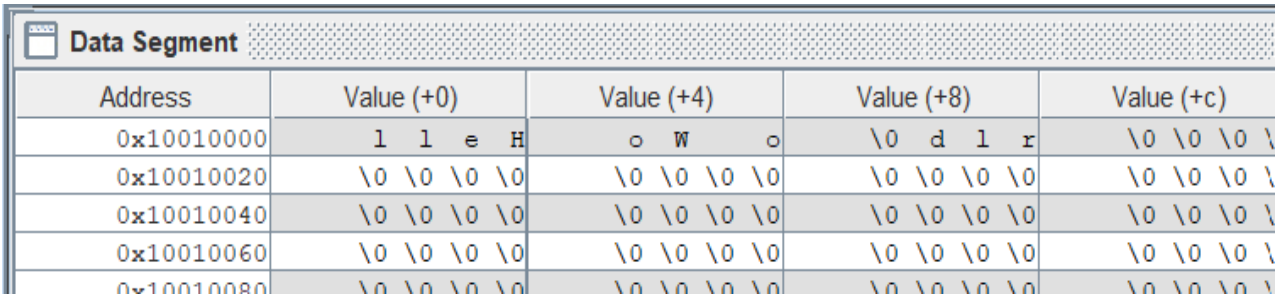
BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 5

Assignment 1



Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x10010000	l l e H	o W o	\0 d l r	\0 \0 \0 \
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \

- Test string “Hello World” được lưu ở các địa chỉ 0x10010000, 0x10010004, 0x10010008
- Được lưu bằng cách gán service number =4 cho $v0 và tải địa chỉ chuỗi test vào biến $a0

Assignment 2

```
.data
Xuat1: .asciiz "Nhap so thu 1 "
Xuat2: .asciiz "Nhap so thu 2 "
Xuat3: .asciiz "Result: "

.text
    la $a0, Xuat1       # print string Xuat1
    li $v0, 4
    syscall

    li $v0, 5       # nhap so nguyen
```

```
syscall
add $t1,$zero,$v0

la $a0, Xuat2       # print string Xuat2
li $v0, 4
syscall

li $v0, 5       # nhap so nguyen
syscall
add $t2,$zero,$v0

add $t3,$t1,$t2

la $a0, Xuat3       # print string Xuat3
li $v0, 4
syscall

li $v0, 1       # In ra tong 2 so
add $a0,$zero,$t3
syscall
```

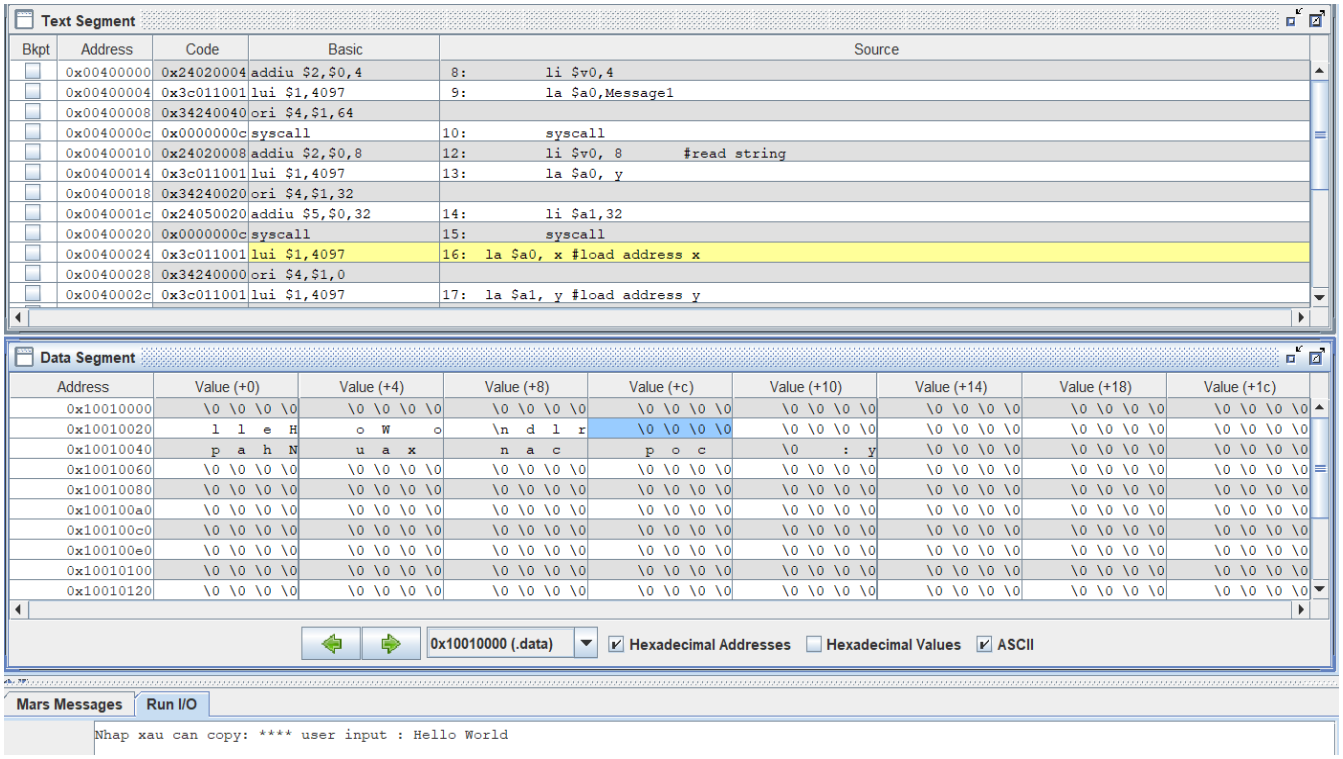
Assignment 3

#Laboratory Exercise 5, Home Assignment 2

```
.data
x: .space 32 # destination string x, empty
y: .asciiz "Hello World"
.text
strcpy:
add $s0,$zero,$zero # $s0 = i = 0
 la $a0, x #load address x
 la $a1, y #load address y
L1:
add $t1,$s0,$a1 # $t1 = $s0 + $a1 = i + y[0]
 # = address of y[i]
lb $t2,0($t1) # $t2 = value at $t1 = y[i]
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

end_of_strcpy:
```

- Bộ nhớ lúc đầu:



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24020004	addiu $2,$0,4	8: li $v0,4
	0x00400004	0x3c011001	lui $1,4097	9: la $a0,Message1
	0x00400008	0x34240040	ori $4,$1,64	
	0x0040000c	0x0000000c	syscall	10: syscall
	0x00400010	0x24020008	addiu $2,$0,8	12: li $v0, 8 #read string
	0x00400014	0x3c011001	lui $1,4097	13: la $a0, y
	0x00400018	0x34240020	ori $4,$1,32	
	0x0040001c	0x24050020	addiu $5,$0,32	14: li $a1,32
	0x00400020	0x0000000c	syscall	15: syscall
	0x00400024	0x3c011001	lui $1,4097	16: la $a0, x #load address x
	0x00400028	0x34240000	ori $4,$1,0	
	0x0040002c	0x3c011001	lui $1,4097	17: la $a1, y #load address y

Data Segment

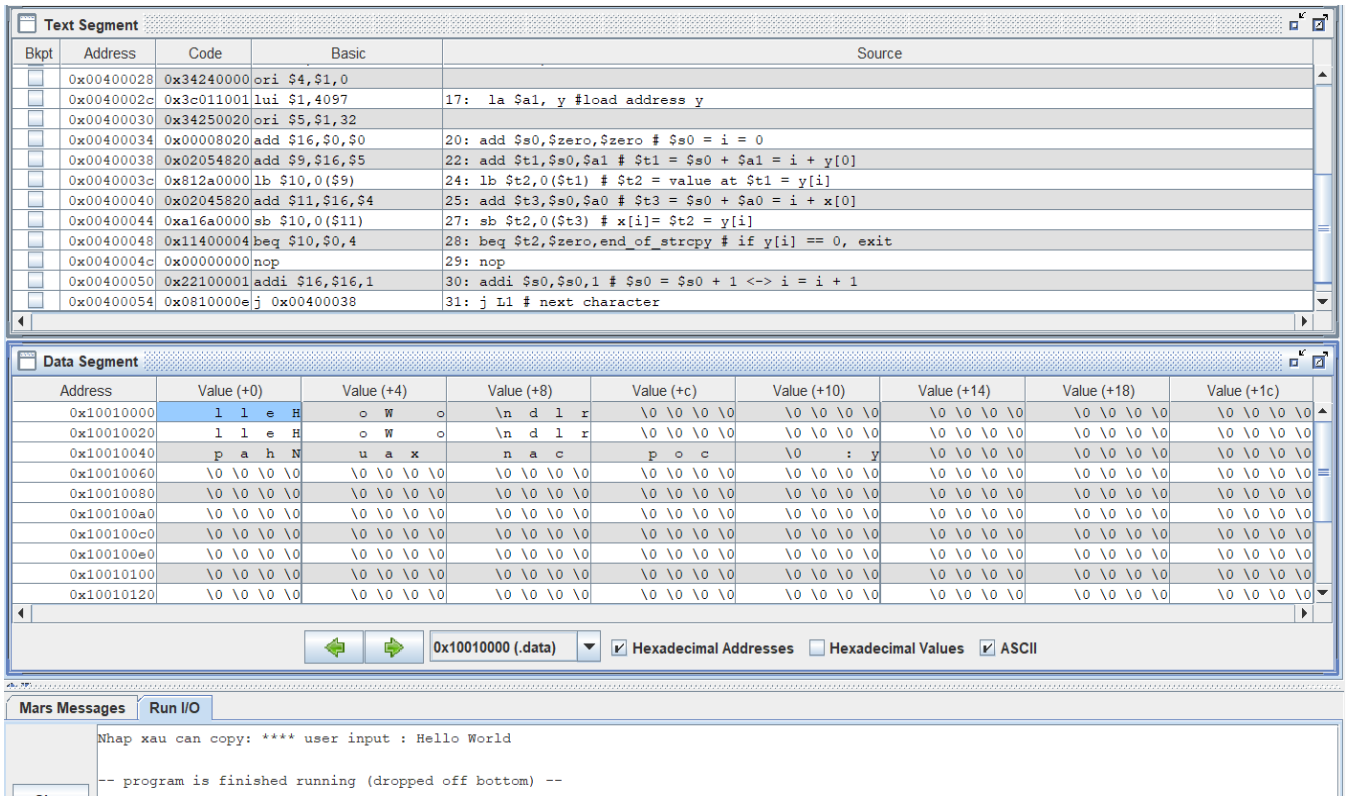
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	l l e H	o W o	\n d l r	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	p a h N	u a x	n a c	p o c	\0 : y	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☑ ASCII

Mars Messages | Run I/O

```
Nhap xau can copy: **** user input : Hello World
```

- Bộ nhớ lúc sau khi kết thúc chương trình:



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400028	0x34240000	ori $4,$1,0	
	0x0040002c	0x3c011001	lui $1,4097	17: la $a1, y #load address y
	0x00400030	0x34250020	ori $5,$1,32	
	0x00400034	0x00008020	add $16,$0,$0	20: add $s0,$zero,$zero # $s0 = i = 0
	0x00400038	0x02054820	add $9,$16,$5	22: add $t1,$s0,$a1 # $t1 = $s0 + $a1 = i + y[0]
	0x0040003c	0x812a0000	lb $10,0($9)	24: lb $t2,0($t1) # $t2 = value at $t1 = y[i]
	0x00400040	0x02045820	add $11,$16,$4	25: add $t3,$s0,$a0 # $t3 = $s0 + $a0 = i + x[0]
	0x00400044	0xa16a0000	sb $10,0($11)	27: sb $t2,0($t3) # x[i]= $t2 = y[i]
	0x00400048	0x11400004	beq $10,$0,4	28: beq $t2,$zero,end_of_strcpy # if y[i] == 0, exit
	0x0040004c	0x00000000	nop	29: nop
	0x00400050	0x22100001	addi $16,$16,1	30: addi $s0,$s0,1 # $s0 = $s0 + 1 <-> i = i + 1
	0x00400054	0x0810000e	j 0x00400038	31: j L1 # next character

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	l l e H	o W o	\n d l r	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	l l e H	o W o	\n d l r	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	p a h N	u a x	n a c	p o c	\0 : y	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☑ ASCII

Mars Messages | Run I/O

```
Nhap xau can copy: **** user input : Hello World

-- program is finished running (dropped off bottom) --
```

Assignment 4

```
#Laboratory Exercise 5, Home Assignment 3
.data
string: .space 50
Message1: .asciiz "Nhap xau: "
Message2: .asciiz "Do dai xau la: "
.text
main:
get_string:       li $v0,4
                  la $a0,Message1
                  syscall

                  li $v0, 8    #read string
                  la $a0, string
                  li $a1,50
                  syscall

get_length:       la $a0,string # $a0 = address(string[0])
                  add $t0,$zero,$zero # $t0 = i = 0
check_char:     add $t1,$a0,$t0 # $t1 = $a0 + $t0
                  # = address(string[i])

                  lb $t2, 0($t1) # $t2 = string[i]
                  beq $t2, $zero, end_of_str # is null char?
                  addi $t0, $t0, 1 # $t0 = $t0 + 1 -> i = i + 1
```

```
 j check_char
end_of_str:
end_of_get_length:

print_length:    li $v0,56   # Intenger Message Dialog
                 la $a0,Message2
                 add $a1,$zero,$t0
                 syscall
```

Assignment 5

```
.data
string: .space 21
.text
la $a0, string # a0 chua dia chi cua string
li $s0, -1 # s0 la i
loop:
add $s0, $s0, 1 #s0 = s0 + 1
addi $s3, $s0, -19 #s3 = s1 -20 = count - 20
beq $s3, $zero, endloop#neu so luong qua 20, thoat
nop
li $v0, 12 #doc vao v0
```

```
syscall
add $s4, $v0, -10 # Kiem tra v0 co phai Enter?
beq $s4, $zero, endloop# Neu co thoat
add $s1, $s0, $a0 #s1 = dia chi cua a[i]
sb $v0, 0($s1) #Luu vao a[i]
j loop
nop
endloop:
print_loop:
li $v0, 11 # Ham 11 in character
la $a2, string
lb $a0, 0($s1) # In s1
syscall
beq $s1, $a2, end_print_loop # Neu den a[0], thoat
nop
addi $s1, $s1, -1 # s1= s1-1
j print_loop
nop
end_print_loop:
```