

# Laboratory Exercise 5

## Character string with SYSCALL function, and sorting

Đỗ Hải Dương – 20194528

### Assignment 1

```
1 .data
2 test: .asciiz "Hello World, I'm Do Hai Duong 20194528"
3 .text
4     li $v0, 4
5     la $a0, test
6     syscall
```

Kết quả:

The screenshot shows a MIPS simulator interface. At the top, there are 'Edit' and 'Execute' tabs. Below them is the 'Text Segment' window, which contains a table of instructions with columns for Bkpt, Address, Code, Basic, and Source. The instructions are:

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x24020004	addiu \$2,\$0,4	4: li \$v0, 4
<input type="checkbox"/>	0x00400004	0x3c011001	lui \$1,4097	5: la \$a0, test
<input type="checkbox"/>	0x00400008	0x34240000	ori \$4,\$1,0	
<input type="checkbox"/>	0x0040000c	0x0000000c	syscall	6: syscall

Below the 'Text Segment' window is the 'Data Segment' window, which shows a table of memory addresses and their values. The addresses range from 0x10010000 to 0x100100e0. The values are mostly 0, except for the first few which are 1819043144, 892614961, and 14386.

At the bottom of the simulator is a console window. It shows the output of the program: 'Hello World, I'm Do Hai Duong 20194528'. The console also shows the program starting and finishing running.

Nhận xét:

- 4 (4 là option in chuỗi ra I/O) được lưu vào thanh ghi \$v0, thanh ghi lưu các service number
- Sau đó giá trị test string được load vào \$a0

- Ở Data Segment, mỗi ô nhớ lưu 4 kí tự của test string (4 bytes = 32 bit), mỗi kí tự được lưu trong 1 byte nhớ.

## Assignment 2

```
1  .data
2  string1:      .asciiiz "The sum of "
3  string2:      .asciiiz " and "
4  string3:      .asciiiz " is "
5  .text
6              li $s0, 2019
7              li $s1, 4528
8  # Print "The sum of "
9              li $v0, 4
10             la $a0, string1
11             syscall
12 # Print $s0 value
13             li $v0, 1
14             add $a0, $zero, $s0
15             syscall
16 # Print " and "
17             li $v0, 4
18             la $a0, string2
19             syscall
20 # Print $s1 value
21             li $v0, 1
22             add $a0, $zero, $s1
23             syscall
24 # Print " is "
25             li $v0, 4
26             la $a0, string3
27             syscall
28 # Print sum of $s0 and $s1
29             li $v0, 1
30             add $a0, $s0, $s1
31             syscall
32
```

## Kết quả:

Text Segment				
Bkpt	Address	Code	Basic	Source
	0x00400000	0x241007e3	addiu \$t6,\$0,2019	6: li \$s0, 2019
	0x00400004	0x241111b0	addiu \$t7,\$0,4528	7: li \$s1, 4528
	0x00400008	0x24020004	addiu \$t2,\$0,4	9: li \$v0, 4
	0x0040000c	0x3c011001	lui \$t1,4097	10: la \$a0, string1
	0x00400010	0x34240000	ori \$t4,\$t1,0	
	0x00400014	0x0000000c	syscall	11: syscall
	0x00400018	0x24020001	addiu \$t2,\$0,1	13: li \$v0, 1
	0x0040001c	0x00102020	add \$t4,\$0,\$t6	14: add \$a0, \$zero, \$s0
	0x00400020	0x0000000c	syscall	15: syscall
	0x00400024	0x24020004	addiu \$t2,\$0,4	17: li \$v0, 4

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	543516756	544044403	2123375	1684955424	1763704864	8307	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

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

The sum of 2019 and 4528 is 6547

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

Giải thích : Lưu giá trị s0 , s1

Gán v0 = 4 để in ra string1

V0=1 để in ra value s0(là số nguyên )

Tương tự như thế tới hết chương trình

Kết quả 2019 + 4528 = 6547

## Assignment 3

```

1  .data
2  x: .space 1000                # destination string x, empty
3  y: .ascii "Hello, I'm Do Hai Duong 20194528" # source string y
4
5  .text
6      la $a0, x                # assign string x address to a0
7      la $a1, y                # assign string y address to a1
8  strcpy:
9      add    $s0, $zero, $zero    #s0 = i = 0
10 L1:
11      add    $t1, $s0, $a1        #t1 = s0 + a1 = i + y[0]
12                                # = address of y[i]
13      lb     $t2, 0($t1)          #t2 = value at t1 = y[i]
14      add    $t3, $s0, $a0        #t3 = s0 + a0 = i + x[0]
15                                # = address of x[i]
16      sb     $t2, 0($t3)          #x[i]= t2 = y[i]
17      beq    $t2, $zero, end_of_strcpy #if y[i]==0, exit
18      nop
19
19      addi    $s0, $s0, 1          #s0=s0 + 1 <-> i=i+1
20      j       L1                  #next character
21      nop
22 end_of_strcpy:
23 # print value of x after strcpy
24      li $v0, 4
25      la $a0, x
26      syscall

```

Kết quả:

Text Segment				
Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,4097	6: la \$a0, x # assign string x address to a0
	0x00400004	0x34240000	ori \$4,\$1,0	
	0x00400008	0x3c011001	lui \$1,4097	7: la \$a1, y # assign string y address to a1
	0x0040000c	0x342503e8	ori \$5,\$1,1000	
	0x00400010	0x00008020	add \$16,\$0,\$0	9: add \$s0,\$zero,\$zero #s0 = i = 0
	0x00400014	0x02054820	add \$9,\$16,\$5	11: add \$t1,\$s0,\$a1 #t1 = s0 + a1 = i + y[0]
	0x00400018	0x812a0000	lb \$10,0(\$9)	13: lb \$t2,0(\$t1) #t2 = value at t1 = y[i]
	0x0040001c	0x02045820	add \$11,\$16,\$4	14: add \$t3,\$s0,\$a0 #t3 = s0 + a0 = i + x[0]
	0x00400020	0x16a00000	sb \$10,0(\$11)	16: sb \$t2,0(\$t3) #x[i]= t2 = y[i]
	0x00400024	0x11400004	beq \$10,\$0,4	17: beq \$t2,\$zero,end_of_strcpy #if y[i]==0, exit

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1819043144	1226845295	1142975783	1632116847	1967399017	543649391	959524914	942814516
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

```

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

Hello, I'm Do Hai Duong 20194528

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

```

Giải thích : Gán chuỗi y : "Hello, I'm Do Hai Duong 20194528"

A1 lưu địa chỉ của chuỗi y , a0 lưu địa chỉ chuỗi x

Gán địa chỉ của y[i] cho t1, gán giá trị của t1 cho t2

Gán địa chỉ của x cho t3 ,lưu giá trị vào mảng x

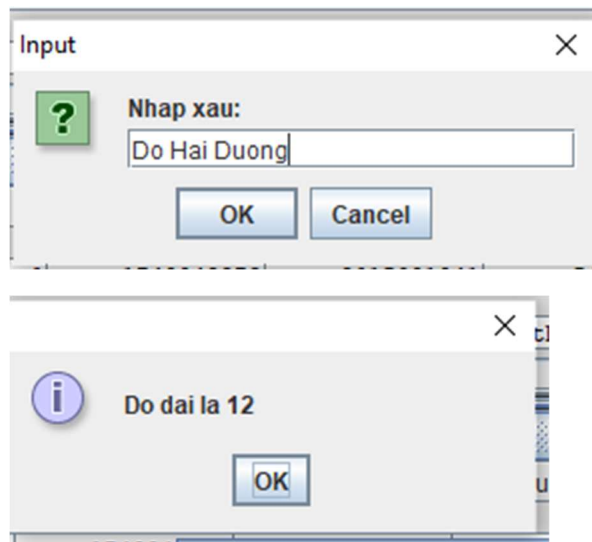
Nếu đọc hết chuỗi thì đến end\_of\_strcpy , không thì tăng biến đếm lên 1 , dùng lệnh j quay lại L1

Đưa địa chỉ x vào thanh ghi a0 và in ra bằng cách gán v0 =4

## Assignment 4

```
1  .data
2  string:          .space 50
3  Message1:        .asciiz "Nhap xau: "
4  Message2:        .asciiz "Do dai la "
5  .text
6  main:
7  get_string:
8                  li $v0, 54                # load option InputDialogString
9                  la $a0, Message1          # load message address
10                 la $a1, string            # load input string address
11                 li $a2, 50                # Maximum length
12                 syscall
13
14  get_length:      la $a0, string           # a0 = Address(string[0])
15                 xor $v0, $zero, $zero     # v0 = length = 0
16                 xor $t0, $zero, $zero     # t0 = i = 0
17
18  check_char:      add $t1, $a0, $t0        # t1 = a0 + t0
19
20                 #= Address(string[0]+i)
21                 lb $t2, 0($t1)            # t2 = string[i]
22                 beq $t2, $zero, end_of_str # Is null char?
23                 addi $v0, $v0, 1          # v0=v0+1->length=length+1
24                 addi $t0, $t0, 1          # t0=t0+1->i = i + 1
25                 j check_char
26  end_of_str:
27  end_of_get_length:
28  print_length:    add $s0, $v0, -1         # s0 store the result -1
29                 # (because InputDialogString automatically add a n character to the end)
30                 li $v0, 56                # option MessageDialogInt
31                 la $a0, Message2          # load message address
32                 add $a1, $s0, $zero        # load the length to be printed
33                 syscall                   # execute
```

**Kết quả:**



Giải thích : Gán v0 =54 để hiển thị hộp thoại nhập chuỗi kí tự

Đưa địa chỉ biến vào thanh ghi a0,v1 lưu độ dài xâu

Khởi tạo i=0. Gán t1 = địa chỉ string[i]

Lưu giá trị t1 vào t3, nếu t3=0(đọc hết ) thì nhảy tới end\_get\_length

Nếu chưa thì tăng biến đếm lên 1 , nhảy lại nhãn check\_char

Cuối cùng đưa Message2 vào a0 , giảm v1 đi 1 , lấy giá trị gán vào a0 và đưa ra màn hình

## Assignment 5

```

1  .data
2  Message:      .ascii "Nhap xau (khong qua 20 ki tu): "
3  string:       .space 20
4  .text
5              li      $v0, 4
6              la      $a0, Message
7              li      $t0, 20          # Maximum length
8              syscall
9              la      $s0, string      # s0 = address of string
10             li      $s1, 0          # s1 = i = 0
11
12 read_str:     li      $v0, 12
13             syscall
14
15 check_char:
16             bge      $s1, $t0, print_reverse # branch if s1 >= t0 <=> i >= 20
17             beq      $v0, 10, print_reverse # branch if input is enter
18             add      $s2, $s0, $s1      # s2 = s0 + s1 = a[0] + i = add of s[i]

```



```

19         sb      $v0, 0($s2)          #store input char to s[i]
20         add     $s1, $s1, 1          # i = i + 1
21         j       read_str
22
23 print_reverse:
24         blt     $s1, $zero, exit      # exit if end of string
25         add     $s2, $s0, $s1        # s2 = add of s[i]
26         lb      $t2, 0($s2)          # t2 = char s[i]
27         li      $v0, 11              # print char
28         add     $a0, $t2, $zero
29         syscall
30         addi    $s1, $s1, -1
31         j       print_reverse
32 exit:

```

## Kết quả:

```

Nhap xau (khong qua 20 ki tu): Do Hai Duong
gnouD iaH oD
-- program is finished running (dropped off bottom) --

```

Nhập quá 20 kí tự:

```

Nhap xau (khong qua 20 ki tu): Do Hai Duong 201945282549102 gnouD iaH oD
-- program is finished running (dropped off bottom) --

```

(Tự kết thúc và in ngược lại luôn)

Giải thích : Gán v0 cho 4 để hiện thị message

Gán địa chỉ của string cho s0 ,gán l = 0

Đọc kí tự bằng cách ván v =12

Nếu i >=20 hoặc v=10 thì nhảy tới print\_reverse

Gán s2 bằng địa chỉ string[i]

Lưu từng kí tự của chuỗi vào v0

Tăng biến l lên 1 đơn vị sau mỗi lần lặp

Vòng lặp in ngược lại : nếu i =0 => thoát

Lưu s2 bằng đ/c của string[i],gán từng giá trị của s2 vào t2 , In t2 ra màn hình bằng cách gán v0 =11

Giảm l đi 1 đơn vị và lặp lại cho đến khi l =0 thì dừng