

Nguyễn Trọng Tuệ 20194710

Bài thực hành tuần 5

1. Assignment 1

```
1 #Laboratory Exercise 5, Home Assignment 1
2 .data
3 test: .asciiz "Hello World"
4 .text
5 li $v0, 4
6 la $a0, test
7 syscall
```

- Khi thực hiện chương trình:

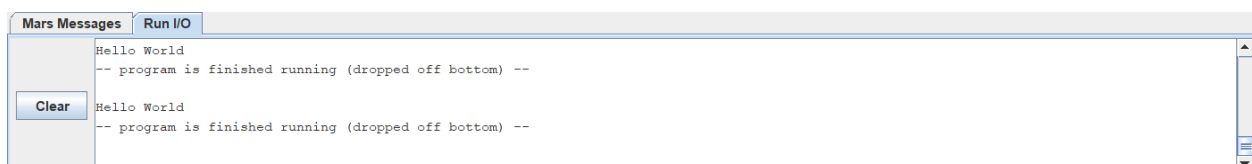
- Đầu tiên, giá trị của thanh ghi chứa giá trị trả về của thủ tục \$v0 được gán giá trị = 4

\$at	1	0x00000000
\$v0	2	0x00000004
\$v1	3	0x00000000

- Giá trị địa chỉ của biến test (chuỗi ký tự “Hello World”) được lưu vào trong thanh ghi \$a0

assignment1.asm		
test		0x10010000
\$at	1	0x10010000
\$v0	2	0x00000004
\$v1	3	0x00000000
\$a0	4	0x10010000

- Cuối cùng, chương trình thực hiện lệnh syscall, giá trị trả của thủ tục ở thanh ghi \$v0 lúc này được lấy ra là 4 => chương trình thực hiện thủ tục print String với chuỗi kết thúc bằng ký tự null được in ra có địa chỉ là 0x10010000 đã được lưu vào thanh ghi tham số \$a0 (thanh ghi tham số đầu vào của thủ tục). Kết quả của chương trình như sau:



2. Assignment 2

```
assignment1.asm  assignment2.asm
1  #Laboratory Exercise 5, Home Assignment 2
2  .data
3      mess1: .asciiz  "The sum of "
4      mess2: .asciiz  " and "
5      mess3: .asciiz  " is "
6  .text
7      addi $s0, $s0, 5
8      addi $s1, $0, 4
9      add  $s2, $s0, $s1
10     li   $v0, 4
11     la   $a0, mess1
12     syscall
13     li   $v0, 1
14     la   $a0, 0($s0)
15     syscall
16     li   $v0, 4
17     la   $a0, mess2
18     syscall
19     li   $v0, 1
20     la   $a0, 0($s1)
21     syscall
22     li   $v0, 4
23     la   $a0, mess3
24     syscall
25     li   $v0, 1
26     la   $a0, 0($s2)
27     syscall
```

- Kết quả:

```
Mars Messages  Run I/O
Clear
The sum of 5 and 4 is 9
-- program is finished running (dropped off bottom) --
```

3. Assignment 3

```

1 #Laboratory Exercise 5, Assignment 3
2 .data
3 x: .space 1000           # destination string x, empty
4 y: .asciiz "Hello"       # source string y
5 .text
6 strcpy:
7     la $a0, x             # a0 = destination string x
8     la $a1, y             # a1 = source string y
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     addi $s0, $s0, 1      #s0=s0 + 1 <-> i=i+1
20     j L1                  #next character
21     nop
22 end_of_strcpy:

```

- Kết quả:

Address	Value (+0)	Value (+4)
0x10010000	1 1 e H	\0 \0 \0 o

- Giải thích cho kết quả thu được:

- Chương trình MIPS trên tương đương với đoạn mã C sau:

```

void strcpy (char x[], char y[]) {
    int i = 0;
    while ((x[i] = y[i] != '\0')    i += 1;
}

```

- Hai thanh ghi \$a0 và \$a1 chứa lần lượt giá trị của xâu đích x và xâu nguồn “Hello” y.

x	0x10010000	
y	0x100103e8	
\$a0	4	0x10010000
\$a1	5	0x100103e8

- Thanh ghi \$s0 chứa giá trị chỉ số (index) [i] , sau mỗi lần sao chép y[i] sang x[i], thanh ghi \$s0 này sẽ tự động cộng thêm 1 đơn vị (i += 1)

- \$t1 và \$t3 lần lượt chứa giá trị địa chỉ của x[i] và y[i] được lấy vào thông qua lệnh lb và sb (đọc/ lưu giá trị của từng byte với offset 0 + địa chỉ y[i] hay x[i]). Thanh ghi \$t2 được sử dụng làm cầu nối chứa giá trị của y[i], sau đó lưu giá trị này ra địa chỉ x[i] trên thanh ghi \$t3. Ví dụ với chỉ số i = 1:

Address	Value (+0)
0x10010000	\0 \0 e H

\$t1	9	0x100103e9
\$t2	10	0x00000065
\$t3	11	0x10010001
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000001

- Ở đây, i = 1 => \$s0 = 0x00000001
- x[1] có địa chỉ = x + 1 hay x[0] + 1 = 0x10010000 + 1 = 0x10010001
- y[1] có địa chỉ = y + 1 hay y[0] + 1 = 0x100103e8 + 1 = 0x100103e9
- Giá trị được đọc qua lệnh lb vào thanh ghi \$t2 là 0x65 ⇔ 101 = 'e' theo bảng mã ASCII
- Sau đó, thông qua lệnh sb tại offset 0 của địa chỉ x[1] đang được lưu trữ trong thanh ghi \$t3, x[1] tại 0x10010001 được gán giá trị 'e' như trên hình
- Chương trình tiếp tục thực hiện như trên, kiểm tra điều kiện giá trị được lưu vào ở \$t2 có bằng 0 ⇔ '\0' hay không? Nếu không thì \$s0 += 1, gán '\0' tạm thời sau giá trị vừa đọc được và nhảy đến nhãn L1 để tiếp tục, còn nếu xảy ra \$t2 = 0 nghĩa là đã đọc đến ký tự kết thúc null '\0' và nhảy đến nhãn end_of_strcpy để kết thúc chương trình.

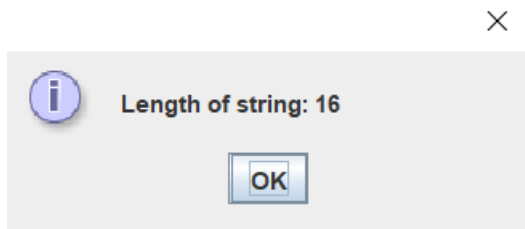
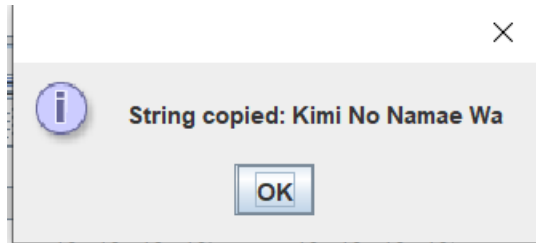
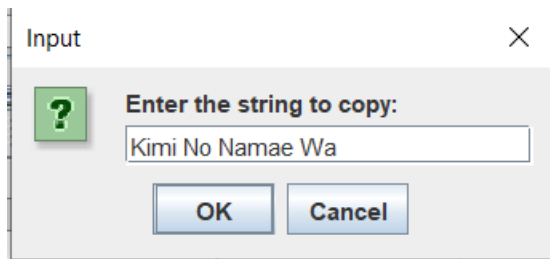
4. Assignment 4

```

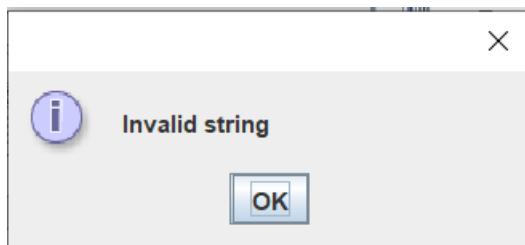
1  #Laboratory Exercise 5, Assignment 4
2  .data
3  x:  .space 100           # destination string x, empty
4  y:  .space 100           # source string y
5  mes_in: .asciiz "Enter the string to copy: "
6  mes_out: .asciiz "String copied: "
7  mes_lgth: .asciiz "Length of string: "
8  mes_err: .asciiz "Invalid string"
9  .text
10  li $v0, 54
11  la $a0, mes_in
12  la $a1, y
13  la $a2, 100
14  syscall
15  beq $a1, $0, strcpy      #check status value if != 0 (OK)
16  li $v0, 59
17  la $a0, mes_err
18  syscall
19  j exit
20 strcpy:
21  la $a0, x                # a0 = destination string x
22  la $a1, y                # a1 = source string y
23  add $s0, $zero, $zero    #s0 = i=0
24 L1:
25  add $t1, $s0, $a1        #t1 = s0 + a1 = i + y[0]
26                                # = address of y[i]
27  lb $t2, 0($t1)           #t2 = value at t1 = y[i]
28  add $t3, $s0, $a0        #t3 = s0 + a0 = i + x[0]
29                                # = address of x[i]
30  sb $t2, 0($t3)           #x[i]= t2 = y[i]
31  beq $t2, $zero, end_of_strcpy #if y[i]=0, exit
32  nop
33  addi $s0, $s0, 1         #s0=s0 + 1 <-> i=i+1
34  j L1                    #next character
35  nop
36 end_of_strcpy:
37  li $v0, 59
38  la $a1, 0($a0)
39  la $a0, mes_out
40  syscall
41
42  li $v0, 56
43  la $a0, mes_lgth
44  la $a1, 0($s0)
45  subi $a1, $a1, 1         #length of x without null character at the end of the string
46  syscall
47 exit:

```

- Kết quả khi nhập xâu hợp lệ:



- Ngoại lệ (ấn X hoặc Cancel):



5. Assignment 5

```

1  #Laboratory Exercise 5, Assignment 5
2  .data
3  mes:      .asciiz  "Enter a character (press Enter to end): "
4  str:      .space 20
5  .text
6  init:
7      add $s0, $zero, $zero
8      la  $s1, str
9  read:
10     li   $v0, 12          # input character
11     la   $a0, mes
12     syscall
13     nop
14  check:
15     beq $v0, 10, print    # terminate execution -> jump to print
16     add $t1, $s0, $s1     # load str[i] -> t1
17     sb   $v0, 0($t1)      # load $v0 -> $t1 since $v0 contains character read
18     addi $s0, $s0, 1      # $s0++
19     slti $t0, $s0, 20     # if $s0 == 20: jump to print
20     beq $t0, $zero, print
21     j    read
22  print:
23     slt  $t0, $s0, $zero   # for $s0 -> 0
24     bne $t0, $zero, exit
25     add $t1, $s0, $s1     # $t1 = address of string[$s0]
26     lb   $t2, 0($t1)      # $t2 = string[$s0]
27     li   $v0, 11
28     add $a0, $zero, $t2   # print $t2
29     syscall
30     addi $s0, $s0, -1     # i -= 1
31     j    print
32  exit:

```

- Kết quả:

Mars Messages	Run I/O
<div>Clear</div>	<pre> Nguyen Trong Tue euT gnorT neyugN -- program is finished running (dropped off bottom) -- </pre>