

Assignment 1

Mã nguồn:

```
#Laboratory Exercise 5, Assignment 1
.data
test: .asciiz "Hoang Quoc Bao"
.text
li $v0, 4
la $a0, test
syscall
```

Giải thích:

Thanh ghi \$v0 có giá trị = 4, tức là hệ thống sẽ thực hiện chức năng "print string" khi dùng lệnh syscall. Thanh ghi \$a0 chứa địa chỉ của chuỗi *test*, vậy hệ thống sẽ in ra màn hình giá trị của chuỗi *test*.

Trong bộ nhớ, chuỗi *test* được lưu như sau:

Value (+0)	Value (+4)	Value (+8)	Value (+c)
n a o H	u Q g	B c o	\0 \0 o a

Do 1 char = 1 byte = 8 bit. Vậy mỗi thanh ghi 32 bit sẽ lưu được $32/8 = 4$ char, các char được lưu từ phải qua trái. (bit thấp đến cao)

Kết quả thực hiện:

```
Hoang Quoc Bao
-- program is finished running (dropped off bottom) --
```

Assignment 2

Mã nguồn:

```
#Laboratory Exercise 5, Assignment 2
.data
Message1: .asciiz "The sum of ("
Message2: .asciiz ") and ("
Message3: .asciiz ") is ("
Message4: .asciiz ")"
.text
addi $s0, $zero, 0x01341330 # $s0 = 20190000
addi $s1, $zero, 0x1184 # $s1 = 4484
add $s2, $s1, $s0 # $s2 = $s1 + $s0
addi $v0, $zero, 4 # print string
la $a0, Message1
syscall

addi $v0, $zero, 1 # print integer
add $a0, $zero, $s0 # $a0 = $s0
syscall

addi $v0, $zero, 4 # print string
la $a0, Message2
syscall

addi $v0, $zero, 1 # print integer
add $a0, $zero, $s1 # $a0 = $s1
syscall

addi $v0, $zero, 4 # print string
la $a0, Message3
syscall

addi $v0, $zero, 1 # print integer
add $a0, $zero, $s2 # $a0 = $s2
syscall

addi $v0, $zero, 4 # print string
la $a0, Message4
syscall
```

Giải thích:

Ta chia chuỗi lớn The sum of (s0) and (s1) is (result) thành 7 chuỗi nhỏ: The sum of (+ s0 +) and (+ s1 +) is (+ result +). Chạy từng lệnh syscall để in ra từng chuỗi nhỏ trên.

Kết quả thực hiện:

```
The sum of (20190000) and (4484) is (20194484)
-- program is finished running (dropped off bottom) --
```

Assignment 3

Mã nguồn:

```
#Laboratory Exercise 5, Home Assignment 2
.data
x: .space 1000 # destination string x, empty
y: .asciiz "Hoang Quoc Bao\n" # source string y

.text
strcpy:
add $s0,$zero,$zero #s0 = i=0
L1:
la $a1, y # $a1 = address of y[0]
la $a0, x # $a0 = address of x[0]
add $t1,$s0,$a1 #t1 = s0 + a1 = i + y[0]
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:
# Printf x and y
li $v0, 4
la $a0, x
syscall
la $a0, y
syscall
```

Giải thích:

Ta duyệt từng phần tử của chuỗi y bằng lệnh *lb* và lưu vào mảng x bằng lệnh *sb*. Chương trình kết thúc khi giá trị $y[i] == 0$ tức là ta đã duyệt đến cuối mảng.

Kết quả thực hiện:

```
Hoang Quoc Bao
Hoang Quoc Bao
-- program is finished running (dropped off bottom) --
```

Assignment 4

Mã nguồn:

```
#Laboratory Exercise 5, Home Assignment 3
.data
string: .space 50
Message1: .ascii "Nhap xau:"
Message2: .ascii "Do dai la: "
.text
main:
get_string:
addi $v0, $zero, 8
la $a0, string
li $a1, 100
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:
print_length:
add $s0, $zero, $v0
addi $v0, $zero, 4
la $a0, Message2
syscall

addi $v0, $zero, 1
add $a0, $s0, $zero
syscall
```

Giải thích:

Lần lượt duyệt các phần tử trong mảng rồi tăng biến đếm lên 1. Chương trình kết thúc khi `string[i] == 0`. (Chú ý: không nên dùng thanh ghi `$v0` để lưu giá trị của biến như trong bài)

Kết quả thực hiện:

```
Hoang Quoc Bao
Do dai la: 15
-- program is finished running (dropped off bottom) --
```

Assignment 5

Mã nguồn:

```
#Laboratory Exercise 5, Assignment 5
.data
string: .space 100
Message1: .ascii "Nhap xau: "
Message2: .ascii "Xau dao nguoc: "
.text
addi $v0, $zero, 4 # printf Message1
la $a0, Message1
syscall

addi $v0, $zero, 8
la $a0, string
li $a1, 20 # set max length = 20
syscall

addi $v0, $zero, 4 # printf Message2
la $a0, Message2
syscall

print_inverse_string:
la $t0, string # $t0 = string[0]
find_strlen:
add $s1, $t0, $t1 # $t1 = i, $s1 = address of string[i]
lb $s0, 0($s1) # s0 = string[i]
beq $s0, $zero, traverse_inverse # branch if i == strlen(string)
addi $t1, $t1, 1 # i++
j find_strlen

traverse_inverse:
addi $v0, $zero, 11 #print char
print_char:
addi $t1, $t1, -1 # i--
add $s1, $t0, $t1 # s1 = address of string[i]
lb $a0, 0($s1) # $a0 = string[i]
syscall
beq $t1, $zero, done # if i == 0 then done
j print_char
done:
```

Giải thích:

Duyệt mảng theo chiều xuôi để lấy giá trị độ dài của xâu. Sau đó duyệt ngược lại để in ra các chữ theo giá trị đảo ngược.

Kết quả thực hiện:

```
Nhap xau: 20194484
Xau dao nguoc:
48449102
-- program is finished running (dropped off bottom) --
```