

Báo cáo thực hành Kỹ thuật máy tính

Assignment 1:

#Laboratory Exercise 5, Home Assignment 1

.data

test: **.ascii**z "Hello World"

.text

li \$v0, 4

la \$a0, test

syscall

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1 1 e H	o W o	\0 d 1 z	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\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
0x10010040	\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
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 MessagesRun I/O

Clear

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

Assignment 2

.data

```
mess1: .asciiz    "The sum of "  
mess2: .asciiz    " and "  
mess3: .asciiz    " is "
```

.text

```
li    $s0, 30      # s0 = 20  
li    $s1, 20      # s1 = 30  
add   $s2, $s0, $s1 # s2 = s0 + s1  
  
li    $v0, 4  
la    $a0, mess1    #print "The sum of "  
syscall  
  
li    $v0, 1  
add   $a0, $zero, $s1 # $a0 = $s1  
syscall  
  
li    $v0, 4  
la    $a0, mess2    #print "and "  
syscall  
|  
li    $v0, 1  
add   $a0, $zero, $s2 # $a0 = $s2  
syscall  
  
li    $v0, 4  
la    $a0, mess3    #print "is "  
syscall  
  
li    $v0, 1  
add   $a0, $s1, $s2 # a0 = s1 + s2  
syscall
```

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	e h T	m u s	\0 f o	d n a	i \0	\0 \0 s	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\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
0x10010040	\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
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

.data

mess1: .asciiz "The sum of "

mess2: .asciiz " and "

mess3: .asciiz " is "

.text

li \$s0, 30 # s0 = 20

li \$s1, 20 # s1 = 30

add \$s2, \$s0, \$s1 # s2 = s0 + s1

li \$v0, 4

la \$a0, mess1 #print "The sum of "

syscall

li \$v0, 1

add \$a0, \$zero, \$s1 # \$a0 = \$s1

syscall

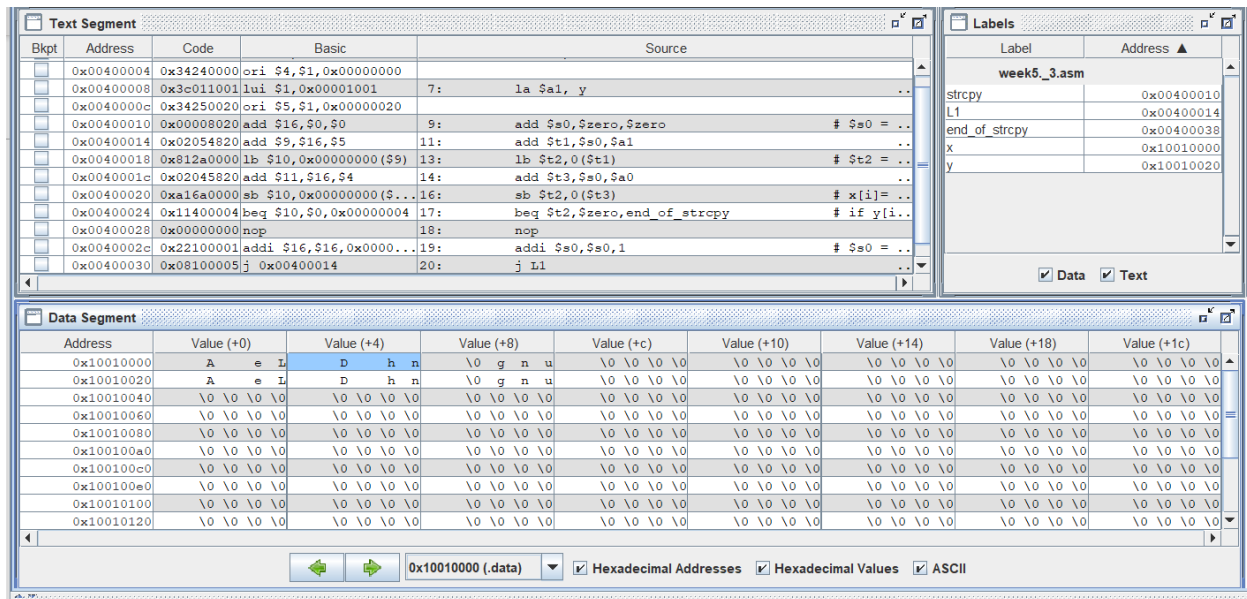
li \$v0, 4

la \$a0, mess2 #print "and "

syscall

The sum of 20 and 50 is 70
-- program is finished running (dropped off bottom) --

Clear



#Laboratory Exercise 5, Home Assignment 2

.data

x: `.space 32` **# destination string x, empty**

y: `.ascii "Le Anh Dung"` **# source string y**

.text

`la $a0, x` **#a0 = address x**

`la $a1, y` **#a1 = address y**

strcpy:

`add $s0,$zero,$zero` **# \$s0 = i = 0**

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:

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:

TODO

li \$v0, 54

la \$a0, Message1

Message1 = "Nhap xau"

la \$a1, string

input String

la \$a2, 100

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

lb \$t2, 0(\$t1)

= address(string[i])

beq \$t2, \$zero, end_of_str

\$t2 = string[i]

addi \$t0, \$t0, 1

is null char?

j check_char

\$t0 = \$t0 + 1 -> i = i + 1

end_of_str:

end_of_get_length:

addi \$a1, \$t0, -1

a1 = length

print_length:

TODO

li \$v0, 56

la \$a0, Message2

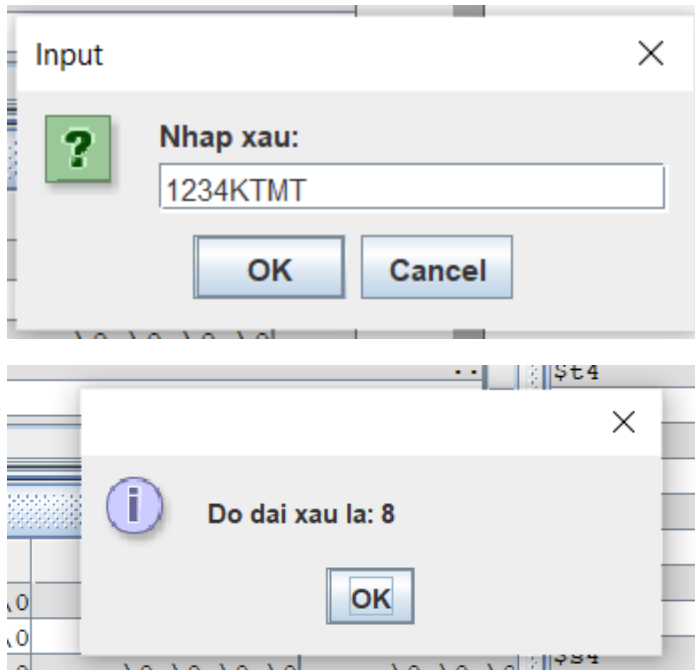
#Message2 = "Do dai xau la:"

add \$a1, \$a1, \$zero

the interger to be printed

syscall

execute



#Laboratory Exercise 5, Home Assignment 3

.data

string: .space 50

Message1: .ascii "Nhap xau: "

Message2: .ascii "Do dai xau la: "

.text

main:

get_string: # TODO

li \$v0, 54

la \$a0, Message1 # Message1 = "Nhap xau"

la \$a1, string # input String

la \$a2, 100

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:	
addi \$a1, \$t0, -1	# a1 = length
print_length:	# TODO
li \$v0, 56	
la \$a0, Message2	#Message2 = "Do dai xau la:"
add \$a1, \$a1, \$zero	# the interger to be printed
syscall	# execute

Assignment 5

```
.data
    Message: .asciiz "\nChuoi nguoc: "
    buffer: .space 50
.text
    la $s0, buffer           # Load address buffer
    li $s1, 0                # index = 0
    li $s2, 0                # i = 0
    li $t1, 20               # Max length
    li $t2, 10               # ASCII code '\n'
start_read_char:
    li $v0, 12
    syscall
    add $s1, $s0, $s2        # Load new address buffer

    addi $s2, $s2, 1         # i++
    sb $v0, 0($s1)           # save byte v0 to address s1
    beq $s2, $t1, end_read_char # exit when max length
    beq $v0, $t2, end_read_char # exit when enter

    j start_read_char

end_read_char:
    li $v0, 4
    la $a0, Message          # Message: "\nChuoi nguoc: "
    syscall
Reverse_string:
    li $v0, 11               # Print character
    lb $a0, 0($s1)
    syscall

    beq $s1, $s0, exit
    addi $s1, $s1, -1
    j Reverse_string
exit:
```

Khi enter:

```
abcdef
Chuoi nguoc:
fedcba
```

Khi nhập quá 12 kí tự:

```
123456789abcdefghijkl
Chuoi nguoc: kjihgfedcba987654321
```

.data

Message: .asciiz "\nChuoi nguoc: "

buffer: .space 50

.text

la \$s0, buffer # Load address buffer

li \$s1, 0 # index = 0

li \$s2, 0 # i = 0

li \$t1, 20 # Max length

li \$t2, 10 #ASCII code '\n'

start_read_char:

li \$v0, 12

syscall

add \$s1, \$s0, \$s2 # Load new address buffe

addi \$s2, \$s2, 1 # i++

sb \$v0, 0(\$s1) # save byte v0 to address s1

beq \$s2, \$t1, end_read_char # exit when max length

beq \$v0, \$t2, end_read_char # exit when enter

j start_read_char

end_read_char:

li \$v0, 4

la \$a0, Message # Message: "\nChuoi nguoc: "

syscall

Reverse_string:

li \$v0, 11 # Print charecter

lb \$a0, 0(\$s1)

syscall

beq \$s1, \$s0, exit

```
addi    $s1, $s1, -1
```

```
j Reverse_string:
```

```
exit:
```