

BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 5

Họ và tên: Hoàng Văn Thắng

MSSV: 20235828

Assignment 1

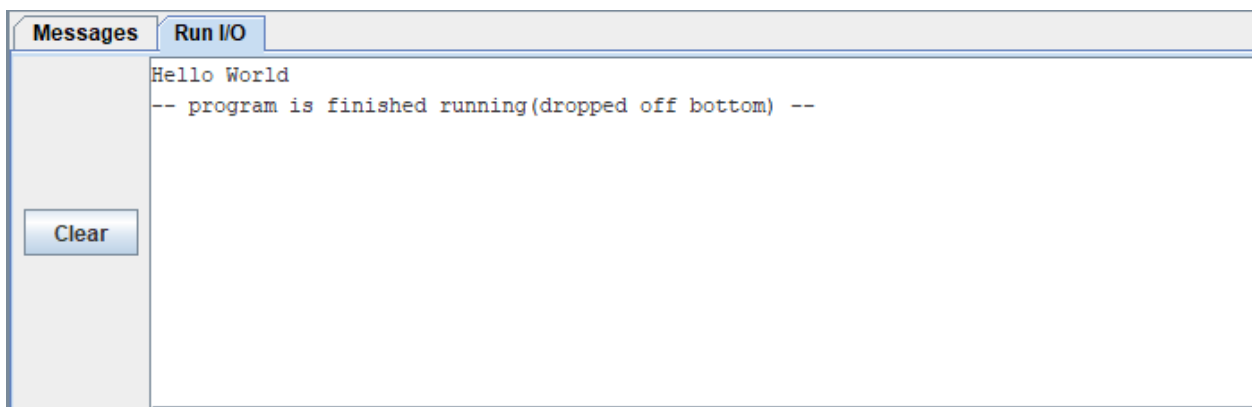
Tạo project thực hiện Home Assignment 1. Dịch và nạp chương trình lên trình mô phỏng. Chạy và quan sát kết quả. Chuyển đến Data Segment, kiểm tra cách chuỗi ký tự được lưu trữ trong bộ nhớ.

Nhập chương trình

```
1 # Laboratory Exercise 5, Home Assignment 1
2 .data
3     test: .asciz "Hello World"
4 .text
5     li     a7, 4
6     la     a0, test
7     ecall
```

```
# Laboratory Exercise 5, Home Assignment 1
.data
    test: .asciz "Hello World"
.text
    li     a7, 4
    la     a0, test
    ecall
```

Kết quả chạy



Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x10010000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000004
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400014

Data Segment:

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1 1 e 8	o W o	10 d 1 z	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010020	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010040	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010060	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010080	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x100100a0	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x100100c0	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x100100e0	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010100	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010120	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010140	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010160	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x10010180	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10
0x100101a0	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10	10 10 10 10

0x10010000 (.data)
☒ Hexadecimal Addresses
 ☒ Hexadecimal Values
 ☒ ASCII

Trong **Data Segment**: chuỗi test (“Hello World”) được lưu tuần tự trong bộ nhớ dưới dạng giá trị ASCII, với mỗi ký tự chiếm 1 byte. Chuỗi bắt đầu tại địa chỉ bộ nhớ 0x10010000 và được lưu liên tiếp theo cách sau:

H	0x10010000
e	0x10010001
l	0x10010002
l	0x10010003
o	0x10010004
space	0x10010005
W	0x10010006
o	0x10010007
l	0x10010008
r	0x10010009
d	0x1001000A

Chuỗi được kết thúc bằng một ký tự NULL (\0) tại địa chỉ 0x1001000B (giá trị 0x00), đánh dấu kết thúc của chuỗi trong bộ nhớ. Ký tự NULL này cần thiết để chỉ định điểm kết thúc của chuỗi trong bộ nhớ.

Assignment 2

Tạo project thực hiện chương trình in tổng của hai toán hạng nằm trong thanh ghi s0 và s1 theo định dạng sau:

“The sum of (s0) and (s1) is (result)”

Nhập chương trình

```

1  .data
2      message1: .asciz "The sum of " # Chuỗi đầu tiên
3      message2: .asciz " and "      # Chuỗi nối giữa
4      message3: .asciz " is "      # Chuỗi kết thúc
5  .text
6      # Khởi tạo giá trị cho các thanh ghi s0 và s1
7      li s0, 5                      # s0 = 5
8      li s1, 10                    # s1 = 10
9
10     # Tính tổng
11     add t0, s0, s1                # t0 = s0 + s1
12
13     # In ra chuỗi "The sum of"
14     la a0, message1              # Load giá trị của chuỗi message1 vào a0
15     li a7, 4                     # Service number: 4 (print string)
16     ecall                        # Gọi hệ thống để in chuỗi message1
17
18     # In giá trị của s0
19     mv a0, s0                    # Move giá trị của s0 vào a0
20     li a7, 1                     # Service number: 1 (print integer)
21     ecall                        # Gọi hệ thống để in giá trị s0
22
23     # In ra chuỗi " and"
24     la a0, message2              # Load giá trị của chuỗi message2 vào a0
25     li a7, 4                     # Service number: 4 (print string)
26     ecall                        # Gọi hệ thống để in chuỗi message2
27
28     # In giá trị của s1
29     mv a0, s1                    # Move giá trị của s1 vào a0
30     li a7, 1                     # Service number: 1 (print integer)
31     ecall                        # Gọi hệ thống để in giá trị s1
32
33     # In ra chuỗi " is "
34     la a0, message3              # Load địa chỉ của chuỗi message3 vào a0
35     li a7, 4                     # Service number: 4 (print string)
36     ecall                        # Gọi hệ thống để in chuỗi message3
37
38     # In kết quả tổng
39     mv a0, t0                    # Move giá trị tổng từ t0 vào a0
40     li a7, 1                     # Service number: 1 (print integer)
41     ecall                        # Gọi hệ thống để in kết quả tổng
42
43     # Thoát chương trình
44     li a7, 10                    # Service number: 10 (thoát chương trình)
45     ecall

```

.data

```

message1: .asciz "The sum of " # Chuỗi đầu tiên
message2: .asciz " and "      # Chuỗi nối giữa

```

```

message3: .asciz " is "          # Chuỗi kết thúc
.text
# Khởi tạo giá trị cho các thanh ghi s0 và s1
li s0, 5                        # s0 = 5
li s1, 10                       # s1 = 10

# Tính tổng
add t0, s0, s1                  # t0 = s0 + s1

# In ra chuỗi "The sum of "
la a0, message1                 # Load giá trị của chuỗi message1 vào a0
li a7, 4                        # Service number: 4 (print string)
ecall                           # Gọi hệ thống để in chuỗi message1

# In giá trị của s0
mv a0, s0                       # Move giá trị của s0 vào a0
li a7, 1                        # Service number: 1 (print integer)
ecall                           # Gọi hệ thống để in giá trị s0

# In ra chuỗi " and "
la a0, message2                 # Load giá trị của chuỗi message2 vào a0
li a7, 4                        # Service number: 4 (print string)
ecall                           # Gọi hệ thống để in chuỗi message2

# In giá trị của s1
mv a0, s1                       # Move giá trị của s1 vào a0
li a7, 1                        # Service number: 1 (print integer)
ecall                           # Gọi hệ thống để in giá trị s1

# In ra chuỗi " is "
la a0, message3                 # Load địa chỉ của chuỗi message3 vào a0
li a7, 4                        # Service number: 4 (print string)
ecall                           # Gọi hệ thống để in chuỗi message3

# In kết quả tổng
mv a0, t0                       # Move giá trị tổng từ t0 vào a0
li a7, 1                        # Service number: 1 (print integer)
ecall                           # Gọi hệ thống để in kết quả tổng

# Thoát chương trình

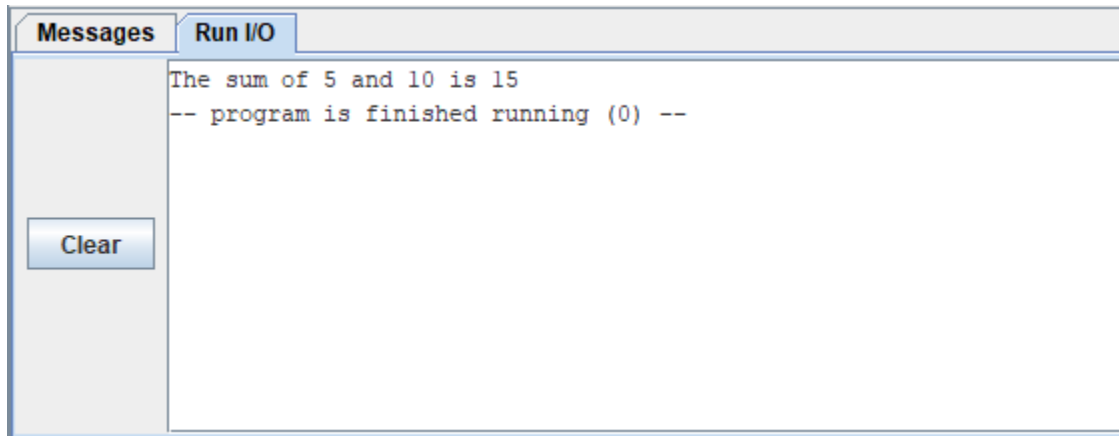
```

li a7, 10
ecall

Service number: 10 (thoát chương trình)

Kết quả chạy:

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffefc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x0000000f
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000005
s1	9	0x0000000a
a0	10	0x0000000f
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400068



Giải thích chương trình:

1. Phần **.data**:

- Đây là nơi các chuỗi ký tự như “The sum of”, “ and ”, “ is ” được lưu trữ để sử dụng khi in ra màn hình.

2. Phần **.text**:

- *li s0, 5* và *li s1, 10* dùng để gán giá trị 5 vào thanh ghi s0 và giá trị 10 vào thanh ghi s1.
- *add t0, s0, s1* tính tổng của s0 và s1, sau đó lưu kết quả vào t0.
- *la a0, message1* nạp địa chỉ của chuỗi message1 vào thanh ghi a0, sau đó gọi hệ thống với lệnh *ecall* để in chuỗi
- Các lệnh tiếp theo tương tự để in giá trị của các thanh ghi và các chuỗi khác theo định dạng yêu cầu.

3. Dịch vụ **ECALL**:

- *li a7, 4*: Dịch vụ số 4 dùng để in chuỗi ra màn hình.
- *li a7, 1*: Dịch vụ số 1 dùng để in số nguyên ra màn hình.
- *li a7, 10*: Dịch vụ số 10 dùng để thoát khỏi chương trình.

Mỗi bước sẽ in theo thứ tự từng sâu ký tự cho trước → Đúng với lý thuyết.

Assignment 3

Tạo project thực hiện Home Assignment 2. Đọc hiểu mã nguồn, khởi tạo các biến cần thiết cho chương trình, thực hiện hàm **strepy**. Dịch và nạp lên mô phỏng, chạy và quan sát kết quả.

Nhập chương trình

```

1  # Laboratory Exercise 5, Home Assignment 2
2  .data
3      x: .space 32                # Chuỗi đích x, khởi tạo là buffer rỗng
4      y: .asciz "Hello"          # Chuỗi nguồn y
5  .text
6  # Nạp địa chỉ của chuỗi đích x và chuỗi nguồn y vào thanh ghi
7      la    a0, x                 # Nạp địa chỉ của chuỗi x vào a0
8      la    a1, y                 # Nạp địa chỉ của chuỗi y vào a1
9  strcpy:
10     add    s0, zero, zero       # s0 = i = 0
11 L1:
12     add    t1, s0, a1           # t1 = s0 + a1 = i + y[0] = 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] = address of x[i]
15     sb     t2, 0(t3)            # x[i] = t2 = y[i]
16     beq    t2, zero, end_of_strcpy # if y[i] = 0, then exit
17     addi   s0, s0, 1            # s0 = s0 + 1 <-> i = i + 1
18     j      L1                  # Ký tự tiếp theo
19 end_of_strcpy:
20     # In chuỗi x ra màn hình để kiểm tra kết quả
21     la     a0, x                # Nạp địa chỉ chuỗi x vào a0
22     li     a7, 4                # Service number: 4 (print string)
23     ecall
24
25     # Thoát chương trình
26     li     a7, 10               # Dịch vụ thoát chương trình
27     ecall

```

Laboratory Exercise 5, Home Assignment 2

```

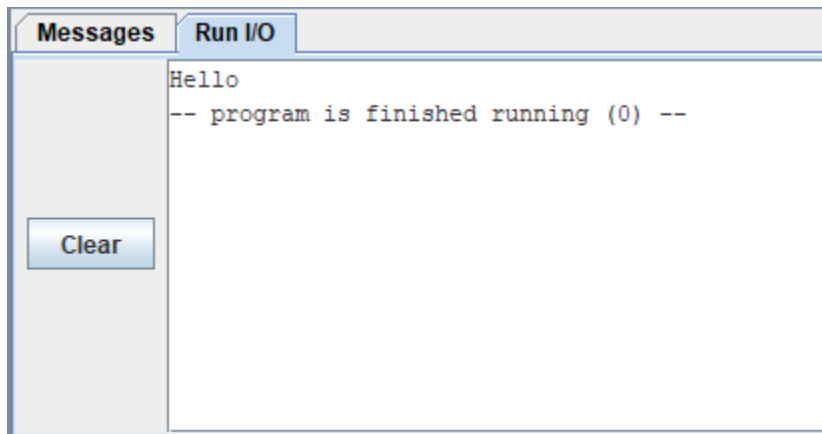
.data
    x: .space 32                # Chuỗi đích x, khởi tạo là buffer rỗng
    y: .asciz "Hello"          # Chuỗi nguồn y
.text
# Nạp địa chỉ của chuỗi đích x và chuỗi nguồn y vào thanh ghi
    la    a0, x                 # Nạp địa chỉ của chuỗi x vào a0
    la    a1, y                 # Nạp địa chỉ của chuỗi y vào a1
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, then exit
    addi   s0, s0, 1            # s0 = s0 + 1 <-> i = i + 1
    j      L1                  # Ký tự tiếp theo
end_of_strcpy:
    # In chuỗi x ra màn hình để kiểm tra kết quả

```


la	a0, x	# Nạp địa chỉ chuỗi x vào a0
li	a7, 4	# Service number: 4 (print string)
ecall		# Gọi hệ thống để in chuỗi x
	# Thoát chương trình	
li	a7, 10	# Dịch vụ thoát chương trình
ecall		

Kết quả chạy:

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x10010025
t2	7	0x00000000
s0	8	0x00000005
s1	9	0x00000000
a0	10	0x10010000
a1	11	0x10010020
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x10010005
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400048



Các bước chạy:

1. Khai báo chuỗi
 - Chuỗi đích x có 32 byte trống, chuỗi nguồn y chứa “Hello”
2. Sao chép chuỗi
 - Hàm strcpy lặp qua từng ký tự trong chuỗi y, sao chép chúng vào chuỗi x.
 - Vòng lặp dừng khi gặp ký tự NULL (\0), tức là kết thúc chuỗi
3. In kết quả:
 - Sau khi sao chép, chương trình in chuỗi x ra màn hình, tiếp theo là ký tự xuống dòng (\n)
 - Cuối cùng, thoát chương trình

Assignment 4

Tạo project thực hiện Home Assignment 3, sử dụng ecall để nhập chuỗi ký tự cần đếm, và in kết quả ra màn hình

Nhập chương trình

```
# Laboratory Exercise 5, Home Assignment 3
.data
    string: .space 50
    message1: .asciz "Nhap xau: "
    message2: .asciz "Do dai xau la: "
    newline: .byte 10          %# Giá trị của ký tự newline (ASCII: 10)
.text
main:
get_string:
    # TODO Nhập chuỗi ký tự từ bàn phím
```

```

# Hiển thị hộp thoại để người dùng nhập chuỗi
la    a0, message1          # Nạp địa chỉ của thông báo yêu cầu nhập
chuỗi

la    a1, string            # Nạp địa chỉ của chuỗi vào buffer
li    a2, 50                # Giới hạn số ký tự nhập là 50
li    a7, 54                # Mã ecall để hiển thị hộp thoại nhập chuỗi
ecall                                # Gọi ecall hiển thị hộp thoại

# Loại bỏ ký tự newline nếu có
la    a0, string            # Nạp địa chỉ của chuỗi vào a0
li    t0, 0                  # Đặt biến đếm i = 0

remove_newline:
    add    t1, a0, t0        # Tính địa chỉ của string[i]
    lb     t2, 0(t1)         # Lấy giá trị string[i]
    beq    t2, zero, and_remove_nl # Nếu gặp ký tự NULL, kết thúc vòng lặp
    li     t3, 10            # Giá trị của ký tự newline (ASCII: 10)
    beq    t2, t3, set_null  # Nếu ký tự là newline, thay thế bằng null
    addi   t0, t0, 1         # Tăng biến đếm i
    j      remove_newline   # Lặp lại vòng lặp

set_null:
    sb     zero, 0(t1)       # Thay newline bằng ký tự NULL

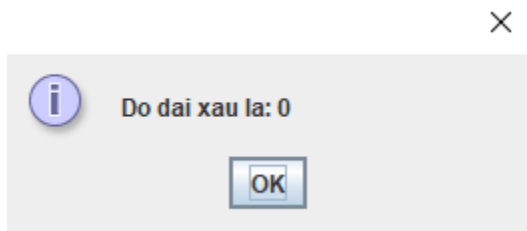
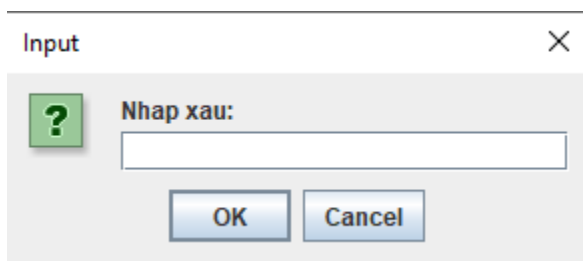
end_remove_nl:
get_length:
    la     a0, string        # a0 = address(string[0])
    li     t0, 0             # t0 = i = 0
check_char:
    add    t1, a0, t0        # t1 = a0 + t0 = address(string[0]+i)
    lb     t2, 0(t1)         # t2 = string[i]
    beq    t2, zero, end_of_str # Nếu là ký tự NULL thì kết thúc
    addi   t0, t0, 1         # t0 = t0 + 1 -> i = i + 1
    j      check_char
end_of_char:
end_of_get_length:
print_length:
    # TODO In kết quả ra màn hình
    la     a0, message2      # Nạp địa chỉ của thông báo "Đo dài xau
la: "

```

	mv	a1, t0	# Chuyển giá trị độ dài chuỗi vào thanh ghi a1
dài	li	a7, 56	# Mã ECALL để hiển thị hộp thoại thông báo độ
	ecall		# Gọi ECALL hiển thị hộp thoại
			# Thoát chương trình
	li	a7, 10	# Mã ECALL để thoát chương trình
	ecall		

Kết quả chạy:

Trường hợp 1: Không nhập chuỗi ký tự



Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x10010000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x1001003d
a1	11	0x00000000
a2	12	0x00000032
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x0040007c

Trường hợp 2: Nhập một chuỗi ký tự không vượt quá giới hạn

Input

×

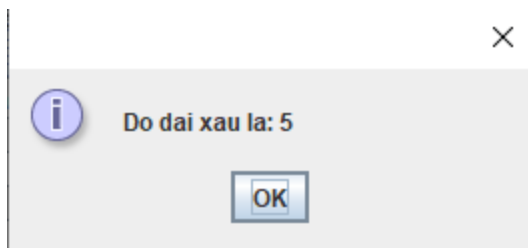
?

Nhap xau:

Thang

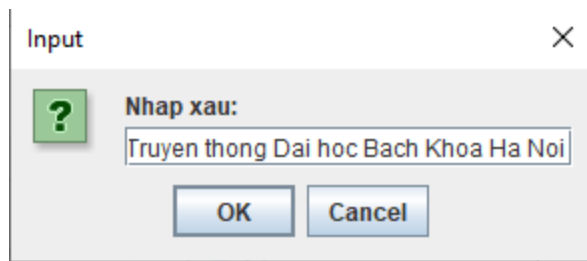
OK

Cancel

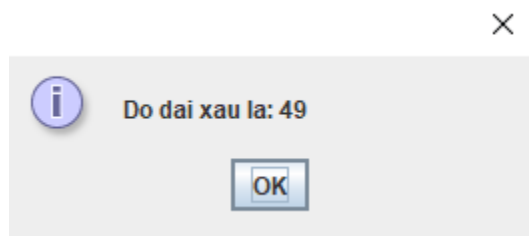


Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffefffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000005
t1	6	0x10010005
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x1001003d
a1	11	0x00000005
a2	12	0x00000032
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x0000000a
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x0040007c

Trường hợp 3: Nhập xâu có số lượng ký tự vượt quá mức cho phép



Nội dung xâu: “Trường Công nghệ Thông tin và Truyền thông Đại học Bách Khoa Hà Nội” – độ dài 88



Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000031
t1	6	0x10010031
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x1001003d
a1	11	0x00000031
a2	12	0x00000032
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x0000000a
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x0040007c

Giải thích kết quả chạy:

- Nhập chuỗi từ hội thoại:
 - Sử dụng ECALL (mã 54) để hiển thị hộp thoại nhập chuỗi và lưu chuỗi vào biến string.
- Loại bỏ ký tự xuống dòng (newline):
 - Vòng lặp remove_newline kiểm tra từng ký tự của chuỗi.
 - Nếu phát hiện ký tự xuống dòng (mã ASCII: 10), nó sẽ thay thế bằng ký tự NULL (\0).
- Tính độ dài chuỗi:

- Vòng lặp `check_char` duyệt qua chuỗi từ đầu đến khi gặp ký tự `NULL` (`\0`) để đếm số ký tự.
4. Hiển thị độ dài chuỗi
- Sử dụng `ECALL` (mã 56) để hiển thị độ dài chuỗi trong hộp thoại tin nhắn
5. Thoát chương trình
- Sử dụng `ECALL` (mã 10) để thoát chương trình sau khi hiển thị kết quả.

Trường hợp không nhập xâu ký tự

- Khi không nhập gì trong hộp thoại nhập (chỉ bấm “Enter”), thì nội dung chuỗi `string` chỉ chứa ký tự `newline` (`\n`, ASCII: 10) và kết thúc bằng ký tự `NULL` (`\0`) tự động thêm vào bởi hệ thống nhập chuỗi (`ECALL` 54).
- Sau đó, đoạn mã `remove_newline` sẽ kiểm tra từng ký tự, nếu thấy `newline` (ASCII: 10) thì thay thế bằng `NULL` (`\0`), như đoạn mã sau:

beq t2, t3, set_null # Nếu gặp newline -> set null

⇒ Kết quả: `string[0]` sẽ trở thành `\0`, tức là một chuỗi rỗng

Đến phần đo độ dài chuỗi:

lb t2, 0(t1)

beq t2, zero, end_of_str

- Ngay từ lần đầu đọc ký tự, thấy là `\0` ⇒ dừng ngay ⇒ độ dài chuỗi = 0

Trường hợp nhập quá độ dài cho phép:

Khi bạn nhập nhiều hơn 50 ký tự, thì:

- Trong lệnh: *li a2, 50* # Giới hạn số ký tự nhập, `ECALL` 54 sẽ chỉ đọc tối đa 50 ký tự, các ký tự sau đó sẽ bị cắt bỏ (bỏ qua)
- Ngoài ra, trong 50 ký tự đó, hệ thống vẫn thêm ký tự `NULL` (`\0`) ở cuối để kết thúc chuỗi

⇒ Kết quả: Chỉ phần đầu chuỗi được lưu vào biến `string`, phần dư bị bỏ

⇒ Phần đếm độ dài vẫn hoạt động bình thường: chương trình đếm đến `\0`, nên kết quả là số ký tự hợp lệ đã được nhập (tối đa 49)

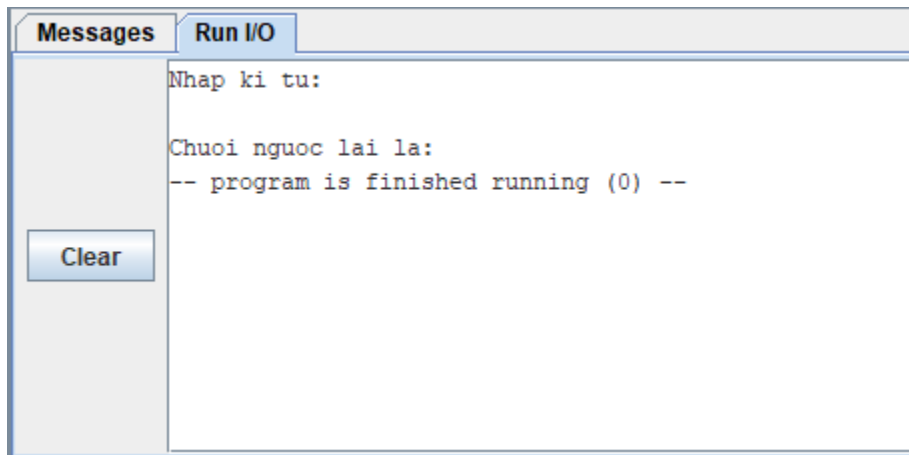
ecall báo	# Thực hiện lệnh syscall để in ra thông
li t0, 0	# t0 là chỉ số hiện tại của chuỗi (để đếm số
la t1, string (string)	# t1 trỏ đến vị trí bắt đầu của chuỗi
get_input:	
li a7, 12	# Mã hệ thống để đọc ký tự (syscall 12)
ecall	# Thực hiện syscall để đọc ký tự từ người
dùng	
sb a0, 0(t1)	# Lưu ký tự vừa nhập vào vị trí hiện tại
của chuỗi (t1)	
li t2, 10	# t2 giữ giá trị '\n' (ASCII: 10)
beq a0, t2, end_input	# Nếu ký tự nhập vào là '\n' (Enter), thì kết
thúc nhập	
addi t0, t0, 1	# Tăng chỉ số chuỗi (số ký tự đã nhập)
addi t1, t1, 1	# Di chuyển con trỏ chuỗi (t1) đến vị trí
tiếp theo	
li t2, 20	# t2 giữ giá trị 20 (số ký tự tối đa có thể
nhập)	
bge t0, t2, end_input	# Nếu số ký tự nhập vào >= 20, thì dừng
nhập	
j get_input	# Quay lại vòng lặp nhập ký tự tiếp theo
end_input:	
sb zero, 0(t1)	# Thêm ký tự NULL '\0' vào cuối chuỗi để
kết thúc chuỗi	
la t1, string	# t1 trở lại vị trí đầu tiên của chuỗi (string)
add t2, t1, t0	# t2 trỏ đến vị trí cuối cùng của chuỗi (dựa
trên số ký tự t0)	
addi t2, t2, -1	# t2 trỏ đến ký tự cuối cùng thực sự của
chuỗi (không phải NULL)	
addi t3, t0, -1	# t3 là chỉ số dùng để duyệt ngược chuỗi

la	a0, msg2	# Đưa ra thông báo "Chuoi nguoc lai la: "
vào a0		

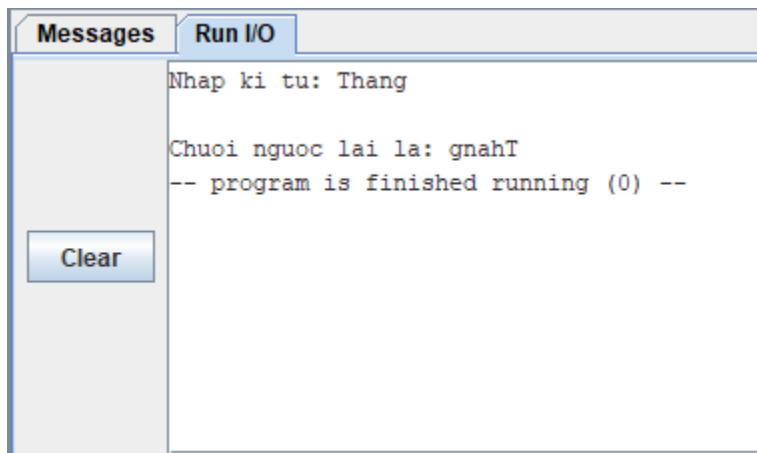
Kết quả chạy chương trình:

Trường hợp 1: Không nhập xâu ký tự

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x10010000
t2	7	0x1000ffff
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x10010022
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0xffffffff
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400090



Trường hợp 2: Nhập xâu ký tự không vượt quá số ký tự cho phép



Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000005
t1	6	0x10010000
t2	7	0x1000ffff
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000054
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0xffffffff
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400090

Trường hợp 3: Nhập quá số ký tự

Messages	Run I/O
Nhập kí tự: 11111222223333344444 Chuoi nguoc lai la: 44444333332222211111 -- program is finished running (0) --	
<div>Clear</div>	

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000014
t1	6	0x10010000
t2	7	0x1000ffff
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000031
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x0000000a
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0xffffffff
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400090

Giải thích kết quả chạy:

1. In thông báo "Nhap ky tu: ":
 - Chương trình in thông báo yêu cầu người dùng nhập chuỗi.
2. Nhập chuỗi từ bàn phím:
 - Chương trình cho phép người dùng nhập từng ký tự vào chuỗi. Khi người dùng nhấn Enter ('\n') hoặc chuỗi đạt đến 20 ký tự, quá trình nhập dừng lại.
3. Kết thúc chuỗi:
 - Sau khi nhập xong, chương trình thêm ký tự null ('\0') để đánh dấu kết thúc chuỗi.

4. In thông báo "Chuoi nguoc lai la ":
 - Chương trình in thông báo chuẩn bị hiển thị chuỗi ngược.
5. In chuỗi ngược:
 - Chương trình bắt đầu từ ký tự cuối cùng của chuỗi và in ngược dần về ký tự đầu tiên.
6. Thoát chương trình:
 - Sau khi in chuỗi ngược xong, chương trình kết thúc bằng lệnh thoát.

TH Không nhập ký tự:

- Khi chương trình chạy đến phần nhập ký tự đầu tiên (syscall 12), nếu người dùng không nhập gì mà chỉ nhấn Enter thì ký tự đầu tiên được đọc sẽ là ký tự newline `\n` (ASCII: 10). Lệnh kiểm tra `beq a0, t2, end_input` sẽ so sánh ký tự nhập với 10 (newline), nếu đúng thì thoát khỏi vòng lặp nhập.
- Lúc này, chuỗi string chưa lưu ký tự nào, và chương trình thực hiện `sb zero, 0(t1)` để gán ký tự null (`\0`) vào đầu chuỗi nhằm kết thúc chuỗi.
- Kết quả là chuỗi rỗng (`string[0] = '\0'`). Khi đến phần in chuỗi ngược, vì độ dài chuỗi là 0, vòng lặp in ngược không thực hiện gì cả. Do đó chỉ hiển thị dòng "Chuoi nguoc lai la: " mà không có ký tự nào được in thêm.

TH Nhập quá 20 ký tự:

- Giới hạn độ dài chuỗi được kiểm tra trong đoạn `bge t0, t2, end_input`, với `t2 = 20`. Nghĩa là chương trình chỉ cho phép nhập tối đa 20 ký tự. Khi số ký tự nhập vào đạt 20 thì vòng lặp nhập sẽ kết thúc, các ký tự sau sẽ không được lưu.
- Khi kết thúc nhập, chương trình gán ký tự null vào cuối chuỗi để kết thúc. Như vậy, dù người dùng gõ nhiều hơn 20 ký tự thì chương trình chỉ ghi nhận 20 ký tự đầu tiên.
- Phần in chuỗi ngược sẽ in ngược đúng 20 ký tự đầu tiên, các ký tự dư không được lưu nên không được in ra.