# 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**

**A screenshot of a computer

AI-generated content may be incorrect.**

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

**Kết quả chạy**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A table with numbers and lines

AI-generated content may be incorrect.**

**Data Segment:**

**A screenshot of a computer

AI-generated content may be incorrect.**

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**

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| .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 # Service number: 10 (thoát chương trình)  ecall |

**Kết quả chạy**:

A table with numbers and a green line

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**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 **strcpy**. Dịch và nạp lên mô phỏng, chạy và quan sát kết quả.

**Nhập chương trình**

A screenshot of a computer program

AI-generated content may be incorrect.

|  |
| --- |
| # 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**:

A table with numbers and a green line

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**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 hội 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\_newlime:  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 # Nap dia chi cua thong bao "Do dai xau la: "  mv a1, t0 # Chuyen gia tri do dai chuoi vao thanh ghi a1  li a7, 56 # Mã ECALL để hiển thị hộp thoại thông báo độ dài  ecall # Goi ECALL hien thi hop thoai    # Thoat chuong trinh  li a7, 10 # Ma ECALL de thoat chuong trinh  ecall |

**Kết quả chạy**:

Trường hợp 1: Không nhập xâu ký tự

A screenshot of a computer error message

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A table with numbers and lines

AI-generated content may be incorrect.

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

A screenshot of a computer error

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A table with numbers and letters

AI-generated content may be incorrect.

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

A screenshot of a computer

AI-generated content may be incorrect.

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

A screenshot of a computer

AI-generated content may be incorrect.

A table of numbers and letters

AI-generated content may be incorrect.

**Giải thích kết quả chạy**:

1. Nhập chuỗi từ hội thoại:
   * Sử dụng ECALL (mã 54) để hiển thị hộp thoại nhập chuỗi và lưu chuỗi vào biến string.
2. 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).
3. 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)

## Assignment 5

Viết chương trình cho phép người dùng nhập chuỗi ký tự bằng cách nhập từng ký tự từ bàn phím. Việc nhập sẽ kết thúc khi người dùng nhấn Enter hoặc khi độ dài chuỗi ký tự vượt quá 20. In chuỗi đã nhập theo chiều ngược lại.

**Nhập chương trình**

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| .data  string: .space 21 # Cấp phát 21 byte cho chuỗi (bao gồm 20 ký tự và 1 ký tự NULL)  msg: .asciz "Nhap ki tu: " # Thông báo yêu cầu nhập ký tự  msg2: .asciz "\nChuoi nguoc lai la: " # Thông báo về chuỗi ngược  .text  get\_str:  la a0, msg # Đưa thông báo "Nhap ky tu: " vào thanh ghi a0 để in ra  li a7, 4 # Mã hệ thống để in chuỗi (syscall 4)  ecall # Thực hiện lệnh syscall để in ra thông báo    li t0, 0 # t0 là chỉ số hiện tại của chuỗi (để đếm số ký tự)  la t1, string # t1 trỏ đến vị trí bắt đầu của chuỗi (string)    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

A table of numbers and digits

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

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

A screenshot of a computer

AI-generated content may be incorrect.

A table of numbers and letters

AI-generated content may be incorrect.

**Trường hợp 3**: Nhập quá số ký tự

A screenshot of a computer

AI-generated content may be incorrect.

A table of numbers and letters

AI-generated content may be incorrect.

**Giải thích kết quả chạy:**

1. In thông báo "Nhap ky tu: ":

* Chương trình in thông báo yêu cầu người dùng nhập chuỗi.

1. 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.

1. Kết thúc chuỗi:

* Sau khi nhập xong, chương trình thêm ký tự null ('\0') để đánh dấu kết thúc chuỗi.

1. In thông báo "Chuoi nguoc lai la: ":

* Chương trình in thông báo chuẩn bị hiển thị chuỗi ngược.

1. In chuỗi ngược:

* Chương trình bắt đầu từ ký tự cuối cùng của chuỗi và in ngược dần về ký tự đầu tiên.

1. 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.