

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA TP.HCM**

KHOA CÔNG NGHỆ THÔNG TIN



ASSIGNMENT 04.01

HỆ ĐIỀU HÀNH

Lớp: 21CLC03

Sinh viên: Trần Nguyên Huân

MSSV: 21127050

Giảng viên hướng dẫn: Thái Hùng Văn, Đặng Trần Minh Hậu

THÀNH PHỐ HỒ CHÍ MINH - 2023

Mục lục

I. Phân tích chương trình demo thao tác nhập xuất trực tiếp bằng ngắt của ROM BIOS	2
II. Thiết lập và chạy chương trình	3
III. Nguồn tham khảo	6

I. Phân tích chương trình demo thao tác nhập xuất trực tiếp bằng ngắt của ROM BIOS

- Chương trình này sử dụng ngôn ngữ Assembly x86, sử dụng phần mềm MASM (Microsoft Macro Assembler) để biên dịch.

- Trong chương trình, có sử dụng hai hàm ngắt BIOS ROM là **int 16h** và **int 10h**.

- Hàm ngắt **int 16h** được sử dụng để nhập ký tự từ bàn phím. Khi chương trình đến điểm nhập ký tự, nó gọi ngắt **int 16h** với đầu vào **ah** bằng **00h**. Điều này yêu cầu BIOS đọc một ký tự từ bàn phím mà không hiển thị lên màn hình và trả về mã ASCII của ký tự đó trong thanh ghi **al**. Sau đó, chương trình kiểm tra xem ký tự vừa nhập có phải là ký tự Enter (mã ASCII **0Dh**) không. Nếu phải, chương trình kết thúc vòng lặp nhập dữ liệu. Nếu không, ký tự đó được lưu vào mảng **str** và vòng lặp tiếp tục.

```
input_loop:
    mov ah, 00h ; nhap mot ki tu khong hien thi len man hinh
    int 16h
    mov dl, al ; luu ki tu vua nhap vao thanh ghi DL
    cmp dl, 0Dh ; kiem tra neu ki tu vua nhap la Enter
    je end_input
    mov [si], dl ; luu ki tu vua nhap vao mang
    inc si
    inc cx

    ; In ky tu vua nhap len man hinh
    mov ah, 02h ; dich chuyen con tro den vi tri tiep theo
    mov dl, al ; lay ki tu vua nhap tu thanh ghi DL
    int 21h

    jmp input_loop
```

- Hàm **int 10h** được sử dụng để hiển thị ký tự lên màn hình. Đầu vào của hàm là giá trị **ah** là **02h** để chỉ định chế độ in ký tự. Ký tự cần in được lưu trong thanh ghi **dl**. Hàm **int 10h** được sử dụng trong vòng lặp để in từng ký tự của chuỗi đã nhập vào. Ngoài ra, hàm này cũng được sử dụng để in ký tự xuống dòng (**0Dh**) và xuống dòng mới (**0Ah**) khi kết thúc in chuỗi.

```
print_loop:
    lodsb ; n?p m?t byte t? DS:[SI] vao AL va tang SI l?n 1
    cmp al, 0 ; kiem tra neu ki tu la null
    je end_print
    int 10h ; in ky tu
    jmp print_loop
```

- Ngoài ra, Hàm ngắt BIOS **int 21h** với ngắt loại 9 (**AH = 9**) trong chương trình trên được sử dụng để thực hiện các chức năng nhập và xuất dữ liệu trên màn hình console của DOS

- Hàm 09h: In ra chuỗi ký tự đã được lưu trữ trong bộ đệm (buffer) của một địa chỉ được xác định. Đây là hàm được sử dụng để in ra các chuỗi thông báo, ví dụ như trong đoạn mã **mov dx, offset prompt_msg** và **mov dx, offset output_msg**.

```
; In prompt message
mov ah, 09h
mov dx, offset prompt_msg
int 21h
```

```
; In output message
mov ah, 09h
mov dx, offset output_msg
int 21h
```

- Hàm 02h: Di chuyển con trỏ đến một vị trí cụ thể trên màn hình console. Đây là hàm được sử dụng để di chuyển con trỏ đến vị trí kế tiếp để in ký tự, ví dụ như trong đoạn mã **mov ah, 02h ; dịch chuyển con trỏ đến vị trí tiếp theo**.

```
; In ký tự xuống dòng
mov ah, 02h ; dịch chuyển con trỏ đến vị trí tiếp theo
mov dl, 0Ah ; ký tự xuống dòng
int 21h
```

I

- Hàm 0Eh: In một ký tự lên màn hình console và di chuyển con trỏ đến vị trí tiếp theo. Đây là hàm được sử dụng để in ra các ký tự trong chuỗi đã nhập, ví dụ như trong đoạn mã **mov ah, 0Eh ; ngắt in ký tự**

* Chương trình sử dụng hàm ngắt 21h (hàm ngắt hệ thống được cung cấp bởi DOS) để có thể in ra các thông điệp prompt, thông điệp output và hiển thị các ký tự input khi nhập xong giúp dễ dàng theo dõi quá trình thực thi chương trình. Chương trình hoàn toàn có thể nhập xuất trực tiếp bằng các ngắt BIOS ROM mà không cần thông qua hàm 21h nhưng có thể sẽ khó theo dõi hơn.

II. Thiết lập và chạy chương trình

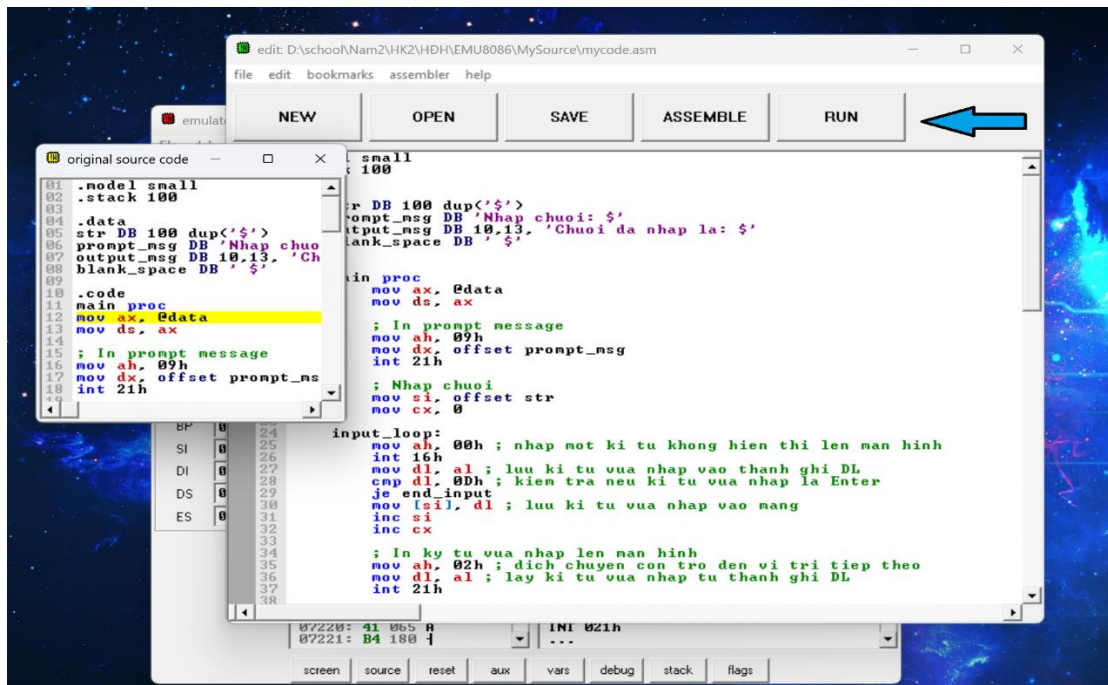
- Để có thể chạy chương trình thì em có cài đặt phần mềm emu8086 - một phần mềm giả lập cho phép chạy các chương trình viết cho bộ vi xử lý 8086. (link download: <https://emu8086.en.lo4d.com/windows>)

- Video chạy chương trình demo được đính kèm trong folder nộp.
- Chạy chương trình:

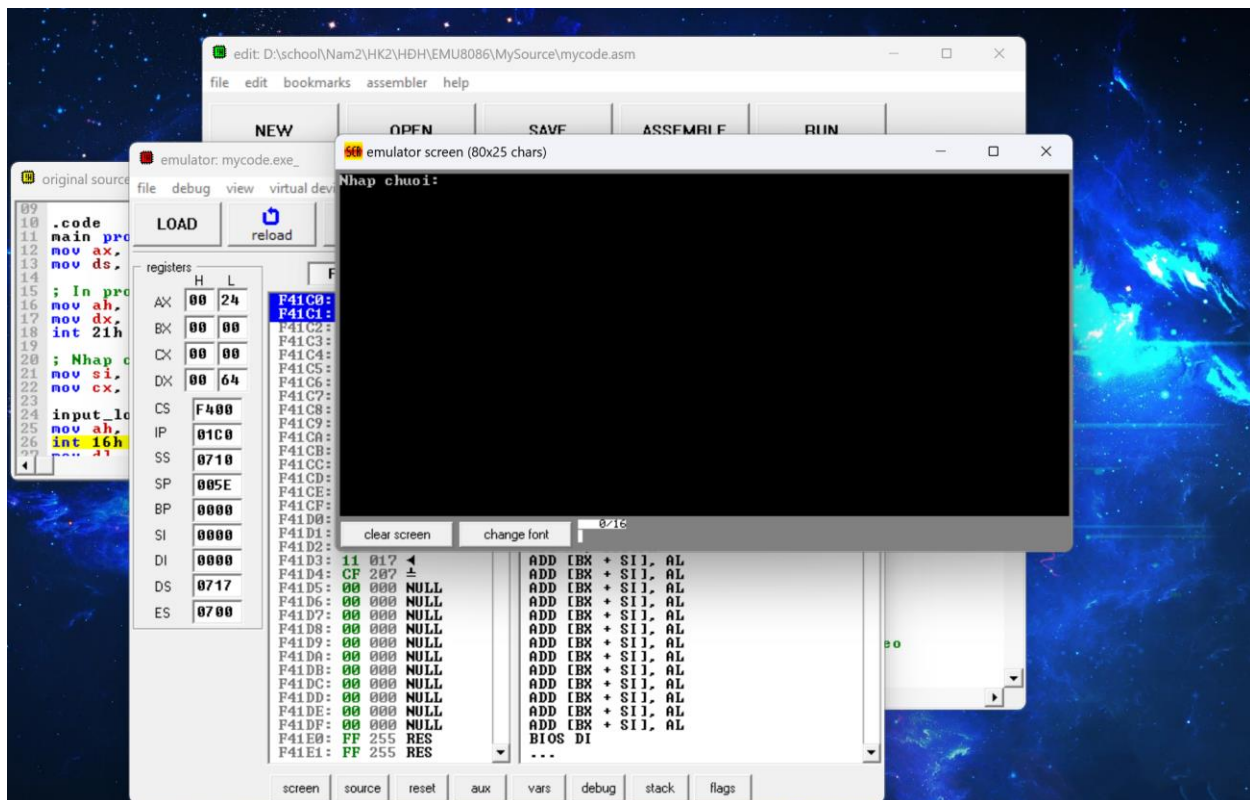
The screenshot shows a window titled "edit: D:\school\Nam2\HK2\HDH\EMU8086\MySource\mycode.asm". The window has a menu bar with "file", "edit", "bookmarks", "assembler", and "help". Below the menu bar are five buttons: "NEW", "OPEN", "SAVE", "ASSEMBLE", and "RUN". The main area contains assembly code with line numbers 01 to 38 on the left. The code defines a data segment with a string buffer, a prompt message, an output message, and a blank space. It then defines a code segment with a "main" procedure. The procedure initializes the data segment, prints a prompt, enters a loop to read characters from the keyboard, and prints the input string. Comments in Vietnamese are provided for several instructions.

```
01 .model small
02 .stack 100
03
04 .data
05     str DB 100 dup('$')
06     prompt_msg DB 'Nhap chuoi: $'
07     output_msg DB 10,13, 'Chuoi da nhap la: $'
08     blank_space DB '$'
09
10 .code
11     main proc
12         mov ax, @data
13         mov ds, ax
14
15         ; In prompt message
16         mov ah, 09h
17         mov dx, offset prompt_msg
18         int 21h
19
20         ; Nhap chuoi
21         mov si, offset str
22         mov cx, 0
23
24     input_loop:
25         mov ah, 00h ; nhap mot ki tu khong hien thi len man hinh
26         int 16h
27         mov dl, al ; luu ki tu vua nhap vao thanh ghi DL
28         cmp dl, 0Dh ; kiem tra neu ki tu vua nhap la Enter
29         je end_input
30         mov [si], dl ; luu ki tu vua nhap vao mang
31         inc si
32         inc cx
33
34         ; In ky tu vua nhap len man hinh
35         mov ah, 02h ; dich chuyen con tro den vi tri tiep theo
36         mov dl, al ; lay ki tu vua nhap tu thanh ghi DL
37         int 21h
38     end proc
```

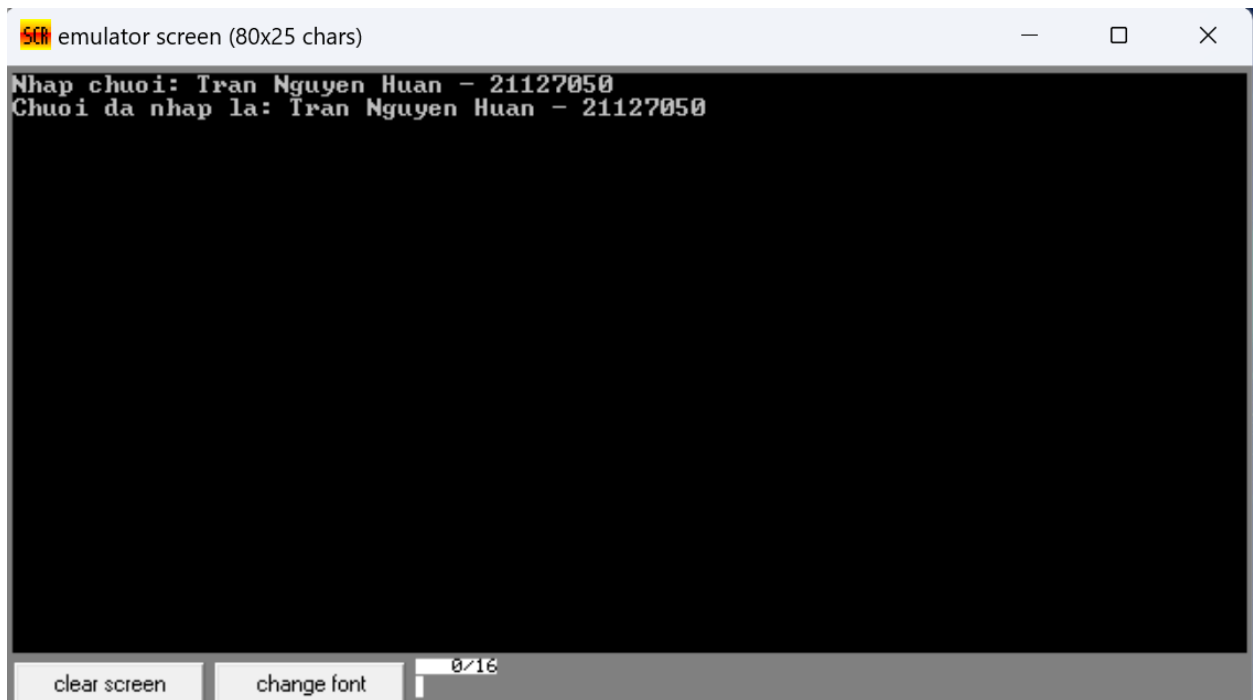
Hình: Màn hình giao diện ban đầu.



Hình: Màn hình sau khi bấm RUN



Hình: Màn hình sau khi bấm RUN emulator: mycode.exe



Hình: Màn hình emulator screen sau khi nhập chuỗi và xuất ra chuỗi đã nhập

III. Nguồn tham khảo

- (1): <https://www.cs.virginia.edu/~evans/cs216/guides/x86.html>
- (2): https://en.wikipedia.org/wiki/X86_assembly_language
- (3): Diễn đàn: <https://www.vogons.org/viewtopic.php?t=62693>