Bài 5a:

Biên dịch và cho chạy file BAI\_5A.ASM để kiểm tra và xem kết quả. Gợi ý: Thư mục TT\_ASM phải có sẵn trong ổ đĩa. Để biết chương trình chạy đúng hay sai, vào thư mục TT\_ASM để xem có tập tin Data.txt hay không, nếu có thì OK.

1. Tại sao thẻ file phải được khai báo như dạng thefile DW ?

Vì trong bài có sự xuất hiện của câu lệnh mov thefile, ax; Nếu thefile là dạng DB (8 bit) sẽ không chứa được thanh ghi ax 16bit;

1. Tại sao trong trường hợp này, chúng ta không phải dùng hàm 08h của int 21h đứng trước hàm 4ch của int 21h ?

Vì không có nhu cầu đọc cái gì in ra từ màn hình cả, nên không cần 08h trước 4ch để giữ màn hình;

1. Trong đoạn chương trình mẫu trên có cần thiết phải đóng tập tin hay không ?

Không cần, nếu không đóng tập tin thì chương trình vẫn chạy bình thường;

1. Có thể bỏ biến thefile trong đoạn chương trình mẫu trên không ?. Khi đó chúng ta phải dùng các lệnh gì để thay thế điều đó. Nếu có thay đổi, hãy biên dịch và cho chạy chương trình để kiểm chứng lại kết quả.

Có thể bỏ qua biến thefile bằng cách trực tiếp xóa các câu lệnh liên quan mà vẫn cho ra được kết quả như mong muốn ( tạo ra được tệp data.txt) Các câu lệnh bị xóa:   
;mov thefile, ax ; cat the file  
;mov ah, 3eh ; dong tap tin  
;mov bx, thefile  
;int 21h

1. Hãy sửa đổi file BAI\_5A.ASM và lưu với tên BAI\_5A1.ASM để có thể thực hiện được yêu cầu sau: tạo **một tập tin mới, tên tập tin được nhập từ bàn phím**.

Có hai cách: Nhập mảng (mov ah,1) và nhập chuỗi (mov ah,9) Với nhập mảng thì sau khi nhập mỗi phần tử thì sử dụng CMP để kiểm tra đó có phải dấu ENTER hay chưa; Với nhập chuỗi, có gợi ý trong slide.

Bài 5b: Soạn thảo như đoạn chương trình mẫu phía dưới và lưu với tên là BAI\_5B.ASM.

1. Biên dịch và cho chạy file BAI\_5B.ASM để kiểm tra và xem kết quả. (vào D:\TT\_ASM để xem tập tin DATA.TXT có trong đó hay chưa và có nội dung hay chưa ?, nếu có là OK.)
2. Xem xét đoạn chương trình mẫu, hãy đưa ra giải thuật ghi nội dung của vùng dữ liệu vào một tập tin vừa tạo.

|  |  |
| --- | --- |
| mov ah, 40h ; ghi file  mov bx, thefile  xor cx, cx | mov cl, len  lea dx, string1  int 21h |

1. Lệnh *len db $ - string1* được dùng để làm gì ?.

Khai báo biến len là độ dài của String1(?)

1. Lệnh *XOR CX, CX* có ý nghĩa gì? Sau khi thực hiên xong lệnh này, thanh ghi CX có giá trị bằng bao nhiêu? Có thể thay thế nó bằng lệnh nào khác được không ?

[dựa theo slide] XOR CX, CX có tác dụng đưa zero về cuối chuỗi. Với ví dụ 5B, CX trước và sau khi XOR đều là 00 – không có gì thay đổi cả, có thể xóa đi mà không cần thay thế bằng cái khác;

1. Tại sao dùng lệnh *MOV CL, LEN* mà không dùng *MOV CX, LEN* hay *MOV CH, LEN* ? Khi dùng lệnh *MOV CX, LEN* thì cần phải thay đổi khai báo biến **len** như thế nào ? nếu không thay đổi thì sẽ có vấn đề gì xảy ra hay không ? Hãy thay đổi, biên dịch và chạy chương trình để kiểm chứng lại kết quả so với chương trình mẫu.

Vì LEN là DB nên không thể dùng CX, LEN. Nếu muốn dùng lệnh MOV CX,LEN thì khi khai báo phải đổi LEN thành dạng DW; Nếu không thay đổi, chương trình sẽ báo lỗi và không thể thực thi.

1. Hãy sửa đổi file BAI\_5B.ASM và lưu với tên BAI\_5B1.ASM để có thể thực hiện được yêu cầu sau: nhập từ một chuỗi ký tự bất kỳ, sau đó lưu vào tập tin có tên là “d:\tt\_asm\solieu.txt”. Gợi ý: dùng hàm 0Ah của int 21h để nhập vào một chuỗi ký tự, sau đó áp dụng toàn bộ giải thuật của BAI\_5B.ASM.

Ý tưởng: Từ BAI\_5B, đổi biến String1 thành biến String trong đó tạo ra một hàm nhập riêng, kết thúc bằng dấu enter; Full code:

|  |  |  |
| --- | --- | --- |
| dseg segment  chuoi Db 100 DUP(0)  len db $ - string1  tenfile db "E:\emu8086\emu8068\_loca\ emu8086\MyBuild\data.txt",0  thefile dw ?  dseg ends  cseg segment  assume cs:cseg, ds:dseg  begin: mov ax, dseg  mov ds, ax  MOV SI, 0 ; chi so mang  MOV CX, 100 ; do dai mang max | LAP:  MOV AH, 1 ; nhap ky tu  INT 21H  CMP AL,0Dh ;check dau enter  JE STOP  MOV chuoi[SI], AL  INC SI  JMP LAP  STOP:  mov ah, 3ch ; tao tap tin moi  lea dx, tenfile  mov cx, 0 ; tap tin co thuoc tinh binh thuong  int 21h  mov cx, si  mov thefile, ax ; cat the file  mov ah, 40h ; ghi file  mov bx, thefile | GHI\_FILE:  ;xor cx, cx  ;mov cl, len  mov dh,0  mov dl, chuoi[SI]  ;lea dx, string1  int 21h  LOOP GHI\_FILE  mov ah, 3eh ; dong tap tin  mov bx, thefile  int 21h  mov ah, 4ch ; thoat ve Dos  int 21h  cseg ends  end begin |

1. Hãy sửa đổi file BAI\_5B1.ASM và lưu với tên BAI\_5B2.ASM để có thể thực hiện được yêu cầu sau: tạo **tên tập tin mới, tên tập tin được nhập từ bàn phím (trùng BAI\_5A1)**  
   Sau đó nhập vào một chuỗi ký tự bất kỳ và lưu chuỗi ký tự đã nhập vào tập tin vừa tạo. Gợi ý: xem lại BAI\_5A1.ASM để lấy lại giải thuật nhập vào tên file và đưa zero (con số 0) về cuối chuỗi đối với trường hợp tên file được nhập từ bàn phím và các vấn đề còn lại thì xem lại file BAI\_5B1.ASM.

**[Hoàn thành, lưu là BAI\_5B2]**

Bài 5c : Đọc nội dung tập tin

1. Xem lại đoạn chương trình mẫu, hãy đưa ra giải thuật đọc nội dung của tập tin và hiển thị nội dung đó ra màn hình.

|  |  |
| --- | --- |
| Đọc nội dung tập tin | In ra màn hình |
| mov ah, 3fh ; doc noi dung file vao vung dem  mov bx, thefile  lea dx, buffer  mov cx, 250 ; so byte can doc tu file da mo  int 21h | mov ah, 09h ; in noi dung cua file ra man hinh  lea dx, buffer  int 21h  mov ah,08h ; dung man hinh de xem ket qua  int 21h |

1. Hãy thử thay đổi thuộc tính tập tin trong lệnh mov al, 2 lần lượt thành các giá trị khác như 0, 1, 3 hoặc 4. Biện dịch và cho chạy chương trình để xem kết quả. Có nhận xét gì về các giá trị này ?.

Giá trị 0: Vẫn đọc được nội dung tệp;  
Giá trị 1: Không đọc được tệp;  
Giá trị 3,4: Tin nhắn trả về: set AL=2 (read/write/shared) or refer to int 21h/4Dh reference;

1. Nếu số byte cần đọc (giá trị chứa trong thanh ghi cx) lớn hơn kích thước thật sự của tập tin thì có gây ra lỗi gì hay không ?

Không gây ra lỗi gì, trái lại, nếu giá trị của CX nhỏ hơn nội dung tệp thì màn hình chỉ hiện lên số ký tự đúng như giá trị CX

1. Sau khi đọc nội dung của tập tin vào vùng đệm bằng hàm 3fh của INT 21h, thanh ghi ax sẽ có giá trị thay đổi hay không và nó chứa (giá trị) gì ?

Thanh ghi AX sẽ thay đổi là AH được gán bằng 3F; Đến lệnh int 21h, thanh ghi AX chuyển thành AL=3

1. Làm sao xác định được khi nào đọc xong nội dung thành tập tin ?. Hãy thử đưa ra hướng giải quyết?

Đọc xong khi chạy hết số vòng lặp lưu ở CX

1. Hãy sửa đổi file BAI\_5C.ASM và lưu với tên BAI\_5C1.ASM để có thể thựchiện được yêu cầu sau: đọc nội dung của một tập tin và hiển thị nội dung đó lên màn hình. Tên tập tin được nhập từ bàn phím.

[Hoàn thành]

1. Hãy sửa đổi file BAI\_5C1.ASM và lưu với tên BAI\_5C2.ASM để có thể thực hiện được yêu cầu sau: copy nội dung của một tập tin bất kỳ sau đó paste sang một vị trí khác.

[Hoàn thành]

1. Hãy sửa đổi file BAI\_5C2.ASM và lưu với tên BAI\_5C3.ASM để có thể thực hiện được yêu cầu sau: copy nội dung của một tập tin bất kỳ sau đó paste sang một vị trí khác. Tên tập tin được copy và tập tin sau khi paste đều nhập từ bàn phím.

[Hoàn thành]

1. Hãy sửa đổi file BAI\_5C3.ASM và lưu với tên BAI\_5C4.ASM để có thể thực hiện được yêu cầu sau: save as nội dung của một tập tin. Tên tập tin được copy và save as đều nhập từ bàn phím.

Code của bài y hệt bài 5C3, chỉ khác địa chỉ của tệp đích là một tệp mới chưa có trong máy;

1. Hãy sửa đổi file BAI\_5C3.ASM và lưu với tên BAI\_5C4.ASM để có thể thực hiện được yêu cầu sau: đọc nội dung của tập tin và sau đó nhập một chuỗi ký tự bất kỳ và ghi tiếp theo sau nội dung của tập tin vừa mở. Tên tập tin nhập từ bàn phím.

**Gợi ý:** xem lại BAI\_5C1.ASM, BAI\_5B1.ASM, BAI\_5B2.ASM và hàm 42h của int 21h (dời vị trí con trỏ tập tin).

[Hiện tại vẫn chưa dời được vị trí con trỏ, việc ghi thêm nội dung là ghi ở ngay trên cùng của tệp]

1. Hãy sửa đổi file BAI\_5C4.ASM và lưu với tên BAI\_5C5.ASM để có thể thực hiện được yêu cầu sau: mã hóa nội dung của tập tin. Tên tập tin cần mà hóa được nhập từ bàn phím.

**Gợi ý:** nhập vào tên file cần mã hóa, mở một file đã có bằng hàm 3dh; đọc nội dung của tập tin vào vùng đệm bằng hàm 3fh; mã hóa vùng đệm đọc được bằng một trong các phép toán cộng, trừ, nhân, chia, and, or, not …, vừa mã hóa vừa lưu nội dung trở lại vùng đệm; dời vị trí con trỏ tập tin về đầu tập tin bằng hàm 42h; sau đó ghi nội dung của vùng đệm trở lại tập tin ban đầu thông qua thẻ file của nó. Tất cả các hàm sử dụng ở đây đều của int 21h.

1. Hãy sửa đổi file BAI\_5C5.ASM và lưu với tên BAI\_5C6.ASM để có thể thực hiện được yêu cầu sau: giải mã nội dung của tập tin đã mã hóa. Tên tập tin cần giải mã được nhập từ bàn phím.

**Gợi ý**: giải mã là trường hợp ngược lại của mã hóa, nếu mã hóa theo phương thức nào thì giải mã phải làm ngược lại phương thức mã hoá đó.

Bài 5d: Xóa tập tin

1. Hãy sửa đổi file BAI\_5D.ASM và lưu với tên BAI\_5D1.ASM để có thể thực hiện được yêu cầu sau: xóa tên một tập tin. Tên tập tin cần được nhập từ bàn phím.

[Hoàn thành]

Bài 5e: Đổi tên tệp

1. Hãy tạo một thư mục con có tên là baitap nằm trong thư mục tt\_asm. Sửa đổi lệnh oldfile db “d:\tt\_asm\data.txt”,0 và newfile db “tt\_asm\solieu.txt”,0 lại thành oldfile db “d:\tt\_asm\solieu.txt”,0 và newfile db “tt\_asm\baitap\data.txt”,0. Biên dịch lại và cho chạy chương trình để xem xét kết quả. Có nhận xét gì về kết quả nhận được.

Kết quả: Tạo được một tệp mới trong folder baitap; Nhận xét: Bản chất của lệnh đổi tên là thay đổi toàn bộ đường dẫn của tệp, nên có thể dùng cho việc di dời (move) tệp nếu cần thiết;

1. Hãy sửa đổi file BAI\_5E.ASM và lưu với tên BAI\_5E1.ASM để có thể thực hiện được yêu cầu sau: đổi tên một tập tin. Tên tập tin cũ và mới được nhập từ bàn phím.

[Hoàn thành]

Bài tập:

1. Viết chương trình sử dụng hàm 41h/ INT 21h để xóa tập tin trên đĩa. Tên tập tin cần xóa được nhập từ bàn phím khi thực hiện chương trình.

[Là bài 5D1]

1. Viết chương trình nhập 1 chuỗi từ bàn phím, sau đó ghép chuỗi nhận được vào cuối của nội dung tập tin có trên đĩa. Tên tập tin nhập từ bàn phím khi chạy chương trình.
2. Viết chương trình nhập 1 chuỗi từ bàn phím, sau đó chèn chuỗi nhận được vào đầu của nội dung tập tin có trên đĩa. Tên tập tin nhập từ bàn phím khi chạy chương trình.

[Là bài 5C4]

1. Viết chương trình ghép nội dung 2 tập tin có sẳn trên đĩa thành 1 tập tin mới. Tên của các tập tin được nhập từ bàn phím khi chạy chương trình.
2. Viết chương trình đọc nội dung tập tin trên đĩa, sau đó đổi tất cả ký tự HOA thành ký tự thường và lưu lại vào tập tin đó. Tên tập tin phải được nhập từ bàn phím

[BAI\_5.5, vẫn còn đang lỗi]