Bài 6a: So sánh chuỗi

1. Trong macro writeln, các lệnh nào có chức năng xuống dòng sau khi in xong chuỗi ký tự.

mov ah,02h  
 mov dl, 0ah ; xuống dòng kèm dấu tab  
 int 21h

mov dl, 0dh; xóa dữ liệu của dòng hiện tại   
 int 21h

1. Mục đích của việc khai báo LOCAL bien1 trong macro là gì?

LOCAL là biến của hàm macro, không phải biến được đưa vào khi gọi hàm;

1. Hãy cho biết địa chỉ của DS và ES có giống nhau hay không? Điều này được thể hiện qua các câu lệnh nào trong đoạn chương trình mẫu? Tại sao người ta không khai báo DS và ES trên các phân đoạn khác nhau ?

DS: SI; địa chỉ chuỗi nguồn

ES: DI; địa chỉ chuỗi đích

1. Tiền tố REPE trong đoạn chương trình mẫu trên có ý nghĩa như thế nào?

repe cmpsb ; so sanh tung ky tu/byte

1. Ta có thể thay đổi lệnh REPE CMPSB thành một nhóm lệnh khác được không? Nếu được hãy thay đổi nó, biên dịch và chạy chương trình để kiểm chứng. <bài lưu là 6A1>
   * Thay vì so sánh cả chuỗi bằng REPE, ta có thể so sánh từng ký tự của chuỗi bằng vòng lặp LOOP với CX là số ký tự/số byte muốn so sánh
   * Đoạn code thay thế REPE:

|  |  |
| --- | --- |
| mov cx,10 ;so ky tu/so byte can so  mov si,0  soTungKyTu:  mov bl,oldpass[si]  mov bh,newpass[si]  CMP bl,bh  jne khongkhop  inc si  LOOP soTungKyTu | ;khong nhay ra 'khongkhop' --> da khop  writeln tbao1  jmp thoat  khongkhop:  writeln tbao2  jmp thoat |

* + Lưu ý: Không thể so sánh CMP trực tiếp phần tử mảng oldpass[si] và newpass[si] mà phải so sánh qua một giá trị đệm (ở đây là BH và BL)

1. Thử thay đổi nội dung ở oldpass và newpass sao cho chúng giống nhau. Biên dịch và chạy chương trình xem kết quả, sau đó hãy giải thích cơ chế làm việc của đoạn lệnh từ lệnh CLD cho đến lệnh REPE CMPSB.

CLD: chọn chiều xử lý chuỗi (khi chạy câu lệnh này, IP tăng một giá trị)

MOV CX,10; LEA SI, OLDPASS; LEA DI,NEWPASS: Ba câu lệnh này đều là gán giá trị cho CX, SI, DI;

REPE CMPSB: chạy vòng lặp LOOP với số vòng là giá trị của CX để so sánh

1. Giả sử, người ta muốn thay thế lệnh cmpsb thành lệnh cmpsw, các bạn có cần sửa đổi các lệnh nào trong chương trình hay không ? Tại sao ?. Biên dịch và chạy chương trình để kiểm chứng.

The size of the source operands is selected with the mnemonic: CMPSB (byte compar-ison), CMPSW (word comparison), CMPSD (doubleword comparison), or CMPSQ (quadword comparison using REX.W).

Với cmpsw, giá trị gán cho CX không còn là số byte nữa mà tính theo công thức sau: <bài 6A2>

|  |
| --- |
| x1:  oldpass dw "0123456789"  newpass dw "0123456789"  size = ($ - x1) / 4  mov cx, size |

1. Lệnh jmp thoat trong đoạn chương trình trên có nhiệm vụ gì? Thử bỏ lệnh jmp thoat sau đó biên dịch và chạy chương trình xem kết quả.

Jmp thoat ở dòng thứ 34 giúp chương trình trực tiếp trỏ đến đoạn lệnh của label thoat. Nếu bỏ lệnh này, chương trình sẽ đi qua cả label intb1 dẫn đến việc in cả hai trường hợp kết quả;

1. Hãy sửa đổi file BAI\_6A.ASM và lưu với tên BAI\_6A1.ASM để có thể thực hiện được nhiệm vụ sau: nhập vào một chuỗi ký tự có tối đa 10 ký tự, trong lúc nhập chỉ hiện thị ra ký tự “\*”. Khi đã nhập đủ 10 ký tự hoặc khi gặp phím ESC thì sẽ in ra các ký tự đã nhập ra màn hình. Gợi ý: dùng hàm 08h, 02h hoặc 09h của int 21h, lệnh loop, cmp, … Cần phải khai báo dùng đệm để lưu các ký tự đã nhập.

<Bài\_6A1\_1.asm>

1. Hãy sửa đổi file BAI\_6A1.ASM, kết hợp với file BAI\_6A.ASM (chương trình mẫu) và lưu với tên BAI\_6A2.ASM để có thể thực hiện được nhiệm vụ sau: nhập vào một chuỗi ký tự có 10 ký tự, trong lúc nhập chỉ hiện thị ra ký tự “\*”. Sau đó so sánh với một oldpass có nội dung tùy ý (nhưng chỉ có độ dài là 10 ký tự mà do chúng ta gán trước, ví dụ như oldpass db “1234567890”). Nếu 10 ký tự vừa nhập có nội dung giống oldpass thì in ra câu thông báo “Ban da nhap dung roi” và thoát, ngược lại thì in ra câu thông báo “Ban da nhap sai roi va vui long nhap lai” và quay trở lại nhập cho đến khi nào đúng mới thoát. Gợi ý: dùng hàm 08h, 02h, 09h của int 21h và các lệnh loop, cmpsb

<Bài 6A1\_2.asm>

Bài 6b: Chuyển nội dung giữa các chuỗi

1. Tại sao chỉ di chuyển 33 ký tự/byte mà lại khai báo biến string2 db 34 dup(‘$”). Thử thay 34 thành 33 và tiến hành biên dịch, chạy chương trình để xem kết quả. Có nhận xét gì về vấn đề này không ?. Giải thích ý nghĩa của việc khai báo này.

Di chuyển 33 ký tự nhưng là từ 0-33 nên về bản chất là 34, vậy nên cần phải khai báo 34dup(‘$’). Nếu thay 34 thành 33 thì chuỗi khi in ra sẽ bị thiếu mất một ký tự.

Giải nghĩa 34 dup (‘$’): tạo ra chuỗi $ dài 34 ký tự ($$$$$...)

1. Thay lệnh CLD trong đoạn chương trình mẫu thành STD. Biên dịch và cho chạy chương trình để kiểm chứng kết quả. Cho nhận xét về kết quả nhận được.

Chương trình hiện lỗi wrong parameters;

1. Nếu thay lệnh movsb thành MOVSW thì chúng ta có phải thay đổi giá trị nào trong đoạn chương trình mẫu trên không? Nếu có thay đổi, hãy biên dịch và chạy chương trình để kiểm chứng lại kết quả.

Nếu thay từ movsB thành movsW thì các biến string1 và string2 cũng cần thay đổi từ dB thành dW

1. Có thể thay thế lệnh REP MOVSB bởi một số lệnh khác hay không? Nếu được thì hãy thay đổi và sau đó biên dịch, cho chạy chương trình để kiểm chứng lại kết quả.
2. Giả sử ta có nội dung của một biến string1 có tổng số byte >256 byte thì lúc đó chúng ta phải khai báo lại các biến này như thế nào ?. Lúc này có khó khăn gì xảy ra không ?. Hãy thử sửa lại, sau đó biên dịch và chạy chương trình để xem kết quả.

256 byte hay 256 ký tự thì tương ứng với 8 bits hay là DB khi khai báo biến. Để tăng giá trị tối đa của mỗi biến, chỉ cần thay đổi khai báo thành DW

1. Có cách nào xác định chiều dài của một biến bất kỳ hay không ?. Hãy cho biết các cú pháp của lệnh có thể thực hiện được yêu cầu này. Sau đó thử áp dụng để xác định chiều dài của biến string1

Bài 6c: Tìm ký tự trong chuỗi

1. Thay lệnh MOV AL, ‘A’ thành MOV AL, ‘B’, sau đó biên dịch và chạy chương trình để xem kết quả.

Kết quả: Chương trình không tìm thấy ký tự

1. Giải thích nhiệm vụ của các lệnh từ CLD cho đến JNE INTB2. Có thể thay thế lệnh REPNE SCASB thành các lệnh khác được không ?. Nếu được hãy thay thế chúng, biên dịch và chạy chương trình để kiểm chứng lại kết quả.

Giải thích:

|  |  |
| --- | --- |
| cld ; chonchieu xu ly chuoi  mov cx, 17 ; so ky tu can tim  mov al, 'A' ; tim kien ky tu A trong string1 | lea di, string1 ; (ES:DI)--> dia chi cua chuoi dich  repne scasb ; lap lai viec tim kiem ky tu cho den  jne intb2 ; khi gap duoc hoac den het chuoi |

REPNE: Repeat following CMPSB, CMPSW, SCASB, SCASW instructions while ZF = 0 (result is Not Equal), maximum CX times. Có thể thay bằng việc so sánh ký tự này với lần lượt từng phần tử mảng trong chuỗi gốc (sử dụng vòng lặp) – Nếu có cờ JE thì ngưng chương trình và thông báo tìm thấy, nếu không, thông báo không tìm thấy ký tự