

Các hàm thao tác chuỗi kí tự

- Mục đích sử dụng
- Các hàm copy, nối chuỗi
- Các hàm so sánh chuỗi
- Ví dụ minh họa

Mục đích sử dụng

- Thư viện `<string.h>` cung cấp các hàm thao tác với chuỗi kí tự như sao chép, so sánh, nối chuỗi, tìm kiếm, phân tách và lấy độ dài chuỗi
- Tìm hiểu và sử dụng các hàm này sẽ giúp quá trình xử lý vấn đề trở nên nhanh gọn đơn giản hơn khi ta thao tác với chuỗi kí tự
- Luôn nhớ `include<string.h>` trước khi sử dụng những hàm này

Các hàm copy, nối và lấy độ dài chuỗi kí tự

- Sau đây là các hàm và mô tả chi tiết:

Hàm nguyên mẫu và mô tả
<code>char * strcpy(char *s1, const char *s2):</code> Copy chuỗi s2 vào chuỗi s1. Giá trị của s1 sau đó sẽ được return
<code>char *strncpy(char *s1, const char *s2, size_t n):</code> Copy tối đa n kí tự từ s2 vào s1 và trả về s1
<code>char *strcat(char *s1, const char *s2):</code> Nối chuỗi s2 vào chuỗi s1, kí tự đầu của s2 sẽ thế chỗ kí tự null của s1 và trả về s1
<code>char *strncat(char *s1, const char *s2, size_t n):</code> Lấy tối đa n kí tự từ chuỗi s2 thêm vào cuối chuỗi s1. Kí tự đầu tiên của s2 sẽ thế chỗ kí tự null của s1. Giá trị của s1 sẽ được trả về
<code>size_t strlen(const char *s):</code> Xác định độ dài chuỗi s. Tổng số kí tự trước kí tự kết thúc chuỗi '\0' sẽ được trả về.

- Chú ý rằng hàm strncpy sẽ không copy kí tự null vào kết quả nếu số kí tự cần copy nhỏ hơn độ dài chuỗi ta cần copy

Ví dụ sử dụng hàm copy

- Mảng chứa kết quả phải đủ lớn để chứa cả nội dung cũ và mới sau khi copy

```
const char* msg = "Welcome to C programming language!";  
char s1[40];  
char s2[20];
```

```
// kiểm tra độ dài chuỗi msg:  
printf("Độ dài chuỗi msg là: %u\n", strlen(msg));
```

```
// copy msg vào chuỗi s1:  
strcpy(s1, msg);
```

```
// copy 10 kí tự của msg vào s2:  
strncpy(s2, msg, 10);
```

Độ dài chuỗi msg là: 34

Chuỗi s1: Welcome to C programming language!

Chuỗi s2: Welcome to Welcome to C programming language!

Ví dụ sử dụng hàm nối chuỗi

- Mảng chứa kết quả phải đủ lớn để chứa cả nội dung cũ và mới sau khi nối

```
char s1[40] = "Good morning sir! ";  
char s2[20] = "I'm student ";  
char s3[40] = "Good morning sir! ";  
// noi msg vao cuoi chuoai s1:  
strcat(s1, s2);  
  
// copy 10 ki tu cua msg vao s2:  
strncat(s3, s2, 6);
```

- Kết quả

```
Chuoai s1: Good morning sir! I'm student  
Chuoai s3: Good morning sir! I'm st
```

Các hàm so sánh

- Sau đây là các hàm so sánh chuỗi kí tự

Hàm nguyên mẫu và mô tả
int strcmp(const char *s1, const char *s2): So sánh s1 với s2, trả về các giá trị 0, nhỏ hơn 0, lớn hơn 0 tương ứng nếu s1 tương đương s2, nhỏ hơn và lớn hơn s2
int strncmp(const char *s1, const char *s2, size_t n): So sánh n kí tự đầu tiên của s1 và s2. Kết quả trả về tương ứng là giá trị nhỏ hơn 0, bằng 0 hoặc lớn hơn 0 tương ứng với trường hợp n kí tự của hai chuỗi này s1 nhỏ hơn, tương đương hoặc lớn hơn s2

Ví dụ sử dụng hàm so sánh

- Ví dụ:

```
char s1[20] = "I'm Iron man"; // Iron man said
char s2[20] = "I'm student"; // Student said
char s3[40] = "I'm inevitable"; // Thanos said
```

```
// so sanh chuoì s1 voi s2
printf("strcmp(s1, s2) = %d\n", strcmp(s1, s2));
```

```
// so sanh s1 voi chinh no
printf("strcmp(s1, s1) = %d\n", strcmp(s1, s1));
```

```
// so sanh 4 ki tu cua s1 va s3
printf("strncmp(s1, s3, 4) = %d\n", strncmp(s1, s3, 4));
```

- Kết quả:

```
strcmp(s1, s2) = -1
strcmp(s1, s1) = 0
strncmp(s1, s3, 4) = 0
```

Tiếp theo

Các hàm tìm kiếm, tách chuỗi kí tự