

Các hàm tìm kiếm và tách chuỗi

- Mục đích sử dụng
- Mô tả các hàm
- Ví dụ minh họa



Mục đích sử dụng

- Các hàm tìm kiếm cho phép ta tìm kiếm sự xuất hiện của kí tự trong chuỗi, chuỗi trong chuỗi
- Hàm tách chuỗi cho phép ta phân tách nội dung của một chuỗi ra thành các phần theo dấu hiệu phân tách nào đó

Mô tả các hàm

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

Hàm nguyên mẫu và mô tả
<code>char *strchr(const char *s, int c):</code> Xác định vị trí xuất hiện đầu tiên của kí tự c trong string s. Nếu tìm thấy, một con trỏ trỏ đến c trong s sẽ trả về, ngược lại ta nhận được con trỏ NULL
<code>size_t strcspn(const char *s1, const char *s2):</code> Xác định và trả về độ dài của phần đầu trong chuỗi s1 không chứa kí tự có trong s2
<code>size_t strspn(const char *s1, const char *s2):</code> Xác định và trả về độ dài của phần đầu trong chuỗi s1 chỉ chứa các kí tự bao gồm trong s2
<code>char *strpbrk(const char *s1, const char *s2):</code> Xác định vị trí xuất hiện đầu tiên của bất kì kí tự nào trong chuỗi s2. Nếu một kí tự của s2 được tìm thấy, một con trỏ trỏ tới kí tự đó trong s1 sẽ được trả về. Nếu không có kết quả sẽ return NULL
<code>char *strrchr(const char *s, int n):</code> Xác định vị trí xuất hiện cuối cùng của n trong s. Nếu có, một con trỏ trỏ tới kí tự n trong s sẽ được trả về. Nếu không có kết quả sẽ return NULL
<code>char *strstr(const char *s1, const char *s2):</code> Tìm vị trí xuất hiện đầu tiên của s2 trong s1. Nếu tìm thấy, một con trỏ trỏ đến string đó trong s1 sẽ được return. Ngược lại con trỏ NULL sẽ được return
<code>char *strtok(char *s1, const char *s2):</code> Dùng để tách chuỗi s1 ra thành các mẫu khác nhau theo các dấu hiệu phân tách cho trong s2. Thường dùng để tách từ hoặc tách chuỗi theo yêu cầu nào đó. Lời gọi đến hàm này thường là một loạt các lời gọi để phân tách chuỗi ban đầu đến NULL thì dừng. Lần gọi đầu tiên sẽ cung cấp s1 và các dấu hiệu phân tách trong s2. Từ lần gọi kế tiếp tham số đầu tiên là NULL nếu vẫn tiếp tục tách phần còn lại của chuỗi gốc, tham số thứ hai giữ nguyên.

Ví dụ sử dụng hàm strstr

- Ví dụ sau tìm kiếm chuỗi s2 trong s1

```
#include <stdio.h>
#include <string.h>

int main() {
    char s1[40] = "Meow meow meow ouch ouch ouch...";
    const char *s2 = "ouch";

    char* result = strstr(s1, s2);
    if (result != NULL) {
        printf("Vi tri xuất hiện đầu tiên của \"%s\" trong s1:\n",
s2, result);
    }
    else {
        puts("Không tìm thấy sự xuất hiện của s2 trong s1.\n");
    }

    return 0;
}
```

Ví dụ sử dụng hàm strtok

- Ví dụ sau tách chuỗi tại vị trí dấu cách

```
char s1[40] = "Meow meow meow ouch ouch ouch...";  
const char* delim = " ";
```

```
char* result = strtok(s1, delim);  
int i = 1;  
while (result != NULL) {  
    printf("Tu thu %d: %s\n", i++, result);  
    // tiếp tục tách phần còn lại chuỗi ban đầu  
    result = strtok(NULL, delim);  
}
```

```
Tu thu 1: Meow  
Tu thu 2: meow  
Tu thu 3: meow  
Tu thu 4: ouch  
Tu thu 5: ouch  
Tu thu 6: ouch...
```

Hàm memset

- Sau đây là hàm reset các giá trị trong chuỗi

`void *memset(void *s, int c, size_t n)`: copy kí tự trong c vào n byte đầu tiên của đối tượng được trỏ đến bởi s. Một con trỏ trỏ đến kết quả được trả về.

- Ví dụ:

```
char s1[] = "HHHHHHHHHHHHHHHHHHHHHHHHHHHH";  
char c = 'a';  
printf("Chuoi goc ban dau: %s\n", s1);  
memset(s1, c, 10);  
printf("Chuoi s1 sau khi copy: %s\n", s1);
```

Tiếp theo

Các hàm thao tác với kí tự đơn