

CHƯƠNG 5

Kỹ thuật xử lý chuỗi ký tự

Design by Minh An

Email: anvanminh.hau@gmail.com

Nội dung

1. Khái niệm chuỗi.
2. Biến chuỗi và hằng chuỗi.
3. Nhập/xuất chuỗi.
4. Các hàm xử lý chuỗi.
5. Kỹ thuật xử lý chuỗi

Design by Minh An

5.1. Khái niệm chuỗi

- ❖ Chuỗi là mảng ký tự kết thúc bởi ký tự `null` (`'\0'`).
- ❖ Khai báo chuỗi theo cú pháp:

```
char tên_chuỗi[N]; // N là kích thước chuỗi
```

- ❖ Ví dụ:

```
char str[15];
```

```
// Chuỗi str chứa được tối đa 15 ký tự bao gồm cả ký tự '\0'
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
str														

- ❖ Khi khai báo một chuỗi, hãy dành thêm một phần tử cho ký tự `'\0'`.

Design by Minh An

5.2. Biến chuỗi và hằng chuỗi

- ❖ Hằng chuỗi là tập các ký tự nằm trong dấu nháy kép.
- ❖ Ví dụ: `"Công nghiệp"` là một hằng chuỗi.
- ❖ Biến chuỗi là mảng kiểu ký tự để chứa 1 hằng chuỗi.
- ❖ `str` là biến chuỗi, `str` chứa hằng chuỗi `"Công nghiệp"` như dưới đây.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
str	C	o	n	g		n	g	h	i	e	p	\0		

- ❖ Ký tự `'\0'` được tự động thêm vào biểu diễn bên trong của chuỗi.

Design by Minh An

5.3. Nhập xuất chuỗi

- ❖ Sử dụng các hàm trong thư viện nhập/xuất chuẩn `stdio.h` để thực hiện các thao tác nhập/xuất chuỗi.
- ❖ Hàm `gets()` được sử dụng để nhập vào một chuỗi thông qua thiết bị nhập chuẩn.
- ❖ Các ký tự được nhập vào cho đến khi bấm enter.
- ❖ Hàm `gets()` tự động thay thế ký tự sang dòng mới `'\n'` bằng ký tự `'\0'`.
- ❖ Cú pháp:

```
gets(biến_chuỗi);
```

- ❖ Ví dụ:

```
gets(str);
```

Design by Minh An

Nhập xuất chuỗi (tt)

- ❖ Hàm `puts()` được dùng để hiển thị một chuỗi trên thiết bị xuất chuẩn.
 - ❖ Cú pháp:
- ```
puts(chuoi);
```
- ❖ Ví dụ:
- ```
puts("Ha Noi");  
puts(str);
```
- ❖ Lệnh `cin>>` được sử dụng để nhập các chuỗi không chứa dấu cách: một từ, một chuỗi số ...
- ```
cin>>str;
```
- ❖ Lệnh `cout<<` được dùng để hiển thị chuỗi.
- ```
cout<<str;
```

Design by Minh An

5.4. Các hàm xử lý chuỗi

- ❖ Các hàm xử lý chuỗi được định nghĩa trong thư viện `string.h`.
- ❖ Các phép xử lý chuỗi gồm:
 - ✓ Tính chiều dài thực của chuỗi.
 - ✓ Gán chuỗi.
 - ✓ So sánh hai chuỗi.
 - ✓ Ghép hai chuỗi.
 - ✓ Tìm vị trí ký tự trong chuỗi v.v...

Design by Minh An

5.3.1. Hàm `strlen()`

- ❖ Tính độ dài thực của chuỗi.
- ❖ Cú pháp:

```
int strlen(const char *str);
```
- ❖ Hàm trả về một giá trị nguyên là độ dài thực của chuỗi `str` với `str` là một hằng chuỗi hoặc một biến chuỗi.
- ❖ Ví dụ:

```
int n = strlen("Cong nghiep");  
//Ta nhận được n = 11.  
char *str = "Ha Noi";  
int m = strlen(str);  
//Ta nhận được m = 6.
```

Design by Minh An

5.3.2. Hàm `strcpy()`

- ❖ Sao chép nội dung của một biến chuỗi hay một hằng chuỗi vào một biến chuỗi khác.
- ❖ Cú pháp:

```
strcpy(str1, str2);
```

//Nội dung cũ của str1 bị xóa
//Nội dung của str2 được sao chép sang str1
//str1 phải là một biến chuỗi,
//str2 có thể là hằng hoặc biến chuỗi
- ❖ Ví dụ:

```
strcpy(str, "Ba Dinh");  
//str chứa nội dung "Ba Dinh"
```

Design by Minh An

5.3.3. Hàm `strcmp()`

- ❖ So sánh hai chuỗi và trả về một giá trị số nguyên dựa trên kết quả của sự so sánh.
- ❖ Cú pháp:

```
strcmp(str1, str2);
```
- ❖ Hàm trả về giá trị nguyên:
 - ✓ Nhỏ hơn 0, nếu `str1 < str2`
 - ✓ Bằng 0, nếu `str1` giống `str2`
 - ✓ Lớn hơn 0, nếu `str1 > str2`
- ❖ Theo thứ tự từ điển chuỗi đứng sau là chuỗi lớn hơn.
- ❖ Ví dụ:

```
k = strcmp("Hai Phong", "Hai Duong");  
//Ta có k = 12 (hoặc 1)
```

Design by Minh An

5.3.4. Hàm `strcat()`

- ❖ Nối hai giá trị của hai chuỗi vào một chuỗi.
- ❖ Cú pháp:

```
strcat(str1, str2);
```

//Nối str2 vào cuối chuỗi str1
//str1 phải là một biến chuỗi
- ❖ Ví dụ:

```
strcpy(str, "Ma Van");  
//str chứa "Ma Van"  
strcat(str, " Khang");  
//str chứa "Ma Van Khang"
```

Design by Minh An

5.3.5. Hàm `strchr()`

- ❖ Xác định vị trí xuất hiện của một ký tự trong một chuỗi.
- ❖ Cú pháp:

```
strchr(str, chr);
```
- ❖ Hàm trả về:
 - ✓ Con trỏ trỏ đến vị trí tìm được đầu tiên của ký tự `chr` trong chuỗi `str`.
 - ✓ `NULL` nếu `chr` không có trong chuỗi `str`.

Design by Minh An

5.4. Kỹ thuật xử lý chuỗi

1. Cài đặt chương trình thực hiện:

- ❖ Nhập vào hai chuỗi str1 và str2 chiều dài mỗi chuỗi không quá 31 ký tự.
- ❖ Hãy cho biết:
 - ✓ Hai chuỗi có giống nhau hay không nếu phân biệt chữ hoa và chữ thường.
 - ✓ Hai chuỗi có giống nhau hay không nếu không phân biệt chữ hoa và chữ thường.
 - ✓ Hai chuỗi có giống nhau hay không nếu chỉ tính 3 ký tự đầu tiên.

Design by Minh An

Kỹ thuật xử lý chuỗi (tt)

2. Cài đặt chương trình thực hiện:

- ❖ Nhập chuỗi str không quá 100 ký tự.
- ❖ Hãy cho biết chuỗi str có bao nhiêu chữ cái in, bao nhiêu chữ cái thường, bao nhiêu chữ số.
- ❖ Tạo một chuỗi mới str1 là đảo ngược của chuỗi str, hiển thị chuỗi str1.
- ❖ Thay thế tất cả các chữ số trong chuỗi str bằng từ đọc nó (ví dụ 9 thay bằng chín), hiển thị lại chuỗi str.

Design by Minh An

Kỹ thuật xử lý chuỗi (tt)

3. Cài đặt chương trình thực hiện:

- ❖ Nhập vào một đoạn văn bản bằng tiếng anh không quá 255 ký tự.
- ❖ Thay thế tất cả các từ child (nếu có) trong đoạn văn bản bằng từ children.
- ❖ Hiển thị chuỗi sau khi thay thế.

4. Cài đặt chương trình thực hiện:

- ❖ Nhập vào một đoạn văn bản bất kỳ.
- ❖ Chỉ bằng một lần duyệt hãy cho biết đoạn văn bản có bao nhiêu từ. Biết rằng từ là một dãy liên tiếp các chữ cái, chữ số.

Design by Minh An

Kỹ thuật xử lý chuỗi (tt)

5. Viết chương trình thực hiện các yêu cầu:

- ❖ Nhập vào một đoạn văn bản bất kỳ không quá 255 ký tự.
- ❖ Hãy chuẩn hóa đoạn văn bản theo các yêu cầu sau:
 - ✓ Không còn các dấu cách ở hai đầu.
 - ✓ Giữa hai từ chỉ cách nhau đúng một dấu cách.
 - ✓ Trước dấu chấm, dấu phẩy không có dấu cách, sau dấu chấm, dấu phẩy có đúng một dấu cách.
 - ✓ Chữ cái đầu đoạn văn bản và sau dấu chấm viết hoa, các chữ khác viết thường.
- ❖ Hiển thị đoạn văn bản sau khi chuẩn hóa.

Design by Minh An

Thank you...!

Design by Minh An