CHƯƠNG 5

Kỹ thuật xử lý chuỗi ký tự

Giảng viên: Nguyễn Chiến Thắng
Email: thangnc.haui@gmail.com

Design by thangnc

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

5.1. Khái niệm chuỗi ký tự

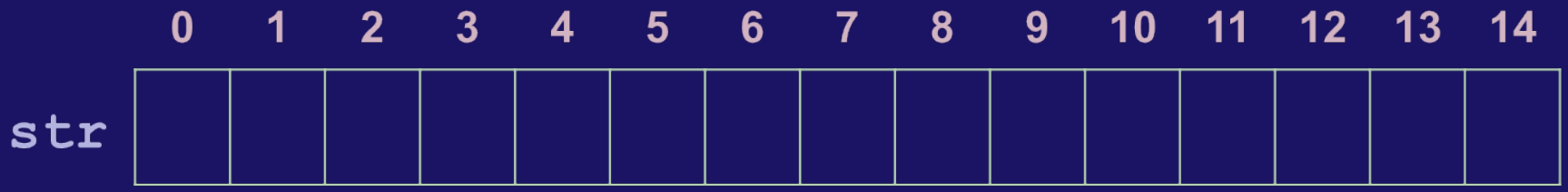
- Chuỗi là mảng ký tự kết thúc bởi ký tự `null` (`'\0'`).
- Khai báo chuỗi theo cú pháp:

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

- Ví dụ:

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

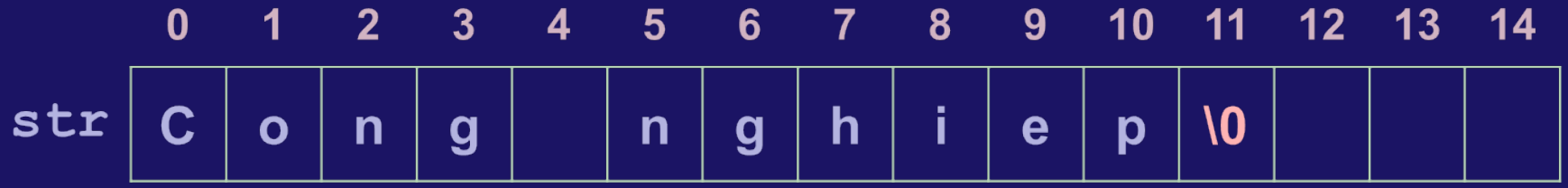
//Chuỗi str chứa được tối đa 15 ký tự bao gồm cả ký tự '\0'



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

5.2. Biến chuỗi và hằng chuỗi

- Hằng chuỗi là tập các ký tự nằm trong dấu nháy kép, hằng chuỗi là dữ liệu cần được lưu trữ.
- Ví dụ: "Cong nghiep" là một hằng chuỗi.
- Biến chuỗi là mảng kiểu ký tự để chứa 1 hằng chuỗi.
- **str** là biến chuỗi, **str** chứa hằng chuỗi "Cong nghiep" như dưới đây.



- Ký tự '\0' (đọc là null) được tự động thêm vào biểu diễn bên trong của chuỗi.

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);
```

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);
```

- Hàm `scanf(...)` được sử dụng để nhập các chuỗi không chứa dấu cách: một từ, một chuỗi số ...

```
scanf("%s", str); //không có & trước str
```

- Hàm `pritnf(…)` được dùng để hiển thị chuỗi.

```
printf("%s", str);
```

5.4. Một số hàm xử lý chuỗi thông dụng

- 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…

5.4.1. 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 thangnc

5.4.2. 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 thangnc

5.4.3. 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)
```

5.4.4. 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 chua "Ma Van Khang"
```

5.4.5. 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.*

5.4.6. Một số hàm khác

- `char *strrchr(char *str, char chr)`: tìm vị trí xuất hiện cuối cùng của ký tự chr trong chuỗi str.
- `char *strstr(char *str, char *s)`: tìm vị trí xuất hiện của chuỗi s trong chuỗi str.
- `char *strlwr(char *str)`: chuyển các chữ cái in trong chuỗi str thành chữ thường.
- `char *strupr(char *str)`: chuyển các chữ cái thường trong chuỗi str thành chữ in.
- `char *strrev(char *str)`: đảo ngược chuỗi str.
- `int strcmpi(char *str1, char *str2)`: so sánh không phân biệt chữ hoa, chữ thường.

5.5. Mảng các chuỗi ký tự

- Là mảng mà mỗi phần tử lưu trữ một chuỗi ký tự.
- Khai báo: `char str[num][length];`
- Ví dụ:

```
char name_list[7][9] = {"Trang", "Cong",
"Thuong","Nguyen","Tien","Binh","Cuong"};
```

	0	1	2	3	4	5	6	7	8
0	T	r	a	n	g	\0			
1	C	o	n	g	\0				
2	T	h	u	o	n	g	\0		
3	N	g	u	y	e	n	\0		
4	T	i	e	n	\0				
5	B	i	n	h	\0				
6	C	u	o	n	g	\0			

5.6. Kỹ thuật xử lý chuỗi

1. Cài đặt chương trình thực hiện:
   - Nhập vào họ và tên của hai học sinh là ht1, ht2.
   - Hãy cho biết:
     - ✓ Cho biết ht1, ht2 có giống nhau hay không nếu phân biệt chữ hoa và chữ thường?
     - ✓ Cho biết ht1, ht2 có giống nhau hay không nếu không phân biệt chữ hoa và chữ thường?
     - ✓ Cho biết ht1, ht2 có giống nhau hay không nếu chỉ tính 3 ký tự đầu tiên?
     - ✓ Cho biết ht1 có xuất hiện trong ht không?

Kỹ thuật xử lý chuỗi (tt)

2. Cài đặt chương trình thực hiện:
   - Cho phép nhập vào một xâu ký tự bất kỳ.
   - Cho biết xâu vừa nhập có bao nhiêu chữ cái thường?
   - Xóa mọi ký tự ‘a’ ra khỏi xâu vừa nhập và in kết quả ra màn hình.
3. Cài đặt chương trình thực hiện:

   Nhập một xâu ký tự có độ dài không quá 50 ký tự từ bàn phím. Một xâu ký tự được gọi là hợp lệ nếu nó không chứa hai dấu cách liền nhau và có không quá 10 từ (một từ được định nghĩa là cụm ký tự liên tiếp, dài nhất không chứa dấu cách). Hãy kiểm tra xem xâu vừa nhập có hợp lệ không.

Design by thangnc

Kỹ thuật xử lý chuỗi (tt)

4. Cài đặt chương trình thực hiện:
   - Nhập một xâu ký tự có độ dài không quá 80 ký tự từ bàn phím.
   - Nhập từ bàn phím một ký tự vào biến C và một số nguyên k (k thuộc [1..n+1]).
   - Hãy chèn ký tự C vào vị trí k trong xâu và in xâu kết quả ra màn hình.

Kỹ thuật xử lý chuỗi (tt)

5. Cài đặt chương trình thực hiện:
   - Cho một xâu ký tự gồm toàn các dấu mở/ đóng ngoặc '(' và ')'. Xâu được gọi là hợp lệ nếu các dấu mở/ đóng ngoặc được đặt phù hợp như khi nó đặt trong biểu thức toán học.
   - Ví dụ: (( )( )) hoặc ((( )))( ) là hợp lệ, xâu)( )) hoặc ((( ))…là không hợp lệ.
   - Hãy cho biết xâu vừa nhập có hợp lệ không?

Design by thangnc

Thank you!