

TIN HỌC CƠ SỞ 2

Xâu ký tự

Nội dung

- Khái niệm
- Khai báo xâu
- Khởi tạo xâu
- Hàm nhập xâu
- Một số hàm thao tác xâu

Khái niệm

- Xâu được xem như một mảng gồm các phần tử kiểu char.
- Ký hiệu kết thúc xâu được quy ước là ký tự ‘\0’ (ký tự có mã ASCII là 0) đặt ở cuối xâu. Như vậy một mảng ký tự gồm n phần tử sẽ lưu được tối đa $n - 1$ ký tự.

0	1	2	3	4	5
H	e	l	l	o	\0

Khai báo chuỗi

Cú pháp:

```
char <Tên chuỗi>[<Kích thước>];
```

Ví dụ:

```
char s[10];
```

Khai báo một chuỗi s chứa tối đa được 9 ký tự. Khi đó trình biên dịch sẽ cấp phát một vùng nhớ 10 byte cho chuỗi s.

Khai báo xâu

```
char *<Tên biến>;
```

Ví dụ: char *str;

Trong khai báo này, bộ nhớ sẽ dành 2 byte để lưu trữ địa chỉ của biến con trỏ str đang chỉ đến, chưa cung cấp nơi để lưu trữ dữ liệu.

Trước khi sử dụng phải dùng từ khóa **new** để cấp phát vùng nhớ.

Ví dụ:char *str;

str = new char[51]; //Cấp phát 51 ký tự

Khai báo chuỗi

Xâu ký tự giống như mảng do đó để khởi tạo một xâu ký tự với những giá trị xác định ta có thể thực hiện tương tự như với mảng.

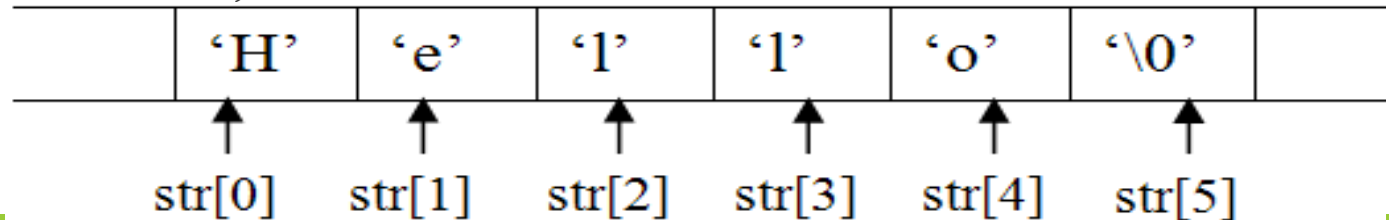
```
char <Biến>[ ]=<“Hàng Xâu”>
```

Ví dụ:

```
char str[] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

```
char str[] = “Hello”;
```

```
char *str = “Hello”;
```



Hàm nhập sâu

Khái niệm về **stdin**

stdin là một vùng nhớ đệm dùng để lưu dữ liệu từ bàn phím. Các hàm `scanf`, `gets` và `getchar` đều nhận dữ liệu từ **stdin**. Cần phân biệt hai trường hợp:

- Nếu trên **stdin** có đủ dữ liệu thì các hàm trên sẽ nhận một phần dữ liệu mà chúng yêu cầu. Phần dữ liệu còn lại (chưa được nhận) vẫn ở trên **stdin**.
- Nếu trên **stdin** không đủ dữ liệu theo yêu cầu của các hàm, thì máy tạm dừng để người dùng đưa thêm dữ liệu từ bàn phím lên **stdin** (cho đến khi nhấn phím Enter)

Hàm nhập xâu

Hàm nhập xâu

char *gets(char *s);

Nhập một dãy ký tự từ stdin cho đến khi gặp ‘\n’ (ký tự ‘\n’ bị loại khỏi stdin). Dãy ký tự được bổ sung thêm ký tự ‘\0’ và đặt vào vùng nhớ do s trỏ đến. Hàm trả về con trỏ đến xâu nhận được.

Chú ý: Hàm scanf có để lại ký tự ‘\n’ trên stdin, ký tự này sẽ làm trôi hàm gets (nếu có) sau đó. Để hàm gets không bị trôi thì phải khử ký tự ‘\n’ trong stdin trước khi gọi hàm gets bằng cách dùng hàm fflush(stdin) để làm sạch stdin.

Ví dụ:

```
void main()
{
    int tuoi;
    char ten[31];
    printf("Nhap tuoi:");
    scanf("%d", &tuoi);
    printf("Nhap ten:");
    fflush(stdin); //Làm sạch stdin
    gets(ten);
}
```

Một số hàm thao tác chuỗi

Các hàm sau đây được định nghĩa sẵn trong thư viện <string.h>

int strlen(char *s);

Trả về độ dài chuỗi s.

char *strcat(char *s1, char *s2);

Ghép s2 vào s1 và trả về con trỏ đến s1.

char *strcpy(char *s1, char *s2);

Chép s2 đè lên s1 và trả về con trỏ đến s1.

*int strcmp(char *s1, char *s2);*

So sánh s1 và s2. Hàm trả về giá trị âm nếu s1 nhỏ hơn s2, giá trị 0 nếu s1 bằng hơn s2 và giá trị dương nếu s1 lớn hơn s2.

*int stricmp(char *s1, char *s2);*

Hàm làm việc tương tự như strcmp nhưng không phân biệt chữ hoa với chữ thường.

*char *strchr(char *s, char c);*

Tìm sự xuất hiện đầu tiên của c trong s. Nếu tìm thấy hàm trả về con trỏ đến ký tự tìm được trong s, nếu không hàm trả về giá trị NULL.

*char *strstr(char *s1, char *s2);*

Tìm sự xuất hiện đầu tiên của s2 trong s1. Nếu tìm thấy hàm trả về con trỏ đến đầu con trỏ tìm được trong s1, nếu không hàm trả về giá trị NULL.