



Trường Đại học Khoa học Tự nhiên  
Khoa Công nghệ thông tin  
Bộ môn Công nghệ phần mềm

# NHẬP MÔN LẬP TRÌNH

ThS. Đặng Bình Phương  
dbphuong@fit.hcmus.edu.vn

## CHUỖI KÝ TỰ





# Nội dung

1

**Khái niệm**

2

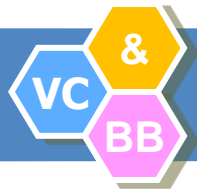
**Khởi tạo**

3

**Các thao tác trên chuỗi ký tự**

4

**Bài tập**



# Khái niệm

## ❖ Khái niệm

- Kiểu **char** chỉ chứa được một ký tự. Để lưu trữ một chuỗi (nhiều ký tự) ta sử dụng mảng (một chiều) các ký tự.
- Chuỗi ký tự kết thúc bằng ký tự '**\0**' (null)  
→ Độ dài chuỗi = kích thước mảng – 1

## ❖ Ví dụ

```
char szHoten[30];           // Dài 29 ký tự  
char szNgaySinh[9];        // Dài 8 ký tự
```

## ❖ Khởi tạo như mảng thông thường

### ■ Độ dài cụ thể

```
char s[10] = {'T', 'H', 'C', 'S', ' ', 'A', '\0'};
```

```
char s[10] = "THCS A"; // Tự động thêm '\0'
```

0 1 2 3 4 5 6 7 8 9

'T' 'H' 'C' 'S' ' ' 'A' '\0' [ ] [ ] [ ]

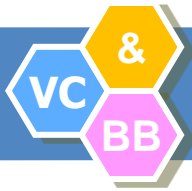
### ■ Tự xác định độ dài

```
char s[] = {'T', 'H', 'C', 'S', ' ', 'A', '\0'};
```

```
char s[] = "THCS A"; // Tự động thêm '\0'
```

0 1 2 3 4 5 6

'T' 'H' 'C' 'S' ' ' 'A' '\0'



# Xuất chuỗi

## ❖ Sử dụng hàm printf với đặc tả “%s”

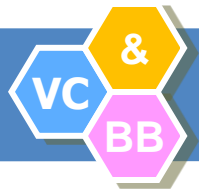
```
char szMonHoc[50] = “Tin hoc co so A”;  
printf(“%s”, szMonHoc);    // Không xuống dòng
```

```
Nhap mon lap trinh_  
_
```

## ❖ Sử dụng hàm puts

```
char szMonHoc[50] = “Tin hoc co so A”;  
puts(szMonHoc);           // Tự động xuống dòng  
⇔ printf(“%s\n”, szMonHoc);
```

```
Nhap mon lap trinh  
_
```



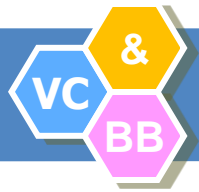
# Nhập chuỗi

## ❖ Sử dụng hàm scanf với đặc tả “%s”

- Chỉ nhận các ký tự từ bàn phím đến khi gặp ký tự khoảng trắng hoặc ký tự xuống dòng.
- Chuỗi nhận được không bao gồm ký tự khoảng trắng và xuống dòng.

```
char szMonHoc[50];  
printf("Nhap mot chuoai: ");  
scanf("%s", szMonHoc);  
printf("Chuoi nhan duoc la: %s", szMonHoc);
```

```
Nhap mot chuoai: Nhap mon lap trinh  
Chuoi nhan duoc la: Nhap_
```



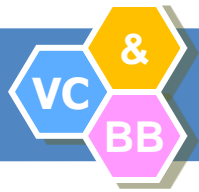
# Nhập chuỗi

## ❖ Sử dụng hàm gets

- Nhận các ký tự từ bàn phím đến khi gặp ký tự xuống dòng.
- Chuỗi nhận được là những gì người dùng nhập (trừ ký tự xuống dòng).

```
char szMonHoc[50];  
printf("Nhap mot chuoai: ");  
gets(szMonHoc);  
printf("Chuoai nhan duoc la: %s", szMonHoc);
```

```
Nhap mot chuoai: Nhap mon lap trinh  
Chuoai nhan duoc la: Nhap mon lap trinh_
```



# Một số hàm thao tác trên chuỗi

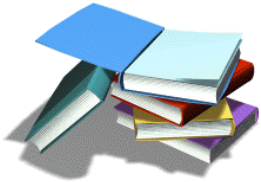
## ❖ Thuộc thư viện `<string.h>`

- `strlen`
- `strcpy`
- `strdup`
- `strlwr/strupr`
- `strrev`
- `strcmp/stricmp`
- `strcat`
- `strstr`



# Hàm tính độ dài chuỗi

`size_t strlen(const char *s)`



Tính độ dài chuỗi **s**.

**size\_t** thay cho unsigned (trong `<stddef.h>`)  
dùng để đo các đại lượng không dấu.



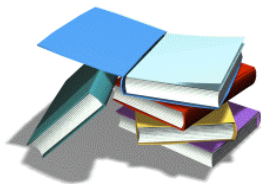
◆ Độ dài chuỗi **s** (không tính ký tự kết thúc)



```
char s[] = "Visual C++ 6.0";  
int len = strlen(s);      // => 14
```

# Hàm sao chép chuỗi

`char *strcpy(char *dest, const char *src)`



Sao chép chuỗi **src** sang chuỗi **dest**, dừng khi ký tự kết thúc chuỗi **'\0'** vừa được chép.  
**!** dest phải đủ lớn để chứa src

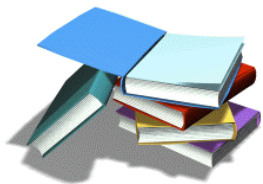


◆ Con trỏ **dest**.



```
char s[100];  
s = "Visual C++ 6.0";           // sai  
strcpy(s, "Visual C++ 6.0");    // đúng
```

`char *strdup(const char *s)`



Tạo bản sao của một chuỗi **s** cho trước. Hàm sẽ tự tạo vùng nhớ dài `strlen(s) + 1` (bytes) để chứa chuỗi **s**. Phải tự hủy vùng nhớ này khi không sử dụng nữa.



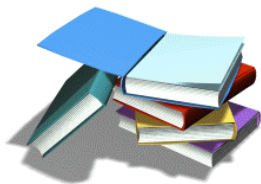
- ◆ **Thành công**: trả về con trỏ đến vùng nhớ chứa chuỗi bản sao.
- ◆ **Thất bại**: trả về **NULL**.



```
char *s;  
s = strdup("Visual C++ 6.0");
```

# Hàm chuyển thành chuỗi thường

`char *strlwr(char *s)`



Chuyển chuỗi **s** thành chuỗi thường ('A' thành 'a', 'B' thành 'b', ..., 'Z' thành 'z')



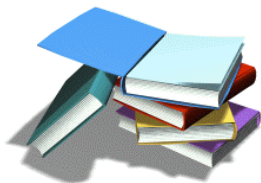
◆ Con trỏ đến chuỗi **s**.



```
char s[] = "Visual C++ 6.0";  
strlwr(s);  
puts(s);           // visual c++ 6.0
```

# Hàm chuyển thành chuỗi IN

`char *strupr(char *s)`



Chuyển chuỗi **s** thành chuỗi IN ('a' thành 'A', 'b' thành 'B', ..., 'z' thành 'Z')

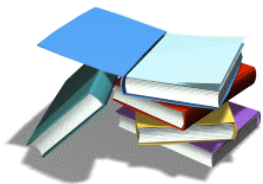


◆ Con trỏ đến chuỗi **s**.



```
char s[] = "Visual C++ 6.0";  
strupr(s);  
puts(s);           // VISUAL C++ 6.0
```

`char *strrev(char *s)`



Đảo ngược thứ tự các ký tự trong chuỗi **s** (trừ ký tự kết thúc chuỗi).



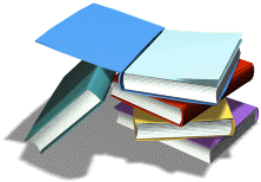
◆ Con trỏ đến chuỗi kết quả.



```
char s[] = "Visual C++ 6.0";  
strrev(s);  
puts(s);           // 0.6 ++C lausiv
```

# Hàm so sánh hai chuỗi

int **strcmp**(const char \***s1**, const char \***s2**)



So sánh hai chuỗi **s1** và **s2** (phân biệt hoa thường).



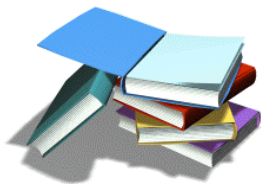
- ◆ < 0 nếu  $s1 < s2$
- ◆ == 0 nếu  $s1 == s2$
- ◆ > 0 nếu  $s1 > s2$



```
char s1[] = "visual C++ 6.0";  
char s2[] = "Visual C++ 6.0";  
int kq = strcmp(s1, s2); // => kq > 0
```

# Hàm so sánh hai chuỗi

int **stricmp**(const char \***s1**, const char \***s2**)



So sánh hai chuỗi **s1** và **s2** (không phân biệt hoa thường).



- ◆ < 0 nếu  $s1 < s2$
- ◆ == 0 nếu  $s1 == s2$
- ◆ > 0 nếu  $s1 > s2$

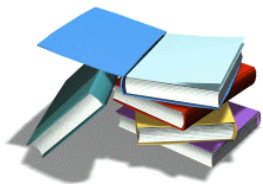


```
char s1[] = "visual c++ 6.0";  
char s2[] = "VISUAL C++ 6.0";  
int kq = stricmp(s1, s2); // => kq == 0
```



# Hàm nối hai chuỗi

`char* strcat(char *dest, const char *src)`



Nối chuỗi **src** vào sau chuỗi **dest**.  
! Chuỗi **dest** phải đủ chứa kết quả



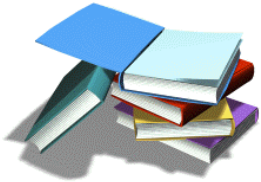
◆ Con trỏ đến chuỗi được nối.



```
char s1[100] = "Visual C++";  
char s2[] = "6.0";  
strcat(s1, " "); // => "Visual C++ "  
strcat(s1, s2); // => "Visual C++ 6.0"
```

# Hàm tìm chuỗi trong chuỗi

**char\* strstr**(const char \***s1**, const char \***s2**)



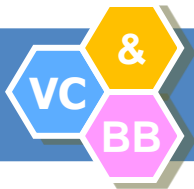
Tìm vị trí xuất hiện đầu tiên của **s2** trong **s1**



- ◆ **Thành công**: trả về con trỏ đến vị trí xuất hiện đầu tiên của **s2** trong **s1**.
- ◆ **Thất bại**: trả về **null**.



```
char s1[] = "Visual C++ 6.0";  
char s2[] = "C++";  
if (strstr(s1, s2) != null)  
    printf("Tim thay s2 trong s1...");
```



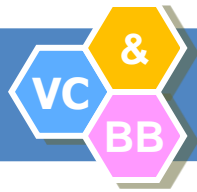
# Bài tập

❖ **Bài 1:** Xem thêm một số hàm khác như:

- **atoi, atol, atof** : đổi chuỗi thành số.
- **itoa, ltoa, ultoa**: đổi số thành chuỗi.
- **strtok**

❖ **Bài 2:** Viết hàm nhận vào một chuỗi và trả về chuỗi tương ứng (**giữ nguyên chuỗi đầu vào**):

- Các ký tự thành ký tự thường (giống **strlwr**).
- Các ký tự thành ký tự hoa (giống **strupr**).
- Các ký tự đầu tiên mỗi từ thành ký tự hoa.
- Chuẩn hóa chuỗi (xóa khoảng trắng thừa).



# Bài tập

- ❖ **Bài 3:** Viết hàm nhận vào một chuỗi s và trả về chuỗi tương ứng sau khi xóa các khoảng trắng.
- ❖ **Bài 4:** Viết hàm nhận vào một chuỗi s và đếm xem có bao nhiêu từ trong chuỗi đó.
- ❖ **Bài 5:** Viết hàm nhận vào một chuỗi s và xuất các từ trên các dòng liên tiếp.
- ❖ **Bài 6:** Viết hàm tìm từ có chiều dài lớn nhất và xuất ra màn hình từ đó và độ dài tương ứng.
- ❖ **Bài 7:** Viết hàm trích ra n ký tự đầu tiên/cuối cùng/bắt đầu tại vị trí pos của chuỗi s cho trước.