Chương 7
XÂU KÍ TỰ

Khoa Hệ thống thông tin quản lý

Hà Nội – 2015

Nội dung

1. Khai báo
2. Các thao tác trên xâu kí tự
3. Mảng xâu kí tự
4. Một số hàm xử lý xâu kí tự

1. Khai báo

- Khái niệm
  - Xâu kí tự trong C được xây dựng như một mảng một chiều các kí tự
  - Xâu kí tự kết thúc bằng kí tự '\0' (kí tự NULL trong bảng mã ASCII)
    - ➔ Độ dài tối đa của xâu= kích thước mảng – 1
    - ➔ Khai báo nên dành ra 1 ô nhớ để chứa kí tự '\0'
- Ví dụ

```
char line[80];     // Dài tối đa 79 kí tự
char hoten[30];    // Dài tối đa 29 kí tự
```

Khai báo có khởi tạo giá trị

- Độ dài cụ thể

```
char string[40]="Ngon ngu C";
```

  - Khởi tạo xâu kí tự có độ dài tối đa 39 kí tự với giá trị ban đầu là xâu "Ngon ngu C"

- Tự xác định độ dài

```
char str[]="Ngon ngu C";
```

  - Chương trình dịch tự bố trí một mảng để chứa dãy kí tự và 1 ô chứa kí hiệu '\0'

- Chú ý:
  - Khai báo xâu kí tự với con trỏ

```
char *message;
message="Xin chao!";
```

2. Các thao tác trên xâu kí tự

- Nhập xâu từ bàn phím
- Xuất xâu ra màn hình
- Xác định độ dài xâu
- Ghép xâu
- Sao chép xâu
- So sánh xâu
- Tìm kiếm kí tự
- …

a) Nhập xâu từ bàn phím

- Sử dụng hàm **scanf** với đặc tả "%s"

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

  - Chỉ nhận các kí tự từ bàn phím đến khi gặp kí tự dấu cách, tab, kí tự xuống dòng.
  - Xâu nhận được không bao gồm dấu cách

- Ví dụ:

```
char monhoc[50];
printf("Nhap mot xau ki tu: ");
scanf("%s", monhoc);
printf("Xau nhan duoc la: %s", monhoc);
```

```
Nhap mot chuoi: Ngon ngu lap trinh C
Chuoi nhan duoc la: Ngon_
```

a) Nhập xâu từ bàn phím (tt)

- Sử dụng hàm **gets**

```
gets(str);
```

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

- Ví dụ

```
char monhoc[50];
printf("Nhap mot chuoi: ");
gets(monhoc);
printf("Chuoi nhan duoc la: %s", monhoc);
```

```
Nhap mot chuoi: Ngon ngu lap trinh C
Chuoi nhan duoc la: Ngon ngu lap trinh C_
```

b) Xuất xâu ra màn hình

- Sử dụng hàm **printf** với đặc tả "%s"

```
char monhoc[50] = "Ngon ngu C";
printf("%s", monhoc);
```

```
Ngon ngu C_
```

- Sử dụng hàm **puts**

```
char monhoc[50] = "Ngon ngu C";
puts(monhoc);
```

```
Ngon ngu C
_
```

⇔ `printf("%s\n", monhoc);`

c) Xác định độ dài xâu

- Tự xác định

  Đếm cho đến khi gặp kí tự '\0'

```
char str[]="Ngon ngu C";
int dem=0;
while (str[dem]!='\0') dem++;
printf("Do dai xau la: %d ki tu",dem);
```

- Sử dụng con trỏ để xử lí xâu

```
char *message;
message="Ngon ngu C";
int dem=0;
while (*message!='\0'){*message++;dem++;}
printf("Do dai xau la: %d ki tu",dem);
```

c) Xác định độ dài xâu (tt)

- Dùng hàm

  Hàm **strlen(str)** trong thư viện `string.h`

```
printf("Do dai xau la:%d",strlen(str));
```

- Bài tập:
  1. Nhập từ bàn phím xâu st1, viết ra màn hình xâu đó theo chiều ngược lại.
  2. Nhập từ bàn phím xâu st2, chuyển xâu st2 sang chữ hoa và viết ra màn hình xâu kết quả ra màn hình.

d) Ghép xâu

- Ghép xâu st2 vào sau xâu st1
  - Hàm strcat(st1,st2): nối xâu st2 vào sau xâu st1

```
#include <stdio.h>
#include <string.h>
#define maxst 40
main()
{    char st1[maxst]="Chao mung";
     char st2[maxst]="Ngon ngu C";
     if (maxst>strlen(st1)+strlen(st2))
         puts(strcat(st1,st2));
     else printf("Khong du bo nho!");
}
```

  - Chú ý: không viết `st=st1+st2;`

e) Sao chép xâu

- C không cho phép gán một xâu cho một biến do xâu kí tự là một mảng.
  - Ví dụ: Không viết được `line="Hello";`

  ➔ Dùng hàm sao chép xâu hoặc viết vòng lặp sao chép từng kí tự
- Hàm sao chép xâu
  - `strcpy(st1,st2)` – gán xâu st2 cho xâu st1
  - Ví dụ: `strcpy(line,"Hello");`
- Chú ý:
  - Hàm không kiểm tra tính đúng đắn về kích thước ô nhớ của st1 có đủ chứa st2 hay không, do đó, cần lưu ý về kích thước xâu kí tự.

f) So sánh xâu kí tự

- Quy tắc so sánh
  - Các kí tự của 2 xâu được so sánh từng cặp từ trái qua phải theo giá trị của mã ASCII
- Hàm **strcmp(st1,st2)** trong thư viện string.h
  - Trả về 0 nếu st1==st2
  - <0 nếu st1<st2
  - >0 nếu st>st2
- Ví dụ: Nhập các xâu cho đến khi nhập xâu "done"

```
char st[80];
do
{     gets(st);
      printf("Xau vua nhap: %s \n",st);
}   while (strcmp(st,"done"));
```

f) So sánh xâu kí tự (tt)

- Một số hàm so sánh khác
  - **`stricmp(st1,st2);`** so sánh xâu st1 với st2 nhưng không phân biệt chữ hoa-thường.
  - **`strncmp(st1,st2,n);`** so sánh n kí tự đầu tiên của st1 và st2.
  - **`strnicmp(st1,st2,n);`** so sánh n kí tự đầu tiên của st1 và st2 nhưng không phân biệt chữ hoa-thường.

g) Tìm kiếm kí tự

- Hàm **strchr(str,c)** tìm kiếm kí tự c trong xâu **str**
  - Kết quả là con trỏ trỏ tới vị trí của kí tự c
  - Nếu không tìm thấy trả về NULL

```
char str[80];   char c;
printf("Nhap xau: ");gets(str);
printf("Nhap ki tu can tim: "); c=getchar();
if (strchr(str,c)) printf("Tim thay %c");
else printf("Khong tim thay!");
```

- Hàm **strstr(str1,str2)** tìm kiếm xâu str2 trong xâu **str1**
  - Kết quả là con trỏ trỏ tới vị trí của xâu str2
  - Nếu không tìm thấy trả về NULL

3. Mảng xâu kí tự

- Bài toán:
  - Nhập mảng các xâu kí tự, sắp xếp các xâu đó theo thứ tự từ điển

➔ Sử dụng mảng 2 chiều để lưu các xâu kí tự

```
void sapxep(int n,char x[][80])
{    char temp[80];
     int i,j;
     for (i=0;i<n-1;i++)
         for (j=i+1;j<n;j++)
             if (strcmp(x[i],x[j])>0)
             {strcpy(temp,x[i]);
              strcpy(x[i],x[j]);
              strcpy(x[j],temp);
             }
}
```

Mảng xâu kí tự (tt)

```
int main()
{
    char st[40][80];
    int i,n=0;
    printf("Nhap cac xau ki tu, ket thuc bang chu
\"end\"\n");
    do
    {
     printf("Nhap xau thu %d: ",n+1);
     gets(st[n]);
    }
    while (strcmp(st[n++],"end"));
    n--;
    sapxep(n,st);
    printf("\Day cac xau ki tu sau khi sap xep\n");
    for(i=0;i<n;i++) puts(st[i]);
}
```

4. Một số hàm xử lý xâu kí tự

- Các hàm trong thư viện **`string.h`**
  - Hàm **`strlwr(st)`**: chuyển xâu st thành chữ thường
  - Hàm **`strupr(st)`**: chuyển xâu st thành chữ hoa
  - Hàm **`strrev(st)`**: đảo ngược xâu st
- Các hàm xử lí kí tự trong **`ctype.h`**
  - Hàm **`toupper(c)`**: chuyển c thành chữ hoa
  - Hàm **`tolower(c)`**: chuyển c thành chữ thường
  - Hàm **`isalpha(c)`**: đúng (khác 0) nếu c là chữ cái
  - Hàm **`islower(c)`**: đúng nếu c là chữ cái thường
  - Hàm **`isupper(c)`**: đúng nếu c là chữ cái hoa
  - Hàm **`isspace(c)`**: đúng nếu c là dấu cách, dấu \n, dấu về đầu dòng \r, tab \t

Một số hàm xử lý xâu kí tự (tt)

- Các hàm chuyển đổi (trong **`stdlib.h`**)
  - Hàm **`atoi(str)`**: chuyển đổi xâu str thành số nguyên int
    - Ví dụ: Đọc một số nguyên có thể dùng cặp lệnh

```
gets(str);
n=atoi(str); //tránh đọc xong số mà bộ đệm vẫn còn \n
```

  - Hàm **`atol(str)`**: chuyển đổi xâu str thành số nguyên long
  - Hàm **`atof(str)`**: chuyển đổi str thành số thực float

Các hàm này bỏ qua các dấu cách ở đầu, chuyển cho đến khi gặp kí tự không thích hợp, nếu không chuyển được thì kết quả là 0

Bài tập thực hành

- Bài 1. Viết hàm upper(char s[]) đổi toàn bộ các kí tự sang kí tự hoa (giống hàm strupr)
- Bài 2. Viết hàm lower(char s[]) đổi toàn bộ các kí tự sang kí tự thường (giống hàm strlwr)
- Bài 3. Viết hàm proper(char s[]) đổi các kí tự đầu tiên của mỗi từ sang kí tự hoa.
- Bài 4. Đếm xem có bao nhiêu từ trong s ('từ' là tập hợp các kí tự in được không chứa các dấu cách, xuống dòng, tab). In ra màn hình các từ trong s.
- Bài 5. Viết các hàm left, right, mid…

Bài tập thực hành

- Bài 6. Tìm từ có độ dài lớn nhất trong xâu s.
- Bài 7. Xóa tất cả các khoảng trắng của s
- Bài 8. Viết hàm standard(char s[]) loại bỏ toàn bộ khoảng trắng đầu xâu, cuối xâu và giữa 2 từ trong s chỉ còn 1 khoảng trắng.
- Bài 9. Đếm số lượng từ ‘em’ trong xâu kí tự được đọc từ bàn phím.
- Bài 10. Kiểm tra xem từ có phải là palindrome hay không? Từ là palindrome là từ đọc từ phải sang trái cũng giống như đọc từ trái sang phải. Ví dụ: civic, madam, level,…