

C/C++ 程序设计

字符串与格式化输入/输出

张晓平

武汉大学数学与统计学院

1. 字符串简介

字符串简介

字符串简介 i

```
#include <stdio.h>
#include <string.h>
// density of human body: 1.04e3 kg / m^3
#define DENSITY 1.04e3
int main(void)
{
    float weight, volume;
    int size, letters;
    char name[40];
    printf("Hi! What's your first name?\n");
    scanf("%s", name);
    printf("%s, what's your weight in kilograms?\n", name);
}
```

字符串简介 ii

```
scanf("%f", &weight);
size = sizeof name;
letters = strlen(name);
volume = weight / DENSITY;
printf("Well, %s, your volume is %2.2f cubic
meters.\n", name, volume);
printf("Also, your first name has %d letters,\n
n", letters);
printf("and we have %d bytes to store in it.\n
", size);
return 0;
}
```

字符串简介

```
Hi! What's your first name?
```

```
xiaoping
```

```
xiaoping, what's your weight in kilograms?
```

```
60
```

```
Well, xiaoping, your volume is 0.06 cubic meters
```

```
.
```

```
Also, your first name has 8 letters,
```

```
and we have 40 bytes to store in it.
```

字符串简介

定义

字符串（string）就是一个或多个字符的序列。例如：

```
"Once more you open the door!"
```

注意

字符串用双引号括起来，但双引号不是字符串的一部分。

字符串简介

C 字符串

- C 没有为字符串定义专门的数据类型，而是把它存储在 `char` 数组中。
- 字符串的字符存放字符数组中，每个字符占用一个单元。

```
#include <stdio.h>
int main(void)
{
    char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
    printf("Greeting message: ");
    printf("%s\n", greeting);
    return 0;
}
```


C++ 字符串

C++ 提供了两种类型的字符串表示形式：

- C 风格字符串
 - C 风格的字符串起源于 C 语言，并在 C++ 中继续得到支持。
- C++ 引入的 `string` 类类型
 - C++ 标准库提供了 `string` 类类型。

字符串简介

```
// C style string
#include <iostream>
using namespace std;
int main()
{
    char greeting[6] = "Hello";
    cout << "Greeting message: ";
    cout << greeting << endl;
    return 0;
}
```

```
// C++ string class
#include <iostream>
#include <string>
using namespace std;
int main(void)
{
    string greeting = "Hello";
    cout << "Greeting message: ";
    cout << greeting << endl;
    return 0;
}
```

C 字符串

- C 字符串存储在字符数组中，最后一个元素为空字符 `'\0'`，用于标记字符串的结束。
- `'\0'` 不是数字 0，它是非打印字符，其 ASCII 码的值为 0。
- `'\0'` 的存在意味着数组长度至少要比存储字符数多 1。

定义：数组（array）

数组是同一类型的数据元素的有序序列。

定义：数组（array）

数组是同一类型的数据元素的有序序列。

```
char name[40];
```

该声明语句创建一个有 40 个存储单元的数组，其中每个单元可存储一个 char 型值。

定义：数组（array）

数组是同一类型的数据元素的有序序列。

```
char name[40];
```

该声明语句创建一个有 40 个存储单元的数组，其中每个单元可存储一个 char 型值。

- [] 说明 name 是一个数组
- [] 中的 40 指出数组的元素个数
- char 标识每个元素的类型

字符串简介：字符串的使用

要使用 C 字符串，必须创建一个数组，把字符串中的字符逐个放入数组中，最后还需在结尾添加一个空字符 `'\0'`。如：

```
char greeting[10] = {'H', 'e', 'l', 'l', 'o', ' ',  
'\0'};
```

字符串简介：字符串的使用

要使用 C 字符串，必须创建一个数组，把字符串中的字符逐个放入数组中，最后还需在结尾添加一个空字符 `'\0'`。如：

```
char greeting[10] = {'H', 'e', 'l', 'l', 'o', ' ', '\0'};
```

但这种方法太麻烦，我们可以通过如下方式来自动完成上述过程：

```
char greeting[10] = "Hello";
```


字符串简介：字符串的使用

```
#include <stdio.h>
#define PRAISE "What's a super marvelous name!"
int main(void)
{
    char name[40];
    printf("What's your name?\n");
    scanf("%s", name);
    printf("Hello, %s. %s\n", name, PRAISE);
    return 0;
}
```

字符串简介：字符串的使用

```
#include <stdio.h>
#define PRAISE "What's a super marvelous name!"
int main(void)
{
    char name[40];
    printf("What's your name?\n");
    scanf("%s", name);
    printf("Hello, %s. %s\n", name, PRAISE);
    return 0;
}
```

What's your name?

Xiaoping Zhang

Hello, Xiaoping. What a super marvelous name!

字符串简介：字符串的使用

```
#include <iostream>
#include <string>
using namespace std;
#define PRAISE "What's a super marvelous name!"
int main(void)
{
    string name;
    cout << "What's your name?" << endl;
    cin >> name;
    cout << "Hello, " << name << ". " << PRAISE <<
        endl;
    return 0;
}
```

字符串简介：字符串的使用

```
#include <iostream>
#include <string>
using namespace std;
#define PRAISE "What's a super marvelous name!"
int main(void)
{
    string name;
    cout << "What's your name?" << endl;
    cin >> name;
    cout << "Hello, " << name << ". " << PRAISE <<
        endl;
    return 0;
}
```

What's your name?

Xiaoping Zhang

Hello, Xiaoping. What a super marvelous name!

字符串简介：字符串的使用

C 字符串的输入

- 无须把 `'\0'` 插入 `name` 数组中，`scanf()` 会在读取输入时完成此任务。
- `name` 前无须加 `&`，因 `name` 本身就表示地址。
- 使用 `%s` 的 `scanf()` 语句会在遇到的第一个空格、制表符或换行符处停止读取，它只会把第一个单词而不是把整条语句作为字符串读入。

C++ 字符串的输入

- 使用 `cin` 读取字符串时，也按单词读取，即遇到第一个空格、制表符或换行符时会自动忽略后面的内容。

"x" 与 'x' 的差别

- 'x' 为字符，而 "x" 为字符串
- "x" 由两个字符 'x' 和 '\0' 组成

字符串简介：字符串长度

C 字符串

若使用字符数组来存储字符串 `char name[40] = "Hello";`, 则

- `sizeof name` 将 `name` 数组所占空间的字节大小, 即 40;
- 使用 `strlen(name)` 统计字符串 `name` 的字符个数, 不包含 `'\0'`, 即 5。

C++ 字符串

若使用字符串对象 `string name = "Hello";`, 则

- `sizeof` 计算 `name` 对象所占空间的字节大小;
- 使用类方法 `name.size()` 来统计字符串的字符个数, 不包含 `'\0'`, 即 5。

字符串简介：字符串长度

```
#include <stdio.h>
#include <string.h>
#define PRAISE "What's a super marvelous name!"
int main(void)
{
    char name[40];
    printf("What's your name?\n");
    scanf("%s", name);
    printf("Hello, %s. %s\n", name, PRAISE);
    printf("Your name of %ld letters occupises %ld
        memory cells.\n", strlen(name), sizeof name);
    printf("PRAISE has %ld letters and occupies %
        ld memory cells.\n", strlen(PRAISE), sizeof
        PRAISE);
    return 0;
}
```


字符串简介

What's your name?

Xiaoping

Hello, Xiaoping. What a super marvelous name!

Your name of 8 letters occupied 40 memory cells.

PRAISE has 30 letters and occpied 31 memory cells.

字符串简介：字符串长度 i

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
#define PRAISE "What's a super marvelous name!"
int main(void)
{
    string name;
    cout << "What's your name?" << endl;
    cin >> name;
    cout << "Hello, " << name << ". "
         << PRAISE << endl;
    cout << "Your name of " << name.size()
```

字符串简介：字符串长度 ii

```
        << " letters occupies " << sizeof name
        << " memory cells." << endl;
    cout << "PRAISE has " << strlen(PRAISE)
        << " letters and occupies " << sizeof
        PRAISE
        << " memory celss." << endl;
    return 0;
}
```

字符串简介：字符串长度

What's your name?

Xiaoping

Hello, Xiaoping. What's a super marvelous name!

Your name of 8 letters occupies 32 memory cells.

PRAISE has 30 letters and occupies 31 memory
cells.

字符串简介：字符串函数

- 头文件 `string.h` 包含许多与字符串相关的函数的原型，包括 `strlen` 函数。
- C 把函数库分成多个相关函数的序列，并为每个序列提供一个头文件。比如：
 - (1) `printf` 和 `scanf` 属于标准输入输出序列，使用 `stdio.h`。
 - (2) `strlen` 和其它一些与字符串相关的函数同属一个系列，使用 `string.h`。

字符串简介：printf 函数处理长字符串

- 一条 printf 语句占用两行，但只能在参数之间断行，不允许在字符串中间断行。
- 使用两个 printf 语句输出一行，换行符只出现在第二条语句。

字符串简介：sizeof 运算符与 strlen 返回值

设 name = "Morgan", 则

```
sizeof name : 40
```

```
strlen(name): 6
```

M	o	r	g	a	n	\0							
---	---	---	---	---	---	----	--	--	--	--	--	--	--

字符串简介：sizeof 运算符与 strlen 返回值

```
sizeof PRAISE : 29  
strlen(PRAISE): 28
```

sizeof 运算符在处理字符串变量时，会将空字符也计算在内。

字符串简介：sizeof 运算符后的圆括号

- 圆括号对于数据类型是必需的，而对于具体量则是可选的。

```
sizeof(float)
```

```
sizeof(char)
```

```
sizeof name
```

```
sizeof 2.15
```

- 建议在所有情况下都使用圆括号。

```
sizeof(name)
```

```
sizeof(2.15)
```