

04 字符串和格式化输入/输出

内容提要

- 字符串简介
- 常量和C预处理器
- printf()和scanf()
- 关键概念

前导程序

1 前导程序

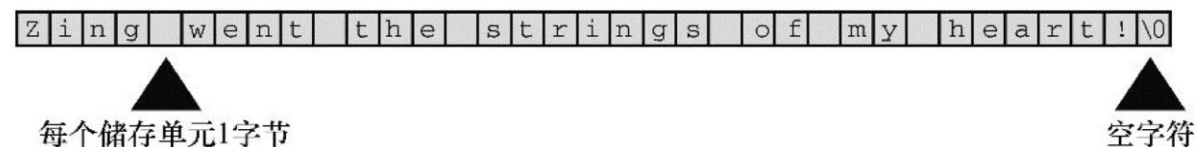
- [4.1 talkback.c](#)
- 数组(array)储存字符串(character string)
 - 输入的名字被储存在数组
 - 内存中40个连续的字节，每字节储存一字符值
- %s转换说明用于处理字符串的输入和输出
 - scanf()中，name没有&前缀
- 预处理器把字符常量DENSITY定义为62.4
- 函数strlen()获取字符串的长度

```
1. #include <stdio.h>
2. #include <string.h> // for strlen() prototype
3. #define DENSITY 62.4// human density in lbs per cu ft
4. int main(){
5.     float weight, volume;
6.     int size, letters;
7.     char name[40]; // name is an array of 40 chars
8.     scanf("%s", name);
9.     scanf("%f", &weight);
10.    size = sizeof name;
11.    letters = strlen(name);
12.    volume = weight / DENSITY;
13.    printf("%s's volume is %2.2f.\n", name, volume);
14.    printf(", %d letters,\n", letters);
15.    printf("%d bytes.\n", size);
16.    return 0;
17. }
```

字符串简介

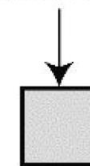
2 字符串简介

- 字符串(character string): 一个或多个字符的序列
- C语言没有专门用于储存字符串的变量类型
 - 字符串储存在char类型的数组中
 - 数组由连续的存储单元组成, 字符串中的字符被储存在相邻的存储单元中, 每个单元储存一个字符
- 空字符(null character)
 - C的字符串存储时通常以这个空字符\0结束
 - 空字符不是数字0, 它是非打印字符, 其ASCII码值是(或等价于)0
 - 空字符意味着数组的单元至少要比存储的字符多1.
- 使用字符串要创建数组
 - 把数组看作是一行连续的多个存储单元
 - `char name[40];` //创建了一个有40个存储单元的的数组
 - 必须先创建一个数组, 把字符串中的字符逐个放入数组, 末尾加上一个\0



分配1个字节
char ch;

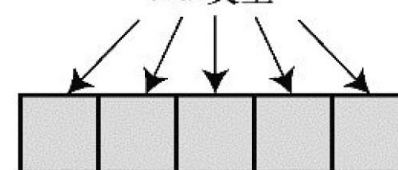
char类型



ch

分配5个字节
char name[5];

char类型



name

简单例子

➤ [程序清单4.2 praise1.c](#)

- %s告诉printf()打印一个字符串
- scanf()在读取输入时把空字符放入字符串末尾
- PRAISE后面用双引号括起来的文本是一个字符串。编译器会在末尾加上空字符
- scanf()只读取了它在遇到第1个空白(空格、制表符或换行符)时就不再读取输入

```
1. /* praise1.c -- uses an assortment of strings */
2. #include <stdio.h>
3. #define PRAISE "You are an extraordinary being."
4. int main(void)
5. {
6.     char name[40];
7.
8.     printf("What's your name? ");
9.     scanf("%s", name);
10.    printf("Hello, %s. %s\n", name, PRAISE);
11.
12.    return 0;
13. }
```

字符串和字符

- 字符串常量"x"和字符常量'x'不同
 - 区别之一在于'x'是基本类型(char)，而"x"是派生类型(char数组)
 - 区别之二是"x"实际上由两个字符组成：'x'和空字符\0

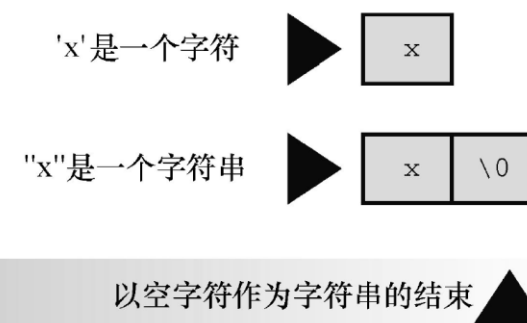
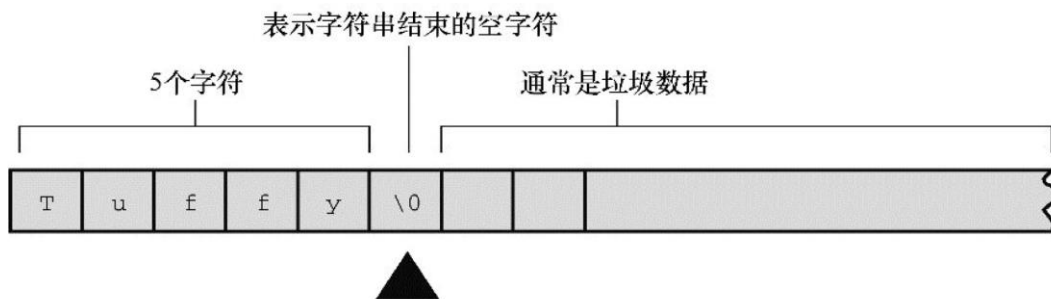


图4.3 字符'x'和字符串"x"

2.3 strlen()函数

- [程序清单4.3 praise2.c](#)
- strlen()函数给出字符串中的字符长度
- string.h头文件，包含多个与字符串相关的函数原型
- 函数库中相关的函数归为一类，每类函数一头文件
 - printf()和scanf()都隶属标准输入和输出函数，stdio.h
 - string.h: strlen()函数和其他与字符串相关的函数
 - 如拷贝字符串和字符串查找
- sizeof运算符
 - name数组有40个存储单元。只有前11个单元用来存储 Serendipity
- strlen()
 - 得出的结果是11。name数组的第12个单元存储空字符，strlen()并未将其计入



```

1. // try the %u or %lu specifiers if your
   implementation, does not recognize the %zd specifier
2. #include <stdio.h>
3. #include <string.h> /* provides strlen() prototype */
4. #define PRAISE "You are an extraordinary being."
5. int main(void){
6.     char name[40];
7.     printf("What's your name? ");
8.     scanf("%s", name);
9.     printf("Hello, %s. %s\n", name, PRAISE);
10.    printf(" %zd letters occupies %zd.\n",
        strlen(name), sizeof name);
11.    printf("The phrase has %zd letters",
        strlen(PRAISE));
12.    printf(", %zd memory cells.\n", sizeof PRAISE);
13.
14.    return 0;
15. }

```

常量和C预处理器

3 常量和C预处理器

- 使用符号常量(symbolic constant), 传达的信息更清楚明了
 - `circumference = 3.14159 * diameter;`
 - `circumference = pi * diameter;`
- 为什么使用符号常量更好?
 - 首先, 常量名比数字表达的信息更多
 - 假设程序中的多处使用一个常量, 有时需要改变它的值
 - 如果程序使用符号常量, 则只需更改符号常量的定义
 - 不用在程序中查找使用常量的地方, 然后逐一修改
- 符号常量的命名规则:
 - 第一个字符不能是数字, 可以用大小写字母, 数字和下划线字符
 - 一个不常用的命名规定, 在名字前加上前缀`c_`或者`k_`来表示常量

建立符号常量

➤ 方法1:

➤ `float taxrate;`

➤ `taxrate = 0.015;`

➤ 方法2: (用C预处理器)

➤ `#define TAXRATE 0.015`

➤ 注意格式: 首先是`#define` 其次是常量的符号名, 接着是常量的值。在C中, 常量通常大写

➤ [程序清单4.4 pizza.c](#)

➤ 方法3 : (比较灵活)

➤ `const float taxrate = 0.015; // taxrate的值不能被改变`

3.2 明示常量

- [程序清单4.5 defines.c程序](#)
- 头文件limits.h和float.h分别提供
 - 与整数类型和浮点类型大小限制相关的详细信息
 - 每个头文件都定义了一系列供实现使用的明示常量

```
1. // defines.c -- uses defined constants from limit.h
   and float.
2. #include <stdio.h>
3. #include <limits.h>    // integer limits
4. #include <float.h>    // floating-point limits
5. int main(void)
6. {
7.     printf("Some number limits for this system:\n");
8.     printf("Biggest int: %d\n", INT_MAX);
9.     printf("Smallest long long: %lld\n", LLONG_MIN);
10.    printf("One byte = %d.\n", CHAR_BIT);
11.    printf("Largest double: %e\n", DBL_MAX);
12.    printf("Smallest normal float: %e\n", FLT_MIN);
13.    printf("float precision = %d digits\n", FLT_DIG);
14.    printf("float epsilon = %e\n", FLT_EPSILON);
15.
16.    return 0;
17. }
```

`printf()`和`scanf()`

4 printf()和scanf()

- printf()函数和scanf()函数能让用户可以与程序交互
 - 它们是输出/输入函数
 - 或简称为I/O函数

- C90和C99标准规定了这些函数的标准版本，本课程亦遵循这一标准

4.1 printf()

- 请求printf()函数打印数据的指令要与待打印数据的类型相匹配
 - 转换说明(conversion specification), 它们指定了如何把数据转换成可显示的形式
- printf的使用格式
 - printf(control-string, item1, item2)
 - item1, item2是要打印的项目
 - 控制字符串control-string描述如何打印字符串
 - 控制字符串应该为每一个要打印的项目包含一个转换说明符
 - 使用修饰符及作为结果的输出



图4.6 printf()的参数

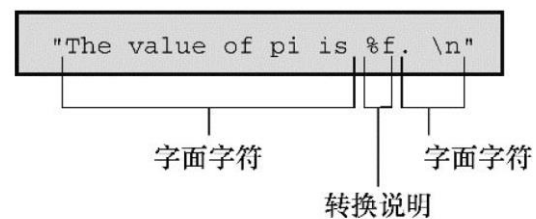


图4.7 剖析格式字符串

printf()用法提示

[程序清单4.6 printout.c](#)

- 打印几列数据时，可以指定固定字段宽度，默认
的字段宽度是数字的宽度
- 在两个转换说明之间放一个空白字符
- 语句中嵌入数字，指定一个小于等于数字宽度的
字段宽度。

[程序清单4.7 width.c](#)

[程序清单4.8 floats.c](#)

[程序清单4.9 flags.c](#)

[程序清单4.10 stringf.c](#)

```
1. /* printout.c -- uses conversion specifiers */
2. #include <stdio.h>
3. #define PI 3.141593
4. int main(void)
5. {
6.     int number = 7;
7.     float pies = 12.75;
8.     int cost = 7800;
9.
10.    printf("The %d contestants ate %f berry pies.\n",
        number, pies);
11.    printf("The value of pi is %f.\n", PI);
12.    printf("Farewell! thou art too dear for my
        possessing,\n");
13.    printf("%c%d\n", '$', 2 * cost);
14.
15.    return 0;
16. }
```

scanf()的使用

- [程序清单4.15 input.c](#)
- scanf()读取基本变量类型的值，在变量名前加&
- scanf()把字符串读进字符数组中，不用&
- scanf()函数使用空白(换行符、制表符和空格)把输入分成多个字段
 - 在依次把转换说明和字段匹配时跳过空白
 - 根据%c，scanf()会读取每个字符，包括空白
- scanf()函数的转换说明字符与printf() 的几乎相同。主要区别是scanf ()的%e %E %g %G只用于float类型，double类型主要用l修饰符

```

1. // input.c -- when to use &
2. #include <stdio.h>
3. int main(void)
4. {
5.     int age;           // variable
6.     float assets;     // variable
7.     char pet[30];     // string
8.
9.     printf("Enter your age, assets, and favorite
   pet.\n");
10.    scanf("%d %f", &age, &assets); // use the & here
11.    scanf("%s", pet); // no & for char array
12.    printf("%d $%.2f %s\n", age, assets, pet);
13.
14.    return 0;
15. }

```

从scanf()角度看输入

- 若使用%d说明符来读取整数
 - scanf()函数开始读取一个输入字符，跳过空白字符直到遇到一个非空白字符，scanf()持续读取保存字符直到它遇到一个非数字的字符
 - 如果第一个空白字符不是数字，scanf()不会去读下一个
 - 如果使用带有多个说明符scanf ()语句，函数在第一个出错的地方停止读取
- 若使用%s说明符
 - scanf ()保存再次遇到空白字符之前的所有非空白字符，scanf读取一个不包含空白字符的字符串
 - 当scanf ()把字符串放在数组中，添加终止的'\0'使数组内容变成字符串
- %c不会跳过空白字符
- 其他的输入函数
 - getchar()和fgets()，这两个函数更适合处理一些特殊情况，如读取单个字符或包含空格的字符串
- scanf("%d,%d", &n, &m);
 - scanf()函数将其解释成：用户将输入一个数字、一个逗号，然后再输入一个数字
 - 88,121
 - 88, 121
- scanf()函数返回成功读取的项数
 - 如果没有读取任何项，且需要读取一个数字而用户却输入一个非数值字符串，scanf()便返回0
 - 当scanf()检测到“文件结尾”时，会返回EOF

关键概念

关键概念

- 字符常量中的字符用双引号括起来，表达字符串要以空字符结尾。
- 参数的类型说明符要与后续参数值匹配，如果是scanf()记得给变量名加上&
- 空白字符影响scanf()的输出结果，除了%c外，在读取输入时，从第一个非空白字符读取到空白字符或一个不符合正在读取的类型的字符