

03 数据和C

内容提要

- 关键字: `int`、`short`、`long`、`unsigned`、`char`、`float`、`double`运算符
- 运算符: `sizeof()`
- 函数: `scanf()`
- 整数类型和浮点类型的区别
- 如何书写常量和声明变量
- 使用`printf()`和`scanf()`函数读写各种类型数据的值

示例程序

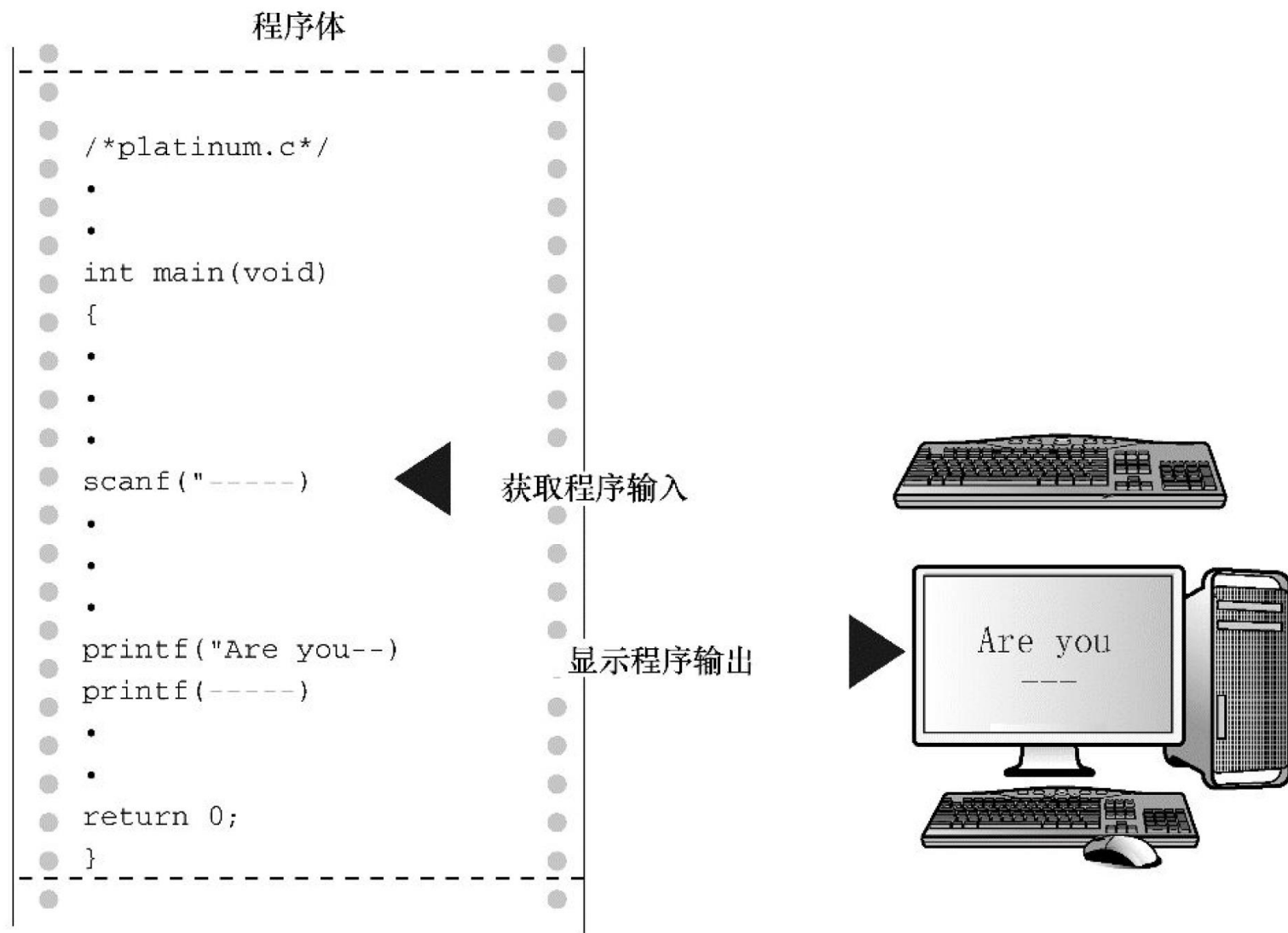
1 示例程序

➤ platinum.c程序

- 使用了一种新的变量声明
- 程序示范了常量的几种新写法，可以使用带有小数点的数
 - %f 说明符来处理浮点值。%.2f精确控制输出格式，使浮点数显示到小数点后两位
- scanf()函数为程序提供键盘输入
 - %f指示scanf() 读取一个浮点数，&weight 指定将输入值赋予变量weight中。scanf()函数使用&符号指示weight变量的位置
- 交互性
 - 计算机询问信息，并使用输入的数字
 - 交互性方法使程序更加灵活

```
1. #include <stdio.h>
2. int main(void){
3.     float weight;    /* user weight          */
4.     float value;    /* platinum equivalent */
5.
6.     printf("Are you worth?\n");
7.     printf("Let's check it out.\n");
8.     printf("Please enter your weight in pounds: ");
9.
10.    /* get input from the user */
11.    scanf("%f", &weight);
12.    /* assume platinum is $1700 per ounce */
13.    /* 14.5833 converts pounds avd. to ounces troy */
14.    value = 1700.0 * weight * 14.5833;
15.    printf("Your weight is worth %.2f.\n", value);
16.    printf("You are easily worth that!\n");
17.    return 0;
18. }
```

scanf()和printf()函数



变量与常量数据

2 变量与常量数据

➤ 数据

- 承载信息的数字与字符
- 对应类型：数字类型（整数，小数）和字符类型（实际上是一种整数特殊）
- 【数据是以 “指定类型的常量或者标识符变量 ” 的方式来实用的】

➤ 常量（constant）

- 在程序使用之前已经预先设定好，在整个程序的运行过程中没有变化
- 1700.0

➤ 变量（variable）

- 变量的值可以在程序执行过程中变化与指定，常量则不可以
- value

数据：数据类型关键字

3 数据：数据类型关键字

- 对变量而言，要在声明时指定其类型
- `int`关键字提供基本整数类型
 - `long`, `short`和`unsigned`, `signed`提供变种
 - `char`表示字母以及其他字符。也可表示小的整数
- `float`, `double`, `long double`
 - 表示带有小数点的数
- `_Bool`类型
 - 表示布尔值 (`true` 和 `false`)
- `_Complex` 和 `_Imaginary`
 - 分别表示复数和虚数
- 按在计算机中的存储方式，基本类型可划分为两个系统
 - 整数 (`integer`) 和浮点数 (`floating-point`) 类型

Table 3.1 C Data Keywords

Original K&R Keywords	C90 K&R Keywords	C99 Keywords
<code>int</code>	<code>signed</code>	<code>_Bool</code>
<code>long</code>	<code>void</code>	<code>_Complex</code>
<code>short</code>		<code>_Imaginary</code>
<code>unsigned</code>		
<code>char</code>		
<code>float</code>		
<code>double</code>		

位，字节和字

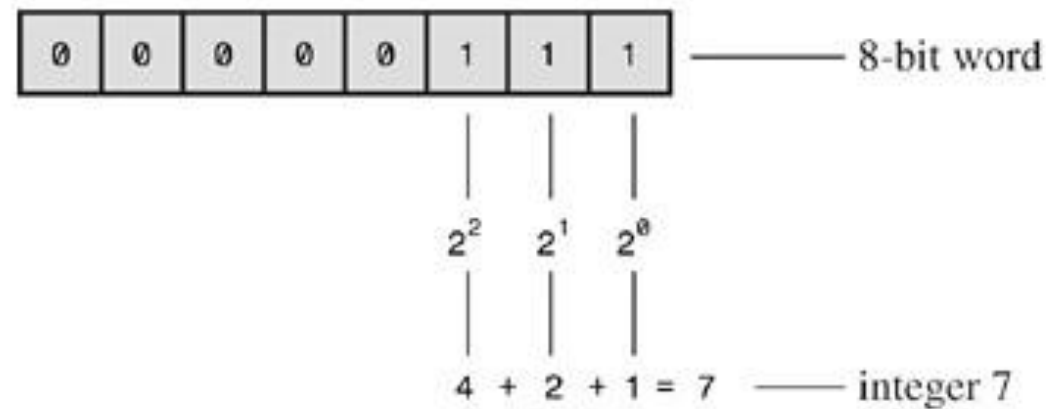
- 位、字节和字是描述计算机数据单元或存储单元的术语
- 位 (bit)
 - 最小的存储单元，储存0或1
- 字节 (byte)，常用的计算机存储单位
 - 对于几乎所有的机器，1字节均为8位。这是字节的标准定义，至少在衡量存储单位时是这样
- 字 (word)，设计计算机时给定的自然存储单位
 - 计算机的字长越大，其数据转移越快，允许的内存访问也更多

3.1 整数类型与浮点数类型

- ▶ 对人类而言，整数和浮点数的区别是它们的书写方式不同。对计算机而言，它们的区别是储存方式不同。

3.2 整数

▶ 整数 (integer)：没有小数部分的数



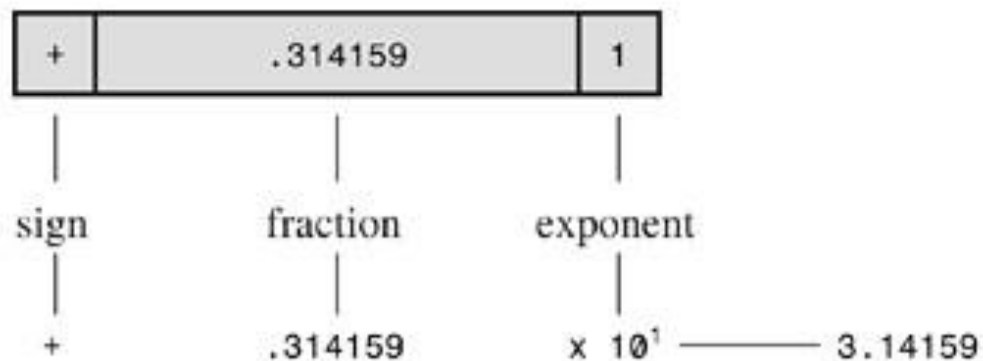
3.3 浮点数

➤ 浮点数(floating-point)

- 差不多可以和数学中实数(real number)概念相对应
- 在一个值后面加上一个小数点，该值就成为一个浮点值

➤ e记数法

- 浮点数分成小数部分和指数部分来表示，且分开储存这两部分



整数和浮点数的区别

- 整数没有小数部分，浮点数可以有小数部分
- 浮点数可以表示比整数范围大得多的数
- 对于一些算术运算（如两个很大的数相减），使用浮点数会损失更多精度
- 在任何区间内都存在无穷多个实数，计算机浮点数不能表示区域内所有的值
 - 浮点数往往只是实际值的近似
- 浮点运算通常比整数运算慢

C语言基本数据类型

4 C语言基本数据类型

- int类型
- 其他整数类型
- 使用字符：char类型
- _Bool类型
- 可移植类型：stdint.h和inttypes.h
- float、double和long double
- 复数和虚数类型
- 其他类型
- 类型大小

4.1 int类型

➤ 数据类型需要考虑

- 所提供数值的范围
- 数值是否可以取负值

➤ int 类型是有符号整数

- 即 int类型的值必须是整数，可以是正的，负的或者是 0

int类型

➤ 声明int变量

- 可以单独声明每个变量，也可在int后面列出多个变量名，变量名之间用逗号分隔

➤ 初始化变量

- 为变量赋一个初始值，直接在声明中完成

- `int cows = 32, goats = 14;`

➤ int 类型常量（整型字面量）

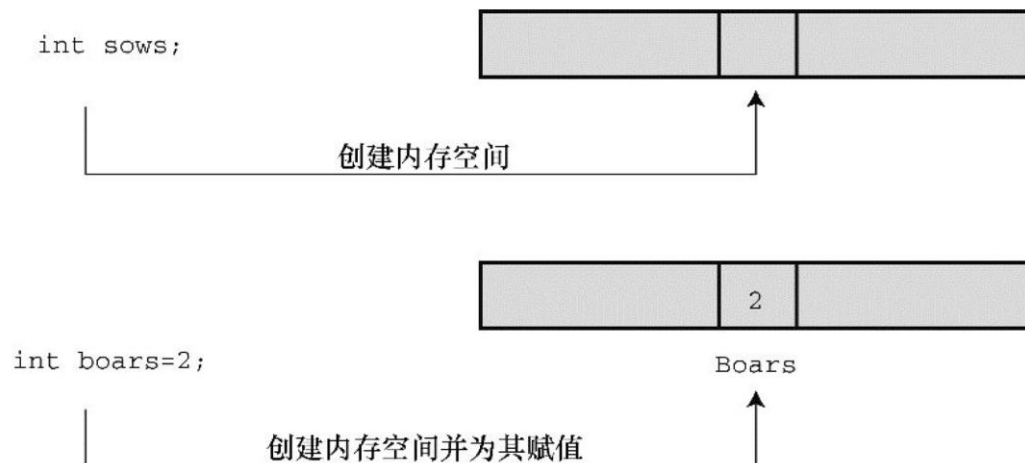
- 21, 22

➤ 打印 int 值([3.2 print1.c](#))

- `%d`

➤ 八进制和十六进制

➤ 显示八进制数和十六进制数([3.3 bases.c](#))



4.2 其他整数类型

- 3个附属关键字修饰基本整数类型：`short`、`long`和`unsigned`
 - `short int`类型（或者简写为`short`）占用的存储空间可能比`int`类型少，常用于较小数值的场合以节省空间。`short`是有符号类型。
 - `long int`或`long`占用的存储空间可能比`int`多，适用于较大数值的场合。`ong`是有符号类型。
 - `long long int`或`long long`占用的存储空间可能比`long`多，适用于更大数值的场合。至少占64位
 - `unsigned int`或`unsigned`只用于非负值的场合。
 - 在任何有符号类型前面添加关键字`signed`，可强调使用有符号类型的意图
- 整数溢出
 - 整数超出了相应类型的取值范围。溢出行为是未定义的行为，C标准未定义有符号类型的溢出规则
 - [toobig.c](#)
- 打印`short`、`long`、`long long`和`unsigned`类型
 - [3.4 print2.c](#)

4.3 使用字符：char类型

- char类型用于存储字符（字母或标点符号）
- 在技术实现上，char是整数类型
 - 用使用数字编码，即特定的整数表示特定的字符
 - 最常用的编码是 ASCII码
- C语言把1字节定义为char类型占用的位（bit）数

使用char

- 声明 char 类型变量
 - char response;
- 字符常量及其初始化
 - char grade = 'A';
 - char grade = 65; /* 对于ASCII, 这样做没问题, 但这是一种不好的编程风格 */
- 非打印字符(表3.2)
 - 转义序列 (escape sequence)
 - char nerf = '\n';
- 打印字符(3.5 charcode.c)
 - [程序清单3.5](#)
- 有符号还是无符号
 - 查阅limits.h头文件
 - unsigned char

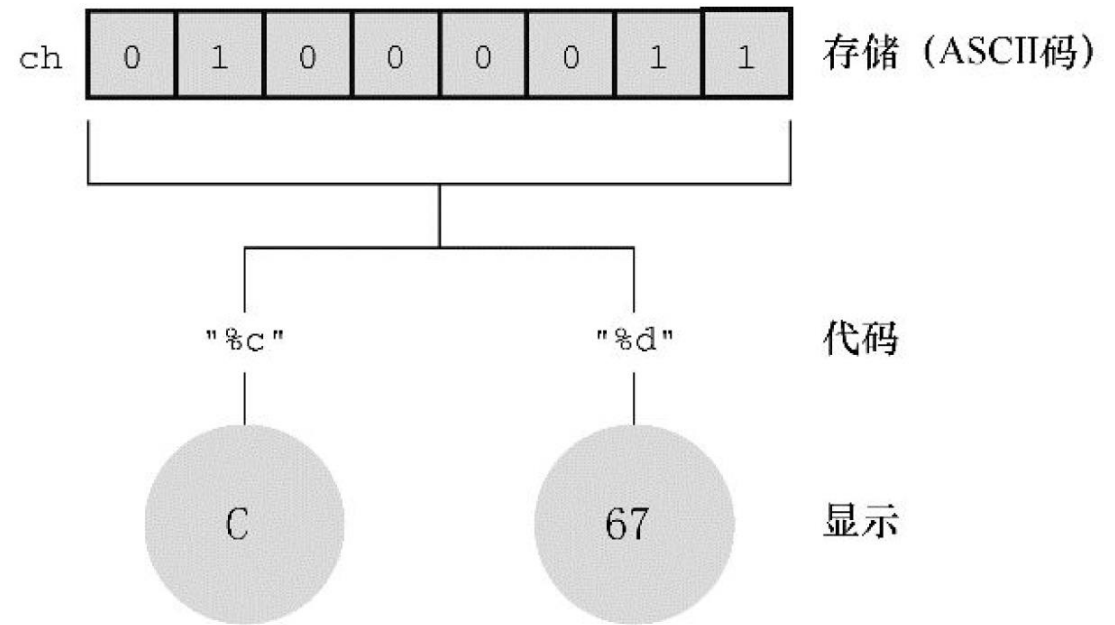


图3.6 数据显示和数据存储

4.4 _Bool类型

- `_Bool`类型由C99引入，用于表示布尔值
 - 逻辑值 `true`（真）与 `false`（假）
 - C用值1表示 `true`，用值0表示 `false`。所以 `_Bool`类型实际上也是一种整数类型

4.5 可移植的类型 : `inttypes.h`

- 已有类型有一些别名。基本的名字不够明确
- `int16_t` 表示一个 16位有符号整数类型, `uint32_t`表示一个 32位无符号整数类型
- 精确宽度整数类型 (`exact-width integer type`)
 - `int32_t`表示整数类型的宽度正好是32位。
 - 但是, 计算机的底层系统可能不支持。因此, 精确宽度整数类型是可选项
- 字符串宏
 - [程序清单3.6 `altnames.c`](#)

4.6 float, double, long double 类型

- C语言中浮点数包括 float, double, 和 long double 类型
- 科学记数法：十进制小数和 10 的幂的乘积来表示数字
- C标准规定，float 类型必须至少能表示6位有效数字
 - 能够表示33.333333的前6位数字
 - 浮点数要占用32位。其中8位用于表示指数的值和符号，剩下24位用于表示非指数部分（也叫作尾数或有效数）及其符号。
- C只保证 long double 类型至少同 double 类型一样精确

float, double, long double 类型

➤ 声明浮点变量

➤ `float planck = 6.63e-34;`

➤ 浮点常量

➤ `-1.56E+12 .45E-6`

➤ 打印浮点值(3.7)

➤ `%f`转换说明打印十进制记数法的`float`和`double`类型浮点数, 用`%e`打印指数记数法的浮点数

➤ 打印`longdouble`类型要使用`%Lf`、`%Le`或`%La`转换说明

➤ [程序清单3.7 showf_pt.c](#)

➤ 浮点值的上溢和下溢

➤ 当计算导致数字过大, 超过当前类型能表达的范围时, 上溢

➤ C语言规定, 在这种情况下会给`toobig`赋一个表示无穷大的特定值, 而且`printf()`显示该值为`inf`或`infinity`

➤ 把最小的正浮点数除以2将得到一个低于正常的值。如果除以一个非常大的值, 会导致所有的位都为0。下溢 (`underflow`)

➤ C语言把损失了类型全精度的浮点值称为低于正常的 (`subnormal`) 浮点值, C库已提供了用于检查计算是否会产生低于正常值的函数

➤ NaN (`not a number`)

4.7 复数和虚数类型

➤ `complex.h`头文件

4.8 其他类型

- 从基本类型中衍生出其他类型包括
 - 数组
 - 指针
 - 结构
 - 联合

4.9 类型大小

➤ sizeof

➤以字节为单位给出指定类型的大小

3.8 typesize.c

➤ 只要不在引号内部或一个单词中间,可以断行

```
1.  /* typesize.c -- prints out type sizes */
2.  #include <stdio.h>
3.  int main(void)
4.  {
5.      /* c99 provides a %zd specifier for sizes */
6.      printf("Type int, %zd bytes.\n", sizeof(int));
7.      printf("Type char, %zd bytes.\n", sizeof(char));
8.      printf("Type long, %zd bytes.\n", sizeof(long));
9.      printf("Type long long,%zd bytes.\n",
             sizeof(long long));
10.     printf("Type double, %zd bytes.\n",
             sizeof(double));
11.     printf("Type long double, %zd bytes.\n",
             sizeof(long double));
12.     return 0;
13. }
```

使用数据类型

5 使用数据类型

- 编写程序时，应注意合理选择所需的变量及其类型
- 用 `int` 或 `double` 类型表示数字，用 `char` 类型表示字符
- 在使用变量之前必须先声明，并选择有意义的变量名
- 初始化变量应使用与变量类型匹配的常数类型
 - `int apples = 3; /* 正确 */`
 - `int oranges = 3.0; /* 不好的形式 */`
 - `int cost = 12.99; /* 用double类型的值初始化int类型的变量 */`
 - `float pi = 3.1415926536; /* 用double类型的值初始化float类型的变量 */`

参数和易犯的错误

6 参数和易犯的错误

- ▶ printf参数个数
- ▶ scanf, &
- ▶ 3.9

转义序列

7 转义序列

➤ [3.10 escape.c](#)

➤ 并不一定能成功地与现代图形接口兼容，并不要求太多

7.2 刷新输出

- 用换行符刷新缓冲区
- `printf("enter your desired monthly salary;\n");`
- `scanf("%f", &salary);`

- [程序清单3.9 badcount.c](#)

- 在不同的平台下，缺少参数或参数类型不匹配导致的结果不同

8 关键概念

- C包含大量数值类型，这体现了为程序员提供方便 C的设计这一意图
- 计算机中的浮点数和整数在本质上不同，其存储方式和运算过程有很大区别
- 计算机在内存中用数值编码来表示字符

9 总结

- 基本的数据类型包含两大类：整数类型和浮点类型。
- 整数类型的两个重要特征是其类型的大小以及它是有符号还是无符号的
- 3种浮点类型为 `float`, `double` 和 ANSI C新增的 `long double`