

第 1 章 C 语言概述

C 语言是一门表达能力强并且逻辑严谨的计算机程序设计语言,使用 C 语言编写的程序,程序的运行效率非常高。正因如此,一些常见的软件系统,例如,操作系统、数据库管理系统、文字处理系统、杀毒软件、无人机飞行控制系统、工业自动化系统等都是使用 C 语言编写的。C 语言特别适合用于编写对运行效率和实时性要求都较高的应用场景的软件系统开发。同时,作为一门逻辑严谨的程序设计语言,学习 C 语言程序设计对建立良好、严谨、细致的程序思维非常有帮助。

1.1 C 语言简介

要介绍 C 语言,首先需要回顾一下计算机语言的发展历程。

在计算机出现之初,人们为了控制计算机执行指定任务,只能使用机器语言编写计算机程序。机器语言程序是一串由 0 和 1 构成的机器指令集合。那个年代,只有计算机科学家才能编写计算机程序。为了改变这种状况,计算机科学家设计了“汇编语言”,一种比机器语言稍微高级一点的计算机语言。汇编语言使用称为“汇编指令”的命令代替由 0、1 构成的机器指令。通过汇编语言,使得一些计算机专业人士也可以编写计算机程序。为了进一步便于人们编写计算机程序,计算机科学家丹尼斯·里奇在 1972 年发明并设计了 C 语言。

C 语言是一门很好的语言,C 语言的出现和广泛使用,使得计算机专业人士可以快速地进行软件系统的开发。到目前为止,C 语言仍然是一门应用极其广泛的语言。C 语言之所受到人们的喜爱和广泛应用,主要有如下特点。

(1) 表达能力强。C 语言作为高级语言的优秀代表和先驱,具有极强的表达能力,也就是说,使用 C 语言几乎可以表达任何复杂的计算逻辑。

(2) 逻辑严谨。C 语言的逻辑非常严谨,对于程序设计中语言上的错误,C 语言编译器都会进行检查并给出提示,进而提高了程序员的工作效率。

(3) 运行高效。使用 C 语言编写的程序是最接近机器语言的程序并且直接在计算机的 CPU 上运行,因此,用 C 语言编写的程序运行效率非常高。

(4) 可移植性强。使用 C 语言编写的程序可以方便地在不同的操作系统之间进行移植,例如,Linux 操作系统上使用 C 语言编写的程序要移植到 Windows 操作系统上运行,一般来说,只需在 Windows 操作系统下编译一下即可。

(5) 面向高级程序员。C 语言充分满足设计系统软件的需求,程序员使用 C 语言可以控制计算机硬件,也可以直接操作内存数据,甚至操作内存数据中的 bit 位。C 语言是开发

系统软件、智能装备控制系统的理想语言。

(6) 有助于程序员建立良好的程序思维。C 语言逻辑严谨,在使用 C 语言编写程序的过程中,不允许程序员出现丝毫马虎和错误,因此,非常有利于程序员形成严谨的程序思维逻辑,也非常有利于之后学习其他的程序设计语言。

1.2 建立 C 语言程序开发环境

C 语言是高级语言,为了使计算机能够理解并执行 C 语言程序,需要将 C 语言程序翻译为机器语言程序。这种可以把 C 语言程序翻译为机器语言程序的工具称为“C 语言编译器”。同时,还需要一款将 C 语言程序录入计算机中的工具,这样的工具称为“编辑器”。为了方便编辑和编译 C 语言程序,一些工具将编辑器和编译器整合在一起,这样的工具称为集成开发环境,简称 IDE。典型的 C 语言 IDE 有 Visual Studio、Dev C++、Code::Blocks、VS Code、CLion 等。不管使用哪款工具,开发 C 语言程序的逻辑和过程都是一样的。本书以跨平台的企业级开发工具 CLion 作为 IDE 工具。

1.2.1 下载并安装 CLion 集成环境

在搜索引擎中搜索 Download CLion,即可搜索到 CLion 工具的下载页面。进入 CLion 下载页面,如图 1-1 所示。

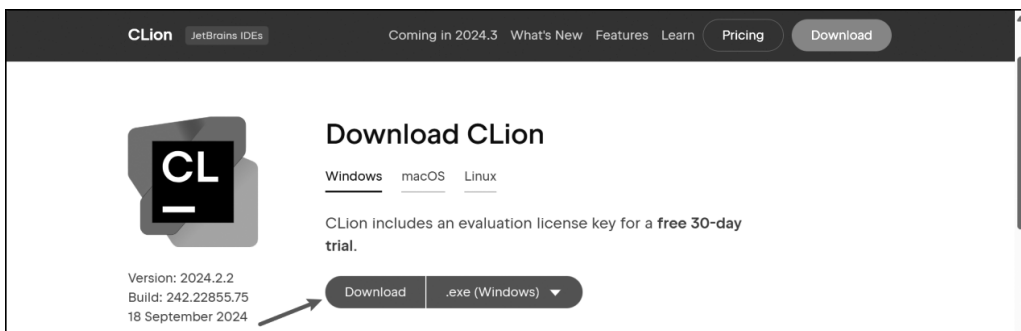


图 1-1 CLion 下载页面

下载完成后,运行这个安装程序,按照安装过程提示即可完成 CLion 工具的安

1.2.2 配置和验证安装

双击桌面上的 CLion 图标,启动 CLion。在“语言和地区”选择界面建议选择 English,然后继续按照提示运行直到进入激活码界面。在使用方式界面选择 Start Free 30-Day Trial 可试用 CLion。试用期后,可以根据需要激活,也可以根据需要使用其他的 C 语言 IDE。启动后的 CLion 界面如图 1-2 所示。在该界面中单击“+”可新建一个 C 语言程序工程,如图 1-3 所示(图中箭头和数字表示操作顺序,余同)。

新建了一个名称为 ch01-01 的 C 语言程序工程,并且把这个工程所涉及的所有文件放置在 C:/Wu/ws/ch01-01 的目录下,同时,该程序工程使用了 C23 标准。单击 Create 按钮,

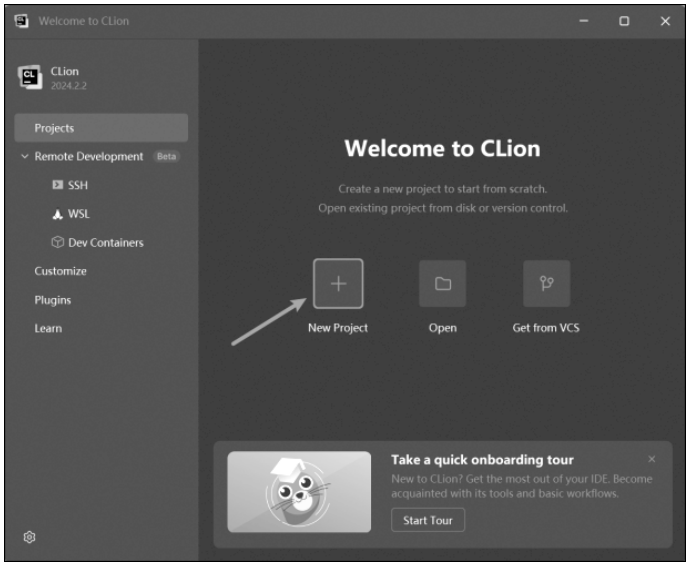


图 1-2 启动后 CLion 的界面

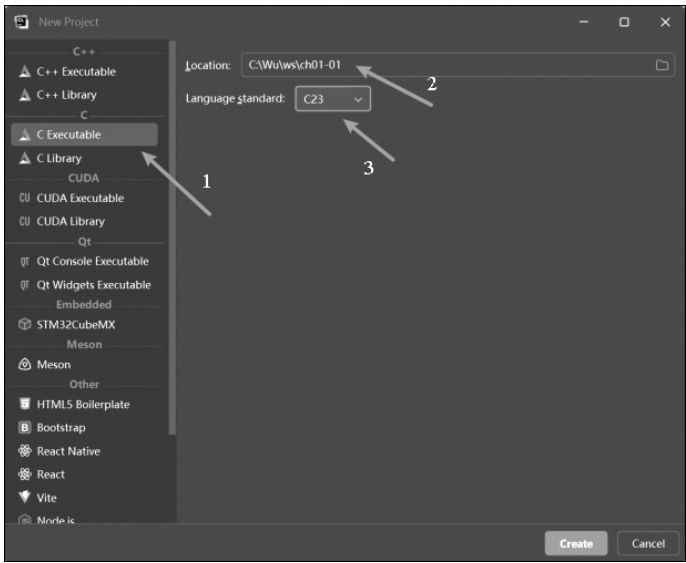


图 1-3 新建 C 语言程序工程

创建程序工程,在出现的 Open Project Wizard 界面中单击 OK 按钮,即可进入程序工程开发主界面,如图 1-4 所示。

可以根据需要调整或者关闭右边的窗格,调整左右窗格的宽度,也可以根据个人喜好修改主界面的背景色、字体和文字大小。在主界面中选择 File→Settings 命令,在弹出的对话框中选择 Appearance & Behavior→Appearance,可以修改背景色和文字大小,如图 1-5 所示。

在创建一个 C 语言程序工程时,CLion 已经创建了一个称为“Hello,World!”的简单程序。运行这个程序,将在计算机显示器上显示“Hello,World!”。为了验证这一点,同时也

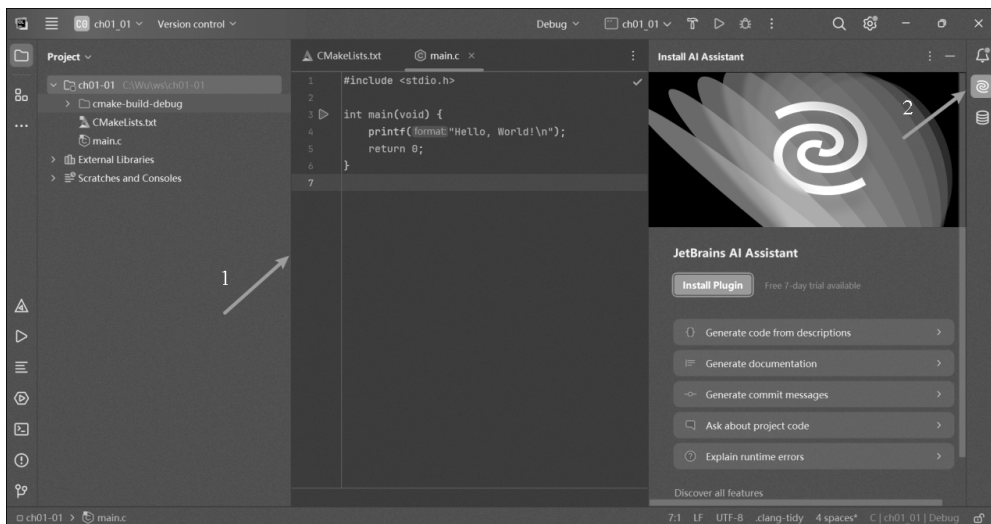


图 1-4 CLion 程序工程开发主界面

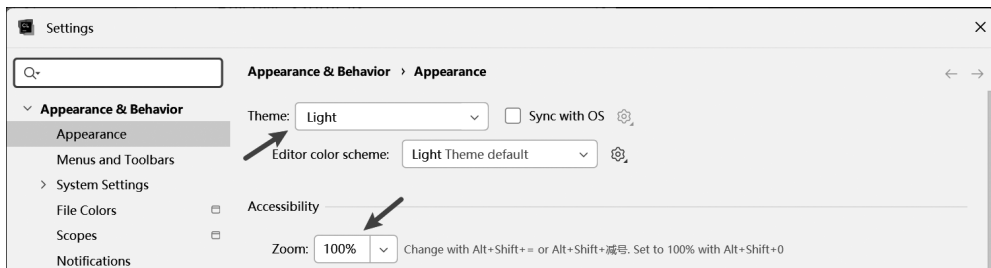


图 1-5 修改界面背景色和文字大小

为了验证安装的正确性,单击主界面工具栏右上方的三角形按钮,即可运行这个程序,如图 1-6 所示。

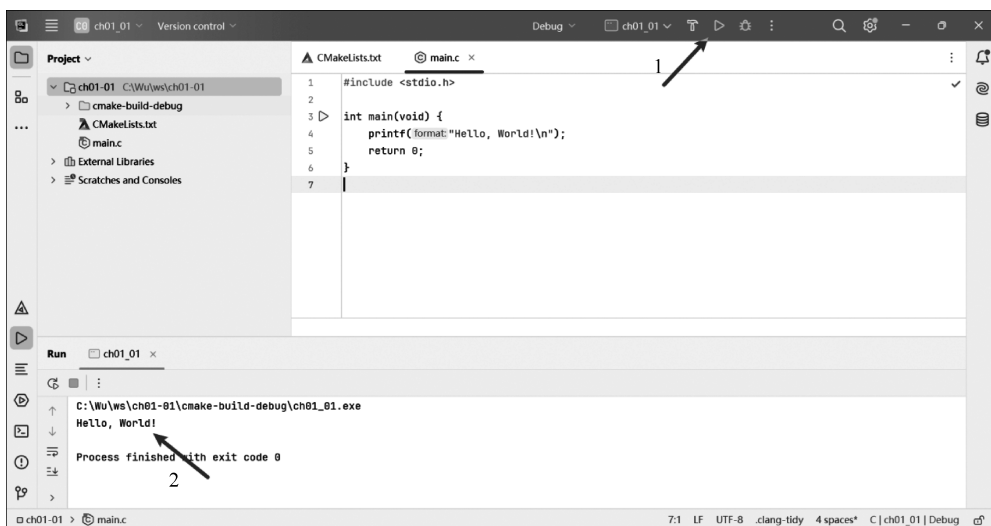


图 1-6 运行“Hello,World!”程序

在计算机显示器上显示了“Hello, World!”文字。至此,已经安装并已经验证了 C 语言开发环境的正确性。



下载、安装并配置 CLion 开发环境

1.3 C 语言程序基本结构

如前所述,在创建 C 语言程序工程时,CLion 会创建了一个最简单的 C 语言程序,人们亲切地称呼这个程序为“Hello, World!”程序。这个程序虽然简单,但是却具备了 C 语言程序的基本结构,任何复杂 C 语言程序都是从这里开始的。先看看“Hello, World!”程序:

```
#include <stdio.h>

int main(void) {
    printf("Hello, World! \n");
    return 0;
}
```

第 1 行:“#include <stdio.h>”是一条称为“包含”的预处理语句。这条语句的作用是将尖括号里指定的文件包含到程序代码中来。关于 #include 预处理语句的使用,在后续章节会再做详细介绍。

第 2 行:没有任何语句,称为空行。在 C 语言程序中,为了增加程序的可读性,会在适当的地方留出空行以便于程序员阅读代码。

第 3 行:“int main(void) {”是 C 语言程序的 main() 函数,也称为主函数,它是程序运行的入口。所谓运行入口,是指当计算机执行这个程序时,将一条语句接着一条语句地执行 main() 函数语句体中的语句。所谓语句体,就是由一对花括号包围的一组语句。任何可运行的 C 语言程序必须有且仅有一个 main() 函数。在 main() 函数前面的 int 表示当这个程序运行结束时,会给这个程序的启动者返回一个整数值(int 是 integer 的缩写)。main(void) 中的 void 表示这个 C 语言的 main() 函数没有参数,当 main() 函数没有参数时,void 是可以省略的。第三行最后的开花括号“{”和第六行闭花括号“}”是成对出现的,表示 main 函数的语句体。

第四行:“printf(“Hello, World! \n”);”语句将调用 C 语言的 printf() 函数在显示器上显示“Hello, World!”文字。C 语言程序中的每条语句必须以英文分号“;”结束。关于 printf() 函数的使用将在 1.5 节介绍。

第五行:“return 0;”语句返回整数 0 给这个程序的启动者。因为第三行中的 main() 函数的前面是一个 int,它要求当这个程序运行结束时,应该返回一个整数值给程序的启动者。当从 CLion 中单击 Run 按钮运行 C 语言程序时,程序的启动者就是 CLion。

第六行：“}”这个闭花括号与第三行末尾的开花括号“{”形成一对，这对花括号中的语句就是 main() 函数的语句体。

运行这个程序，计算机会先找到 main() 函数，然后一条语句接着一条语句执行 main() 函数中的语句，也就是 main() 后面的一对花括号所包围住的语句序列：

```
{
    printf("Hello, World! \n");
    return 0;
}
```

当执行语句体中的第一条语句，也就是“printf(“Hello, World! \n”);”语句时，将在计算机显示器显示“Hello, World!”文字，然后执行“return 0;”语句，将整数 0 返回给启动者并结束程序运行。现在，修改这个程序为如下代码：

```
#include <stdio.h>

int main(void) {
    printf("Hello, World! \n");
    printf("你好，未来的软件工程师! \n");
    return 0;
}
```

运行这个程序，将在计算机显示器上显示两行文字：

```
Hello, World!
你好，未来的软件工程师！
```

然后结束运行，并返回整数 0 给启动者。

1.4 程序运行乱码解决方法

在 Windows 操作系统环境下，由于操作系统使用的字符编码与 CLion 默认编码可以不同，从而导致在显示中文时出现乱码。

如果程序运行结果显示了乱码，特别是在显示中文信息时出现了乱码，可以通过修改 CLion 的相关配置来解决这个问题。具体方法：在 CLion 主界面中双击 Shift 键，显示如图 1-7 所示的窗口。

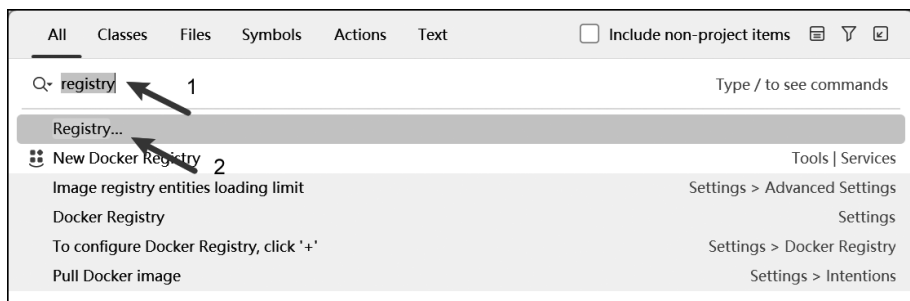


图 1-7 双击 Shift 键后显示的窗口

在图 1-7 的输入栏中输入 registry, 然后选择 Registry... 列表选项, 此时将显示如图 1-8 所示的 CLion 配置界面。

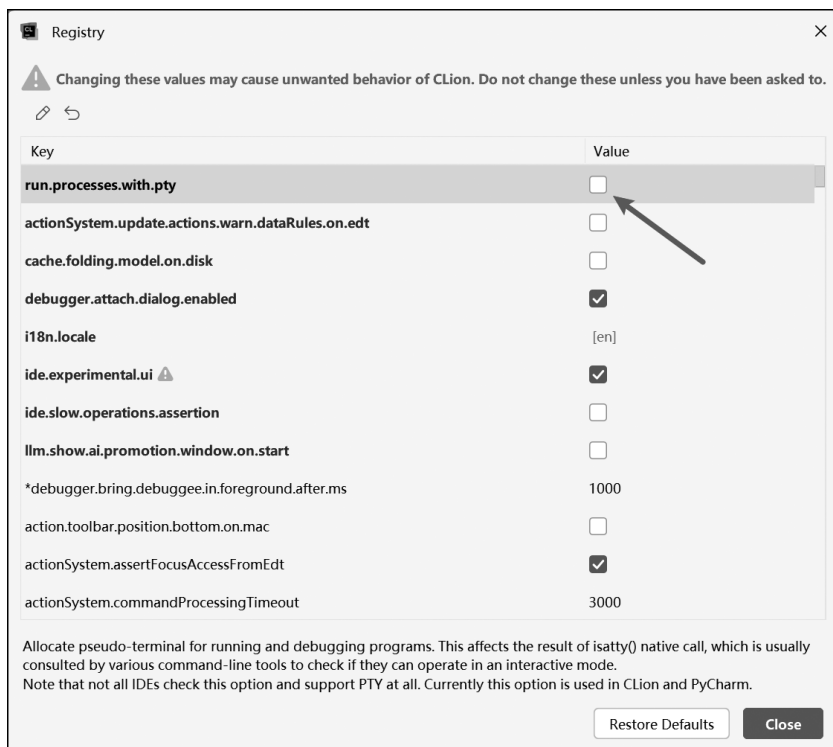


图 1-8 CLion 的配置界面

默认情况下, run.processes.with.pty 选项会被选中。为了正确显示中文, 需要取消选中这个选项。



出现中文乱码的解决方法

1.5 基本输出

在深入介绍 C 语言知识之前, 先介绍一下如何在 C 语言程序中显示数据, 以便为后续的学习奠定基础。C 语言使用 printf() 完成基本的信息输出。printf() 是 C 语言标准库函数, 用于将格式化后的数据输出到标准输出中, 所谓标准输出, 一般就是指计算机显示器。要使用 printf() 函数, 需要先使用 #include 语句包含 <stdio.h> 头文件, 也就是在程序代码中需要先使用如下语句包含 stdio.h 文件:

```
#include <stdio.h>
```

使用 printf() 函数输出数据的一般格式如下：

```
printf(const char * format, ...);
```

其中, format 表示显示数据时的格式;“...”表示要输出的数据,它们都称为函数的参数。例如,如下的程序将在计算机显示器上以十进制和十六进制显示 100 这个整数:

```
#include <stdio.h>

int main(void) {
    printf("十进制: %d,十六进制: %x\n", 100, 100);
    return 0;
}
```

运行这个程序,将显示如图 1-9 所示的结果。

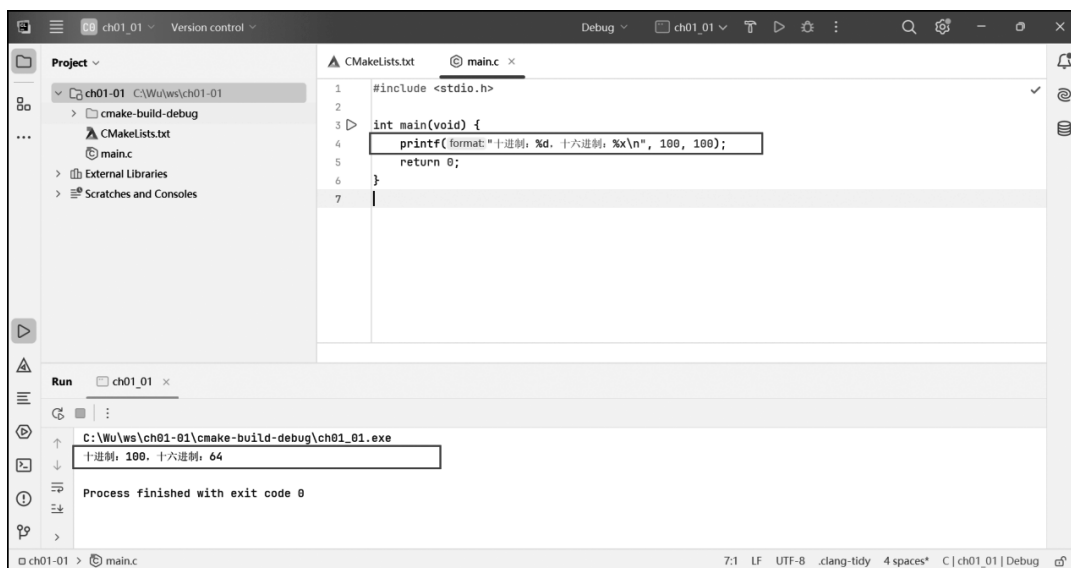


图 1-9 以十进制和十六进制显示整数 100 的值

printf() 函数中的 format 参数用于控制要显示数据的格式,包含了要被写入到标准输出的文本个数和格式要求。format 参数的一般格式如下:

```
%[flags][width][.precision][length]type
```

其中,用中括号括住的格式内容是可选的,也就是可有可无。下面对格式化参数中的符号进行解释说明。

1.5.1 type 类型控制符

type 类型控制符是必须要给出的格式化符号,它用于控制要显示的数据的类型。常用的 type 类型控制符及其含义如表 1-1 所示。

表 1-1 常用的 type 类型控制符及其含义

序号	控制符	含 义
1	d 或者 i	以十进制形式输出整数
2	o	以八进制形式输出无符号整数
3	x 或者 X	以十六进制形式输出无符号整数。小写的 x 控制符使用数字 0~9 和小写字符 a~f 表示十六进制符号,大写的 X 控制符使用数字 0~9 和大写字符 A~F 表示十六进制符号
4	u	以十进制形式输出无符号整数
5	f	以小数形式输出单、双精度实数
6	e 或者 E	以指数形式输出单、双精度实数。小写的 e 控制符以小写的 e 表示底,大写的 E 控制符以大写的 E 表示底
7	c	输出单个字符
8	s	输出字符串

例如,下面的程序以十进制、八进制、十六进制显示整数 200 到标准输出,同时,以双精度浮点和指数形式显示 123.4 这个实数到标准输出:

```
#include <stdio.h>

int main(void) {
    printf("十进制: %d,八进制: %o,十六进制: %x\n", 200, 200, 200);
    printf("%f,%E\n", 123.4, 123.4, 123.4);
    return 0;
}
```

运行这个程序,将显示如下结果:

```
十进制: 200,八进制: 310,十六进制: c8
123.400000,1.234000E+02
```

应注意,printf()函数中类型控制符的个数一定要等于格式化串后面的数据参数个数。例如,上例第一个 printf()函数中,因为在格式化串中有 3 个类型的控制符,所以,在格式化串的后面一定有 3 个要输出的数据;又如在第二个 printf()函数中,因为在格式化串中有两个类型控制符,所以,在格式化串的后面一定有两个要输出的数据。

类型控制符有时也称为“占位符”,其作用就是为 printf()函数格式化串后要显示的数据参数占位。例如:

```
printf("十进制: %d,八进制: %o,十六进制: %x\n", 200, 200, 200);
```

可以用如图 1-10 所示的方式形象地说明类型控制符与要显示的数据之间的对应关系。

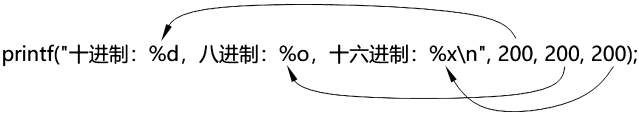


图 1-10 类型控制符与要显示的数据之间的对应关系

从图 1-10 可以看格式化串中的类型控制符的个数,以及类型严格控制数据参数的个数

和显示方式。

1.5.2 flags 标志控制符

flags 标志控制符控制输出的样式。常用的 flags 标志控制符及其含义如表 1-2 所示。

表 1-2 常用的 flags 标志控制符及其含义

序号	控制符	含 义
1	-	结果为左对齐,右边填充空格。默认是结果右对齐,左边填充空格
2	+	输出符号(正号或负号)
3	0	输出的前面补上 0,直到占满指定列宽为止
4	#	type 是 o、x、X 时,增加前缀 0、0x、0X;type 是 e、E、f 时,一定要使用小数点

例如,下面的程序输出整数 100、-200 这两个整数的正负号,当用十六进制输出整数 200 时,显示十六进制的整数符号 0x:

```
#include <stdio.h>

int main(void) {
    printf("%+d %+d\n", 100, -200);    //输出正负号
    printf("%x %#x\n", 100, 200);     //输出 0x
    return 0;
}
```

运行这个程序,显示如下结果:

```
+100 -200
64 0xc8
```

注意上例代码第 4 行和第 5 行的语句,在每行的末尾都有一段以双斜线开始的一句描述性的句子,这些以双斜线开始的描述性的句子称为“程序注释”。程序注释是为了便于程序员阅读而添加的描述性句子,编译程序在将程序编译为代码时会自动删除掉这些句子,因此,程序注释不影响程序的运行过程。双斜线注释的有效范围是从双斜线开始到一行的末尾。

除了可以使用双斜线为程序添加注释外,还可以使用“/*”和“*/”为程序添加多段注释。例如,下面的代码为程序添加了多行注释:

```
#include <stdio.h>

/*
    在这里,可以为程序增加多行注释。
    还可以再增加一行,以及更多行
*/
int main(void) {
    printf("%+d %+d\n", 100, -200);    //输出正负号
    printf("%x %#x\n", 100, 200);     //输出 0x
    return 0;
}
```