

大部分的现代计算机系统都使用 ASCII 标准来表示文本字符, 这种方式实际上就是用 一个唯一的单字节大小的整数值<sup>①</sup>来表示每个字符。比如, 图 1-2 中给出了 hello.c 程序的 ASCII 码表示。

#	i	n	c	l	u	d	e	SP	<	s	t	d	i	o	.
35	105	110	99	108	117	100	101	32	60	115	116	100	105	111	46
h	>	\n	\n	i	n	t	SP	m	a	i	n	(	)	\n	{
104	62	10	10	105	110	116	32	109	97	105	110	40	41	10	123
\n	SP	SP	SP	SP	p	r	i	n	t	f	(	"	h	e	l
10	32	32	32	32	112	114	105	110	116	102	40	34	104	101	108
l	o	,	SP	w	o	r	l	d	\	n	"	)	;	\n	SP
108	111	44	32	119	111	114	108	100	92	110	34	41	59	10	32
SP	SP	SP	r	e	t	u	r	n	SP	0	;	\n	}	\n	
32	32	32	114	101	116	117	114	110	32	48	59	10	125	10	

图 1-2 hello.c 的 ASCII 文本表示

hello.c 程序是以字节序列的方式储存在文件中的。每个字节都有一个整数值, 对应于某些字符。例如, 第一个字节的整数值是 35, 它对应的就是字符“#”。第二个字节的整数值为 105, 它对应的字符是‘i’, 依此类推。注意, 每个文本行都是以一个看不见的换行符‘\n’来结束的, 它所对应的整数值为 10。像 hello.c 这样只由 ASCII 字符构成的文件称为文本文件, 所有其他文件都称为二进制文件。

hello.c 的表示方法说明了一个基本思想: 系统中所有的信息——包括磁盘文件、内存中的程序、内存中存放的用户数据以及网络上传送的数据, 都是由一串比特表示的。区分不同数据对象的唯一方法是我们读到这些数据对象时的上下文。比如, 在不同的上下文中, 一个同样的字节序列可能表示一个整数、浮点数、字符串或者机器指令。

作为程序员, 我们需要了解数字的机器表示方式, 因为它们与实际的整数和实数是不同的。它们是对真值的有限近似值, 有时候会有意想不到的行为表现。这方面的基本原理将在第 2 章中详细描述。

### 旁注 C 编程语言的起源

C 语言是贝尔实验室的 Dennis Ritchie 于 1969 年~1973 年间创建的。美国国家标准学会(American National Standards Institute, ANSI)在 1989 年颁布了 ANSI C 的标准, 后来 C 语言的标准化成成了国际标准化组织(International Standards Organization, ISO)的责任。这些标准定义了 C 语言和一系列函数库, 即所谓的 C 标准库。Kernighan 和 Ritchie 在他们的经典著作中描述了 ANSI C, 这本著作被人们满怀感情地称为“K&R”[61]。用 Ritchie 的话来说[92], C 语言是“古怪的、有缺陷的, 但同时也是一个巨大的成功”。为什么会成功呢?

- C 语言与 Unix 操作系统关系密切。C 从一开始就是作为一种用于 Unix 系统的程序语言开发出来的。大部分 Unix 内核(操作系统的核心部分), 以及所有支撑工具和函数库都是用 C 语言编写的。20 世纪 70 年代后期到 80 年代初期, Unix 风行于高等院校, 许多人开始接触 C 语言并喜欢上它。因为 Unix 几乎全部是用 C 编写的, 它可以很方便地移植到新的机器上, 这种特点为 C 和 Unix 赢得了更为广泛的支持。

① 有其他编码方式用于表示非英语类语言文本。具体讨论参见 2.1.4 节的旁注。