Chapter II

Types, Operators and Expressions

#Data Types and Sizes

/* signed and unsigned */

signed and unsigned are the qualifier in C. Simply, signed int or char are positive or negative while unsigned int or char are positive. To understand the qualifier of int, the way that how integers be stored need to be familiar with first.

Let’s discuss signed first. For instance, an 16-bit (2-Byte) integer in decimal will be converted to binary then be stored in computer as table 1-1. Commonly, the first bit (from the left hand) is the sign of the integer. Thus, the range of the integer is from -2^15 (-32768) to 2^15-1 (32767). This is how signed int works.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 1-1

How to represent negative? The idea is using two ’s complement. First, swapping all 0 and 1, so we got the one’s complement (反码) as table 1-2. Then, adding 1 to the one’s complement so we got two’s complement (补码) as table 1-3.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Table 1-2

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 1-3

Once we comprehend how signed works, unsigned will be easy to understand. Taking unsigned char for an instance. unsigned integers be stored in computer just like signed but removed the first bit which is the sign. Therefore, unsigned integers are able to represent positive numbers from 0 to 2^16-1 (65535). *Whether plain chars are signed to unsigned is machine-dependent, but printable characters are always positive.*