*附录3 运算符优先级表*

就像四则运算中，有个规定是“先加减，后乘除”一样，C语言中的各种运算符也有其优先级，这样当一个表达式中有多个运算符时，程序才知道正确的运算顺序。

运算符有两个属性，一个是优先级，一个是结合性，优先级直接决定了运算符的运算顺序，就像\*和/运算符的优先级高于+和-。而结合性则是在有多个相同的运算符时，每个运算符的执行顺序。

|  |  |  |  |
| --- | --- | --- | --- |
| 优先级 | 运算符 | 作用 | 结合方向 |
| 1 | () | 函数调用 | 从左到右 |
| [] | 下标运算符 | 从左到右 |
| . | 结构体成员选择 | 从左到右 |
| -> | 通过指针选择成员 | 从左到右 |
| 2 | ++ | 自增 | 从右到左 |
| -- | 自减 | 从右到左 |
| - | 取负值 | 从右到左 |
| sizeof | 取大小 | 从右到左 |
| & | 取地址 | 从右到左 |
| ! | 逻辑非 | 从右到左 |
| (type) | 强制类型转换 | 从右到左 |
| \* | 取值 | 从右到左 |
| ~ | 按位取反 | 从右到左 |
| 3 | \* | 乘法 | 从左到右 |
| / | 除法 | 从左到右 |
| % | 取余 | 从左到右 |
| 4 | + | 加法 | 从左到右 |
| - | 减法 | 从左到右 |
| 5 | << | 左移 | 从左到右 |
| >> | 右移 | 从左到右 |
| 6 | > | 大于 | 从左到右 |
| >= | 大于等于 | 从左到右 |
| < | 小于 | 从左到右 |
| <= | 小于等于 | 从左到右 |
| 7 | == | 等于 | 从左到右 |
| != | 不等于 | 从左到右 |
| 8 | & | 按位与 | 从左到右 |
| 9 | ^ | 按位异或 | 从左到右 |
| 10 | | | 按位或 | 从左到右 |
| 11 | && | 逻辑与 | 从左到右 |
| 12 | || | 逻辑或 | 从左到右 |
| 13 | ?: | 条件 | 从右到左 |
| 14 | = | 赋值 | 从右到左 |
| 15 | , | 逗号 | 从左到右 |

说明

一.()除了可以表示函数调用以外，还可以用来分隔表达式以达到改变运算顺序的效果，如：1+2\*3是先计算2\*3，而如果我们加上括号改写为这样：(1+2)\*3，则优先计算的就是1+2。当需要使用多重括号时，一律使用圆括号，而不使用[]和{}，这一点和数学上不同。

二．优先级的用法

从上表可知，逻辑非运算符(!)的优先级高于加法运算符(+)，则如果有如此表达式：!10+1，则运算顺序是优先计算!10，再计算!10+1，所以结果为1。

取值运算符\*的优先级低于下标运算符[]，则\*p[i]的等价于\*(p[i])，而(\*p)[i]表示与\*p[i]不同的意思，\*(p[i])中，p是一个数组，而\*(p[i])则表示这个数组中的某个元素的取值运算的结果；而(\*p)[i]中，p是一个指针，(\*p)[i]表示对p进行取值运算，而取值运算的结果是一个数组，然后再取这个数组中的某个元素，这一点在第7章有所提及。

三．运算符结合性

假设有3个变量a、b=1、c=2，并且有表达式a=b=c;那么最终a的结果到底是1还是2。从上表可以看出，=运算符的结合性是从右到左，所以a=b=c中，第2个=会被优先计算，因此，计算顺序是b=c,a=b，所以最终a的值为2。

[]运算符的结合性是从左到右，因此p[i][j]的运算顺序是先计算p[i]再计算(p[i])[j]，而我们知道p[i][j]可以改写成\*(\*(p+i)+j)，p+i在最内层括号，得到的结论是相同的。