

嵌入式C语言之- 逻辑运算符

讲师：叶大鹏

助力你成为优秀的电子工程师！

整体介绍

- 逻辑运算符用于连接多个关系表达式，包括：

! (一元)、&&、||

- 逻辑表达式的结果是bool类型，要么是0 (false)，要么是1(true)，应用于条件判断语句和循环控制语句中；
- 逻辑运算的真值表：

a(bool值)	b(bool值)	!a(逻辑非，不改变a本身值)	a && b(逻辑与)	a b(逻辑或)
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

- “逻辑与” 口诀：两个操作数都为真，结果才为真；一个操作数为假，结果就为假；
- “逻辑或” 口诀：一个操作数为真，结果就为真；两个操作数都为假，结果才为假；

运算符优先级

运算符（优先级从上往下）	运算符说明及应用场景	结合性
() [] -> .	括号（函数等），数组，结构体指针变量的成员访问，普通结构体变量的成员访问	由左向右
! ~ ++ -- + -	逻辑非，按位取反，自增1，自减1，正号，负号	由右向左
* & (类型) sizeof	间接，取地址，强制类型转换，求占用空间大小	
* / %	乘，除，取模	由左向右
+ -	加，减	由左向右
<< >>	左移，右移	由左向右

运算符优先级

运算符（优先级从上往下）	运算符说明及应用场景	结合性
< <= >= >	是否小于, 是否小于等于, 是否大于等于, 是否大于	由左向右
== !=	是否等于, 是否不等于	由左向右
&	按位与	由左向右
^	按位异或	由左向右
	按位或	由左向右
&&	逻辑与	由左向右
	逻辑或	由左向右
?:	条件	由右向左
= += -= *= /= %= &= ^= = <<= >>=	各种赋值运算符	由右向左
,	逗号（顺序）	由左向右

赋值运算符优先级 < && 和 || < 关系运算符优先级 < 算数运算符优先级 < !

优先级案例

$a > b \ \&\& \ c > d$

等价于: $(a > b) \ \&\& \ (c > d)$

$!b == c \ || \ d < a$

等价于: $((!b) == c) \ || \ (d < a)$

$a + b > c \ \&\& \ x + y < z$

等价于: $((a + b) > c) \ \&\& \ ((x + y) < z)$

- 当表达式比较复杂或者不确定优先级时, 建议使用 `()` 进行明确, 减少失误。

应用案例

```
int year = 2016;
int daysOfFebruary;
if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
{
    daysOfFebruary = 29;
}
else
{
    daysOfFebruary = 28;
}
printf("daysOfFebruary = %d\n", daysOfFebruary);
```

• 输出结果:

```
daysOfFebruary = 29
```

逻辑运算 “短路”

- 逻辑运算是自左向右进行的，如果左边的结果已经能够决定表达式的结果了，就不会再做右边的计算：

`(year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)`

当year为2016时，由于||左边的结果为1，所以表达式的结果已经为1了，所以`year % 400 == 0`不会被执行。

- 对于&&，左边是0时，就不再做右边计算了；
- 对于||，左边是1时，就不再做右边计算了。

```
uint8_t a = 10;
uint8_t b = 9;
if (a != 10 && ++b == 10)
{
    printf("This is true\n");
}
printf("b = %d\n", b);
```

- 输出结果：

```
b = 9
```

编程规范

1. 在编写包含&&的表达式时，将最有可能为假的子表达式写在表达式的最左边；
 2. 在编写包含||的表达式时，将最有可能为真的子表达式写在表达式的最左边；
- 此方法可以提高程序的执行效率。

THANK YOU!