第二十四节：借用 unsigned long 类型的中间变量可以减少溢出现象。

【24.1　为什么要借用 unsigned long 类型的中间变量？】

为什么要借用 unsigned long 类型的中间变量进行算术运算？其实就是为了减少溢出的问题。溢出是因为数据超过了它的最大范围，unsigned char ,unsigned int ,unsigned long 三种数据类型中，unsigned long 的取值是最大的。当参与运算变量中存在非 unsigned long 类型的时候，在运算前，先让每个非 unsigned long 类型的变量借用一个 unsigned long 类型的中间变量，然后才开始运算，可以大大减少运算中的溢出问题。unsigned long 的取值是从 0 到 4294967295，万一数据超过了 4294967295 怎么办？可用 BCD 码的数组方式进行运算，这种数组运算的方法我以后会跟大家介绍，初学者现在暂时不用深入了解它。

【24.2　如何借用 unsigned long 类型的中间变量？】

借用中间变量的方法是引入中间变量，有多少个非 unsigned long 类型变量就引入多少个 unsigned long 中间变量，再借这个“壳”进行运算，最后再把中间变量的计算结果返回给实际变量。请看下面例子。

转换之前：

```
unsigned int  a;
unsigned char x=195;
unsigned long y=101;
a=x-y;       //进行算术减法运算
```

分析：

上述公式用到 3 个变量，其中 a 和 x 都不是 unsigned long 变量，因此需要为它们分别引入两个 unsigned long 类型的中间变量 t 和 s，于是乎，继续往下看.....

转换之后：

```
unsigned int  a;
unsigned char x=195;
unsigned long y=101;

unsigned long t; //引入的中间变量 t，用来给 a 借用。
unsigned long s; //引入的中间变量 s，用来给 x 借用。

//第一步：使用之前先清零
t=0;              //t 在用之前，先把 t 的 32 位全部清零。
s=0;              //s 在用之前，先把 s 的 32 位全部清零。

s=x;              //s 接收 x 原数据，等效于 x 借用 unsigned long 中间变量 s 这个壳。
t=s-y;            //此处 unsigned long 类型的 t 就默认代表了 unsigned int 类型的变量 a。

//第二步：因为其它的变量都是临时的，所以运算结束后再返回计算结果给原来的变量。
a=t;              //运算结束后再把计算结果返回给原来的变量 a。
```

分析：

第一步：unsigned long 类型的中间变量在转换之前为什么要先赋值 0 进行清零，比如上述代码的

“s=0;”？因为它是 32 位的数据类型，它也是一个随机数，如果不清零，后续的其它类型的变量可能是 16 位或者 8 位的类型变量，这些宽度不一的变量在给 32 位的变量赋值的时候，只能覆盖到 32 位变量的低 16 位或者低 8 位，无法等效于实际借用者变量的数值，所以有可能会出错。

第二步：因为其它的变量都是临时的，所以运算结束后应该再返回计算结果给原来的实际变量。在这里要多说一句，实际项目中，最后接收运算结果的变量应该根据项目所需去选择它的类型，建议尽量选择 unsigned long 类型吧，否则，如果中间变量的计算结果大于接收变量本身的类型范围，也会发生溢出。比如，上述最后一行代码 a=t，如果此时 t 的数值大于 65535，a 也会发生溢出的现象。 但是如果 a 本身是 unsigned long 类型，就不会发生这种现象。

加法，乘法，除法在借用中间变量的时候，跟本节减法例子中的思路也大同小异。

【24.3　建议在算术运算中确保所有的变量都是 unsigned long 类型。】

不管是以前讲的加法，现在讲的减法，还是未来讲的乘法和除法，我都会建议“在加减乘除四则运算中，凡是非 unsigned long 类型的变量，都应该借用 unsigned long 类型的中间变量进行运算，最后再返回计算结果给实际的变量。”unsigned long 变量是三种数据类型中取值范围最大的数，借用此类型的中间变量，可以减少在简单运算中可能出现的溢出问题。