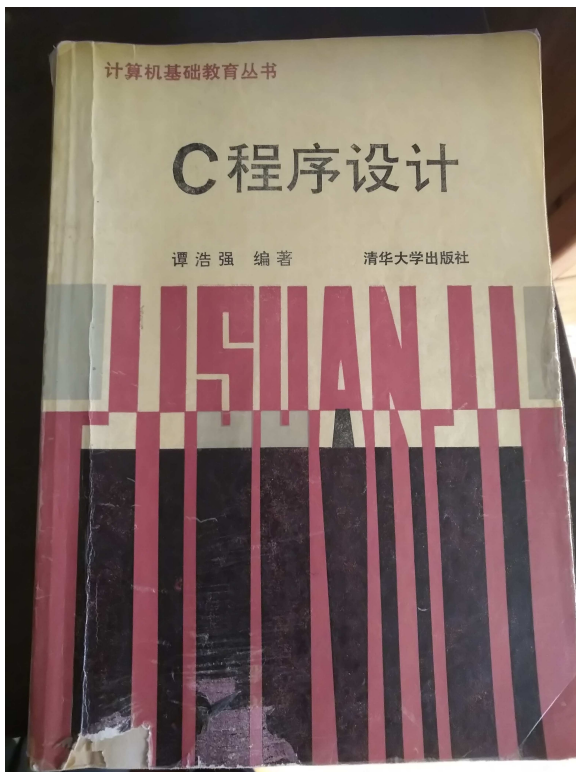


PLC的本质是个计算机。和计算机的编程语言一样，在程序中是需要用到各种变量的。

所以先从计算机高级语言说起。

不管是VB还是C++，各种编程语言都有一个比较重要的章节，讲变量的存储类型。

我一开始以为它们是在第一章第一课就讲的。还好先翻了翻书。



《C程序设计》 谭浩强， 清华大学出版社

第七章

7.8 局部变量和全局变量

7.9 动态存储变量和静态存储变量

在实际的高级语言的编程应用中，用的最多的是局部变量和全部变量，以及极少数情况下会用到局部的静态变量。有多极少数呢？我甚至想不起来应用静态变量的标准场景了。

幸亏翻了下书，书里有例子讲到了，如此珍贵，我抄下来算了：

C语言用static处理输出1到5的阶乘。

```
#include<stdio.h>
int main()
{
int i;
for(i=1;i<=5;++i)
printf("%d!=%d\n",i,fact(i));
return 0;
}
int fact(int n)
{
static int f = 1;
f=f*n;
return (f);
}
```

这里面的static int f 就是静态变量。

其中一个最大的技巧是，在函数计算5! 的时候，用到了前面计算4! 的计算结果。对早期的计算机计算资源特别宝贵的情况下，起到了节省一点点计算资源的作用。

但是，这个技巧又相当扯淡。如果你只想求5! 而对前面的1! ， 2! ， 3! ， 4! 皆不感兴趣，则这个函数反而不能用了。还逼着你非要在外面做一次循环，重复调用那个函数fact，才能真正得到5! 的答案。

一个计算功能，原生的函数还不能直接得到答案，还需要再做一次循环处理，才可以。这事儿换谁谁不烦啊？

嗯，还真有不烦的。谁不烦？

PLC不烦。

因为PLC的OB1是天生自动循环的。

只要计算结果不是需要在一个周期内立马得到答案，就可以充分使用OB1循环运行的特性，在函数中实现想要的功能。

比如上一篇文章《【万泉河】MODBUS轮询中的坑》<https://mp.weixin.qq.com/s/e2uzuNoV9481h5UgRZFI1A> 中讲到的轮询，就可以由OB1的循环调用机制本身来驱动。不需要自己做循环程序（而且也不能做，如果做的话CPU单个周期运转时间过长，会累死。）

所以，以往在高级语言中，只有概念，而鲜有用武之地的STATIC静态变量在PLC系统中，闪亮登场了，而且大放异彩。

变量作用域和生存期列表:

变量类型	地址代号	作用域	生存期
局部临时变量TEMP	L0.0	FB/FC/OB内部*	1个周期, 用完即忘***
局部静态变量 STATIC	DIX0.0**	FB内部, FC/OB没有*	所有周期
全局变量MEMORY	M0.0	整个CPU	CPU上电期间, 掉电会忘
全局静态变量DB	DB	整个CPU	掉电记忆, 可设置

备注

* 对于被嵌套调用的函数块FB/FC/OB来说, 各自定义的TEMP变量不继承, 不渗透。

** 地址区间先分给INPUT和OUTPUT, 而后才是STATIC。

*** 所有TEMP变量, 必须先写入数值后使用。不能假设初始都为0, 或者上个周期运行的结果。

所以, 不同的应用需求, 需要的变量类型是不一样的。

比如要计算 $OUT=(A+B)/(C+D)$,

在LAD或者STL 里, 通常要用1个或者2个临时变量。

```
L #A
```

```
L #B
```

```
+R
```

```
T LD10
```

```
L #C
```

```
L #D
```

```
+R
```

```
T LD14
```

```
L LD10
```

```
L LD14
```

```
/R
```

```
T #OUT
```

这里使用的L区，双字，存储REAL数据。在STEP7中可以未加定义直接使用。而在TIA PORTAL中要么提前定义，要么直接使用名字后补充定义到LOCAL变量。不再允许绝对寻址。

如果同一个函数中，还有要使用同样数据类型的地方，可以重复使用。但都是要先写入后读取。

同样的程序，使用其它的内存地址来实现当然也可以。比如可以使用MD10/MD14替代上面的LD10/LD14。运算效果完全相同。但是，对资源使用造成了浪费，也容易导致变量重复使用，会干扰其他的使用了同地址的函数块。

如果见到有人做的这样的程序，你立马可以判定：这是个新手做的。

上篇文章讲到的MODBUS轮询问题，用SCL语言的部分代码（用LAD和STL原理相同）

```
IF "FirstScan" THEN
```

```
    #cycle := 0;
```

```
    #step := 1;
```

```
    #j := -1;
```

```
END_IF;
```

```
IF #Data.MB_ADDR = 0 THEN
```

```
    #i := #i + 1;
```

```
    RETURN;
```

```
END_IF;
```

其中的#cycle，#step，#i，#j等变量，都是定义的STATIC静态变量。具体细分的话，个别变量可能可以使用TEMP临时变量，但图省事，也就不那么细了。反正最终完全封装，不会影响到外面其它函数的变量使用。

同样的道理，这些变量如果使用全局变量M，能不能同样实现目的呢？

答案是：有时候能，有时候不能！

看这个函数使用的情况。如果在系统中只调用1次，那是可以的。而如果需要调用n次，使用全局变量的话，反而不可以了！多个实例时间会乱串，会互相干扰。所以

如果不幸要多次调用，还需要复制另存，更改其中的M变量，换一批来！

上面的功能中还使用了定时器，只不过代码没有贴出来。也是同样的道理。只不过定时器比较复杂，使用的是多重背景的数据块，本质也是STATIC变量。

emmmm，我这是在做什么？

我是在重复印证《[【万泉河】好的PLC程序和坏的PLC程序的比较标准](#)》

http://www.ad.siemens.com.cn/club/bbs/post.aspx?a_id=1479565&b_id=82&b_sid=19&s_id=&num=158#anch 中所主张的 **“好的程序的标准是：不使用M中间量，不使用Timer。”**

是在推行《[【万泉河】一个完全不使用T和M全局变量的好标准的PLC程序分享计划](#)》

<https://mp.weixin.qq.com/s/W3OOg65AtSwg1Jre4jN4Bg>

为学员们组织学习材料呢！有继续想报名参加的朋友加微信：178616417，注明：参加众筹

万泉河

