

## 第六十八节： 为函数接口指针“定向”的 const 关键词。

### 【68.1 为函数接口指针“定向”的 const 关键词。】

在函数接口处的指针，是一个双向口，既可以作为“输入”也可以作为“输出”，换句话说，既能“读”也能“写”（被更改），这样一来，当你把一个数组（或者某变量）通过指针引入到函数内部的时候，当执行完此函数，这个数组的数值可能已经悄悄发生了更改（“是否被更改”取决于函数内部的具体代码），进来时是“摩托”出来后可能已变成“单车”，而实际项目上，很多时候我们只想传递数组（或者某变量）的数值，并不想数组（或者某变量）本身发生变化，这个时候，本节的主角 const 关键词就派上用场了。

只要在函数接口的指针前面加上 const 关键词，原来双向的指针就立刻变成了单向，只能输入不能输出。这个 const 有两个好处。第一个好处是方便阅读，通过 const 就知道此接口的“入口”和“出口”属性，如果你是用别人已经封装好的函数，一旦发现接口指针带了 const 标签，就足以说明这个指针只能作为输入接口，不用担心输入数据被意外修改。第二个好处是确保数据的安全，函数接口指针一旦加了 const 限定，万一你不小心在函数内部对指针所关联的数据进行了更改（“更改”就意味着“出口”），C 编译器在编译的时候就会报错让你编译失败，及时让你发现程序的 bug（程序的漏洞），这是编译器层面的一道防火墙。例子如下：

```
unsigned char ShuRu(const unsigned char *pu8Data)
{
    unsigned char a;
    a=*pu8Data; //这行代码是合法的，是指针所关联数据的“读”操作。
    *pu8Data=a; //这行代码是非法的，是指针所关联数据的“写”操作，违背 const 的约束。
    return a;
}
```

### 【68.2 例程练习和分析。】

在前面第 65 节讲函数入口的时候，用到一个求数组平均值的程序例子，这个数组是仅仅作为输入用的，不需要被更改，因此，现在借本节讲 const 的机会，为此函数的接口指针补上一个 const 关键词，让该函数更加科学规范，程序如下：

```
/*---C 语言学习区域的开始。-----*/

unsigned long PinJunZhi(const unsigned char *pu8Buffer); //指针前增加一个 const 关键词
unsigned char Gu8Buffer[4]={2,6,8,4};
unsigned long Gu32PinJunZhi;

unsigned long PinJunZhi(const unsigned char *pu8Buffer) //指针前增加一个 const 关键词
{
    unsigned long u32PinJunZhi;
    u32PinJunZhi=(pu8Buffer[0]+pu8Buffer[1]+pu8Buffer[2]+pu8Buffer[3])/4; //求平均值
    return u32PinJunZhi;
}

void main() //主函数
```

```

{
    Gu32PinJunZhi=PinJunZhi(&Gu8Buffer[0]); //不用担心 Gu8Buffer 数组的数据被意外更改。

    View(Gu32PinJunZhi); //把第 1 个数 Gu32PinJunZhi 发送到电脑端的串口助手软件上观察。
    while(1)
    {
    }
}
/*---C 语言学习区域的结束。-----*/

```

在电脑串口助手软件上观察到的程序执行现象如下：

开始...

第 1 个数  
十进制:5  
十六进制:5  
二进制:101

分析：

平均值变量 Gu32PinJunZhi 为 5。

### 【68.3 如何在单片机上练习本章节 C 语言程序？】

直接复制前面章节中第十一节的模板程序，练习代码时只需要更改“C 语言学习区域”的代码就可以了，其它部分的代码不要动。编译后，把程序下载进带串口的 51 学习板，通过电脑端的串口助手软件就可以观察到不同的变量数值，详细方法请看第十一节内容。