C语言Const的用法

Const：常量修饰符，即修饰的对象为常量

1. 函数体内修饰局部变量。

例子

Void function()

{

Const int a = 0;}

Const 要求修饰的对象是常量，不可被改变，不可被赋值，不可作为左值。

Const int a等价int const a。

以下写法错误：const int a ;

a = 0;

例子

Const int \*p;等价于const int(\*p)

Int const (\*p)

即\*p是常量，即p指向的数据是常量。

于是 p+=8; //合法

\*p=0; //非法

声明自身是常量指针的方法：让const 尽可能靠近p;

Int \* const p;

于是p+=8; //非法

\*p=0; //合法

2. 在函数声明时修饰参数

例子

Void \* memmove(void \*dst,const void \*src,size\_t len); //这是标准库中一个函数，用于按字节方式复制字符串（内存）

第一个参数，是将字符串复制到哪里去（dst），是目的地，这段内存区域必须可写。

第二个参数，是要将什么样字符串复制出去，这段内存区域只读不写。

于是，src这个指针指向的内存内所存储的数据在整个函数执行的过程中不变，即src指向的内容是常量，于是就需要用const修饰。

例子

Const char \*s = hello;

Char buf[100];

Memmove(buf, s, 6);

如果反过来写：memmove(s, buf, 6); 编译器就会报错，所以为了在我们把函数参数顺序写反的情况下能够发现错误，加入const就可以很好的帮助我们检查。

总结：const的好处，是引入了常量的概念，让我们不要去修改不该修改的内存，直接的作用就是让更多的逻辑错误在编译期就被发现。