

Linux操作系统编程之Const使用详解 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E6_93_8D_E4_BD_c103_144946.htm — const基础 如果const关键字不涉及到指针，我们很好理解，下面是涉及到指针的情况：
`int b = 500.const int* a = amp.b. [2]int* const a = amp.b. [4]`如果你能区分出上述四种情况，那么，恭喜你，你已经迈出了可喜的一步。不知道，也没关系，我们可以参考《Effective c》Item21上的做法，如果const位于星号的左侧，则const就是用来修饰指针所指向的变量，即指针指向为常量；如果const位于星号的右侧，const就是修饰指针本身，即指针本身是常量。因此，[1]和[2]的情况相同，都是指针所指向的内容为常量（const放在变量声明符的位置无关），这种情况下不允许对内容进行更改操作，如不能`*a = 3`；[3]为指针本身是常量，而指针所指向的内容不是常量，这种情况下不能对指针本身进行更改操作，如`a`是错误的；[4]为指针本身和指向的内容均为常量。另外const的一些强大的功能在于它在函数声明中的应用。在一个函数声明中，const可以修饰函数的返回值，或某个参数；对于成员函数，还可以修饰是整个函数。有如下几种情况，以下会逐渐的说明用法：
`A& a).void fun0(const A* a). void fun1() const. // fun1() 为类成员函数const A fun2().`
二 const的初始化 先看一下const变量初始化的情况
1) 非指针const常量初始化的情况：`A b.const A a = b.`
2) 指针(引用)const常量初始化的情况：`A* d = new A().const A* c = d.`或者：`const A* c = new A().` 引用：`A f. const A& e = f. // 这样作e只能访问声明为const的函数，而不能访问一般的成员函数`

; [思考1]： 以下的这种赋值方法正确吗？ `const A* c=new A(). A* e = c.` [思考2]： 以下的这种赋值方法正确吗？ `A* const c = new A(). A* b = c.` 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com