

对一级指针和二级指针不同位置加const修饰的理解

方法：看const修饰哪个部分，注意：没有const*这个运算，如果出现，const就是修饰前面部分

1. `char *const p;`

const修饰p，故p是一个指向char的指针常量，值可以改变，但是指向不能改变，也就是可以写成* p=... 但是不能写成p=...。

2. `const char* p;`
`char const* p;`

这两个都是一个意思=>常量指针，即值不能改变，指向可以改变，也就是可以写成p=...，不能写成* p=...。

3. `const char** p;`
`char const** p;`

这两个等价，上面的式子说明const修饰char，对指针p来说，没有任何限制，因为p的解引用是* p，没有const修饰* p，对* p来说，限制其值不能变，指向可以变，因为const正好修饰的就是* p的解引用。

4. `char* const* p;`

对p来说，没有const *运算，故const修饰前面的char *，故p的值不能改变，指向可以改变，对 * p来说，const修饰 * p，所以 *p的指向不能变，但是值可以改。

5. `char **const p;`

对p来说，const修饰p，故p的指向不能变，对*p来说，没有任何限制。