
嵌入式开发

void指针和const

创客学院 小美老师

- ▶ 1 / void指针-掌握
- 2 / const修饰符-掌握
- 3 / 总结与思考

void指针

void指针是一种不确定数据类型的指针变量，它可以通过强制类型转换让该变量指向任何数据类型的变量

一般形式为： `void * <指针变量名称> ;`

对于void指针，在没有强制类型转换之前，不能进行任何指针的算术运算

程序举例

编程实现：使用void指针遍历一维数组



案例

```
void * malloc(size_t size);
```

```
int pthread_create(pthread_t *thread, const  
pthread_attr_t *attr,void *(*start_routine)(void *),  
void *arg);
```

```
void qsort(void *base, size_t nmemb, size_t size,  
int(*compar)(const void *, const void *));
```

const变量

常量化变量的值

一般说明形式如下：

➤ `const <数据类型> 变量名 = [<表达式>];`

常量化变量是为了使得变量的值不能修改

变量有const修饰时，若想用指针间接访问变量，指针也要有const修饰。const放在指针声明的什么位置呢？

const修饰指针

常量化指针目标表达式

一般说明形式如下：

➤ `const <数据类型> * <指针变量名称> [= <指针运算表达式>];`

常量化指针目标是限制通过指针改变其目标的数值，但<指针变量>存储的地址值可以修改

const修饰指针

常量化指针变量

一般说明形式如下：

➤ `<数据类型> * const <指针变量名称> [= <指针运算表达式>];`

使得<指针变量>存储的地址值不能修改。但可以通过 *<指针变量名称> 可以修改指针所指向变量的数值

const修饰指针

常量化指针变量及其目标表达式

一般说明形式如下：

➤ `const <数据类型> * const <指针变量名> = <指针
运算表达式> ;`

常量化指针变量及其目标表达式，使得既不可以修改<指针变量>的地址，也不可以通过*<指针变量名称>修改指针所指向变量的值

总结与思考

主要介绍了const修饰指针的用法，有三种情况。还介绍了void指针的用法

思考

- void指针有什么作用？
- main函数是否可以带参数(涉及到const指针)？

扫一扫，获取更多信息



THANK YOU