

函数指针

函数原型在声明的时候无需说明变量名称。

一个函数也有属于自己的类型，那么对应的就有属于自己的指针。函数的关键就是返回值和参数，这也决定了函数指针对应的类型，就由这三个决定。考虑下段代码：

```
1 void test_for_fucptr(){
2     void (*p)() = NULL; //创建一个返回值为void，参数为空的指针
3     p = test_for_purevir;
4     (*p)();
5 }
6 //一定要注意的是指针的括号
```

函数指针的作用就在于函数之间的调用，以及使用一个类型的指针，访问、调用参数、返回值均相同的函数。

函数指针在函数之间的调用示例1：

```
1 void test_for_fucptr(void(*p)()){
2     (*p)();
3 }
4 int main(){
5     void (*p)() = NULL; //创建一个返回值为void，参数为空的指针
6     p = test_for_purevir; //注意，这里不给参数
7     test_for_fucptr(p);
8     return 0;
9 }
```

示例2：函数指针的调用参数只给类型，不给实际值，只有解引用的时候才会给实际值

```
1 int add(int a, int b){
2     return a+b;
3 }
4 int sub(int a, int b){
5     return a-b;
6 }
7 void func(int e, int d, int(*f)(int, int)){
8     printf("%d\n", (*f)(e,d));
9 }
10 int main(){
11     int (*p)(int,int) = NULL;
12     p = add;
13     func(2,3,p);
14     func(2,3,sub);
15     return 0;
16 }
```