

函数指针

概述

函数指针是指向函数的指针变量。

通常我们说的指针变量是指向一个整型、字符型或数组等变量，而函数指针是指向函数。

函数指针可以像一般函数一样，用于调用函数、传递参数。

函数指针变量的声明：

```
typedef int (*fun_ptr)(int,int); // 声明一个指向同样参数、返回值的函数指针类型
```

实例

以下实例声明了函数指针变量 p，指向函数 max：

```
#include <stdio.h>

int max(int x, int y)
{
    return x > y ? x : y;
}

int main(void)
{
    /* p 是函数指针 */
    int (* p)(int, int) = & max; // &可以省略
    int a, b, c, d;

    printf("请输入三个数字:");
    scanf("%d %d %d", & a, & b, & c);

    /* 与直接调用函数等价，d = max(max(a, b), c) */
    d = p(p(a, b), c);
}
```

```
printf("最大的数字是: %d\n", d);

return 0;
}
```

回调函数

函数指针作为某个函数的参数

函数指针变量可以作为某个函数的参数来使用的，回调函数就是一个通过函数指针调用的函数。简单讲：回调函数是由别人的函数执行时调用你实现的函数。

“

你到一个商店买东西，刚好你要的东西没有货，于是你在店员那里留下了你的电话，过了几天店里有货了，店员就打了你的电话，然后你接到电话后就到店里去取了货。在这个例子里，你的电话号码就叫回调函数，你把电话留给店员 就叫登记回调函数，店里后来有货了叫做触发了回调关联的事件，店员给你打电话叫做调用回调函数，你到店里去取货叫做响应回调事件。

”

实例

实例中 `populate_array` 函数定义了三个参数，其中第三个参数是函数的指针，通过该函数来设置数组的值。实例中我们定义了回调函数 `getNextRandomValue`，它返回一个随机值，它作为一个函数指针传递给 `populate_array` 函数。`populate_array` 将调用 10 次回调函数，并将回调函数的返回值赋值给数组。

```
#include <stdlib.h>
#include <stdio.h>

// 回调函数
void populate_array(int *array, size_t arraySize, int (*getNextValue)(void))
{
    for (size_t i=0; i<arraySize; i++)
        array[i] = getNextValue();
}
```

```
}

// 获取随机值
int getNextRandomValue(void)
{
    return rand();
}

int main(void)
{
    int myarray[10];
    /* getNextRandomValue 不能加括号，否则无法编译，因为加上括号之后相当于传入此参数时传入了 int，而不是函数指针*/
    populate_array(myarray, 10, getNextRandomValue);
    for(int i = 0; i < 10; i++) {
        printf("%d ", myarray[i]);
    }
    printf("\n");
    return 0;
}
```