

58 Function Pointers in C++

本节会讲的是原始风格的函数指针（来自C语言）

1. 什么是函数指针？

Function pointer（函数指针）是将一个函数赋值给一个变量的方法，从这里拓展，你还可以将函数作为参数传递给其它函数。实际上函数可以用来做很多事，这创造了很多有趣而复杂的逻辑，如果不用函数指针就会非常麻烦。

auto 关键字对于函数指针之类的东西非常有用：

```
C++  
  
void HelloWorld()  
{  
    std::cout << "HelloWorld" << std::endl;  
}  
  
int main()  
{  
    auto function = HelloWorld;    // 注意这里没有括号  
  
    std::cin.get();  
}
```

具体来说，这里就像是带了 **&** 取地址符号一样，我们得到了这个函数的内存地址，可以想象一下当你编译你的代码时，每个函数都被编译成CPU指令，它们就在我们的二进制文件中。我们要做的就是当这个函数被调用时，我们检索要执行的指令的位置，我们没必要写成 **&HelloWorld** 因为这个里有个**隐式转换**

```
auto function:void(*)() = HelloWorld;  
  
function();  
function();
```

auto 的实际类型：

```
C++  
  
void(*function)() = HelloWorld;    // 同 auto function = HelloWorld;
```

或者创建一个 *alias*：

```
C++  
  
typedef void(*HelloWorldFunction)();  
  
HelloWorldFunction function = HelloWorld;  
  
function();
```

带参数的函数：

```
C++

void HelloWorld(int a)
{
    std::cout << "HelloWorld:" << a << std::endl;
}

int main()
{
    typedef void(*HelloWorldFunction)(int);

    HelloWorldFunction function = HelloWorld;

    function(3);    // HelloWorld:3
}
```

2. 为什么要使用函数指针

```
C++

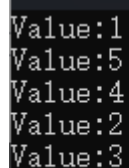
void PrintValue(int value)
{
    std::cout << "Value:" << value << std::endl;
}

void ForEach(const std::vector<int>& values, void(*func)(int))
{
    for (int value : values) //这里是一个范围-based for 循环。
        func(value);
}

//在这个语法中, `int value` 是定义的循环变量, `values` 是要遍历的容器。循环的含义是: 对于容器
//`values` 中的每个元素, 将其值赋给变量 `value`, 然后在循环体中执行相关操作。这种语法简化了传统的索引或
//迭代器遍历方式, 使得代码更加简洁和易读。

int main()
{
    std::vector<int> values = { 1,5,4,2,3 };
    ForEach(values, PrintValue);

    std::cin.get();
}
```



```
Value:1
Value:5
Value:4
Value:2
Value:3
_
```

这种方法可以告诉函数，让它在特定的时间做这个事，不过这方法做的事有点多。所以再用一个方法：

3. lambda

我们如果不想这样定义PrintValue这个函数，因为这么一点信息不需要专门额外写一个函数，特别时如果我们只打算在ForEach函数内部使用的话，所以我们可以用一个叫做`lambda`的东西。

lambda本质上就是一个普通函数，只不过不像普通函数这样做声明，它是我们的代码在过程中生成并且**用完即弃**的函数：

```
C++  
  
void ForEach(const std::vector<int>& values, void(*func)(int))  
{  
    for (int value : values)  
        func(value);  
}  
  
int main()  
{  
    std::vector<int> values = { 1,5,4,2,3 };  
    ForEach(values, [](int value) {std::cout << value << std::endl; });  
  
    std::cin.get();  
}
```

这里的 `[]` 叫做`capture method`（捕获方式），也就是如何传入传出参数，后面会介绍更多。