

C++之 static 及其文件作用域

一、面向过程设计中的 static

1、静态全局变量

在全局变量前，加上关键字 `static`，该变量就被定义成为一个静态全局变量。我们先举一个全局变量和静态全局变量的例子，例如，在文件 A 中定义静态全局变量 `i` 和全局变量 `j`：

[cpp] view plain copy

```
1.  #include<iostream>
2.  using namespace std;
3.  static int i = 5;
4.  int j = 3; //默认为 extern
5.  int main()
6.  {
7.      /*code*/
8.  }
```

全局变量 `j` 和静态变量 `i` 都存放于程序的全局数据区域，它们的生存周期都是程序的整个运行期，但是 `j` 的作用域为全局作用域，可以通过 `extern` 在其他文件中使用，而 `i` 的作用域为文件作用域，只能在文件 A 中使用，例如在文件 B 中：

[cpp] view plain copy

```
1.  extern int j; //OK
2.  extern int i; // error: i 在文件 B 中不可见
3.  int a = j; // OK
4.  int b = i; // error
```

也就是说，在声明全局的 `static` 变量时，`static` 没有改变它的生存周期，也即存储位置（因为全局变量本来就存储在全局数据域），而是将变量的作用域限制在当前文件中。

总：静态全局变量 `i` 有以下特点：

- 该变量在全局数据区分配内存；

- 未经初始化的静态全局变量会被程序自动初始化为 0（自动变量的值是随机的，除非它被显式初始化）；

- 静态全局变量在声明它的整个文件都是可见的，而在文件之外是不可见的；