

杂项讨论

条款53: 不要轻视编译器的警告

- 严肃对待编译器发出的警告信息。努力在你得编译器的最高警告级别下争取“无任何警告”的荣誉。
 - 不要过度依赖编译器的报警能力，因为不同的编译器对待事情的态度并不相同。一旦移植到另一个编译器上，你原本依赖的警告信息有可能消失。
-

许多程序员会忽略编译器警告，但是有时候警告会产生运行时异常。例如下面这段代码：

```
class B {
public:
    virtual void f() const {
        std::cout << "i'm B" << std::endl;
    }
};

class D : public B {
public:
    virtual void f() {
        std::cout << "i'm D" << std::endl;
    }
};

B* pb = new D;
pb->f();
```

编译器可能会说 `warning: D::f() hides virtual B::f()`，看上去这是理所应当的事情，但是实际上它想告诉你，这并不是重写操作，这种名称掩盖会导致最终结果与期望不一的情况。

尽管写出一个完全没有警告的代码是十分困难的，但是你应该去理解这些警告的含义，并在完全理解后来判断是否应该忽略这些警告，否则造成错误后很有可能就是警告造成的。

条款54: 让自己熟悉包括 TR1 在内的标准程序库

条款55: 让自己熟悉 Boost