

LongXiaJun30天读书笔记分享计划  
Day 30: C++

2019.1.5 Saturday. By KuKuXia@github.com

严肃对待编译器发出的警告信息，努力在你的编译器的最高（最严苛）警告级别下争取“无人恶化警告”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。



## EC条款53：不要轻忽编译器的警告



EC条款54: 让自己熟悉包括TR1在内的标准程序库



EC条款55: 让自己熟悉Boost [🔗](#)

Boost是一个社群，也是一个网站。致力于免费、源码开放、同僚复审的C++程序库开发。Boost在C++标准化过程中扮演极具影响力的角色。

Boost提供许多TR1组件实现品，以及其他许多程序库



EC条款52: 写了placement new也要写placement delete

如果operator new接受的参数除了一定会有那个size\_t之外还有其他, 这便是个所谓的placement new

如果operator delete接受的参数除了一定会有那个size\_t之外还有其他，这便是个所谓的placement delete

当你写一个placement operator new, 请确定也写出了对应的placement operator delete. 如果没有这样做, 你的程序可能会发生隐微而时断时续的内存泄漏

当你声明placement new 和 placement delete, 请确定不要无意识(非故意)地遮掩了它们的正常版本



C++ STL

```
uninitialized_copy( ForwardIt uninitialized_copy( InputIt first, InputIt last, ForwardIt d_first )
```

复制来自范围 [first, last) 的元素到始于 d\_first 的未初始化内存

```
ForwardIt uninitialized_copy_n( InputIt first, Size count, ForwardIt d_first)
```

从始于 first 的范围复制 count 个元素到始于 d\_first 的未初始化内存区域

```
void uninitialized_fill( ForwardIt first, ForwardIt last, const T&
value )
```

uninitialized\_fill 复制给定的 value 到以 [first, last) 定义的未初始化内存区域

ForwardIt uninitialized fill n( ForwardIt first, Size count, const T& value )

复制给定值 value 到始于 first 的未初始化内存区域的首 count 个元素

若初始化中抛异常，则以未指定顺序销毁已构造的对象