

瘦身前后兼谈C 语言进化(5) PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/469/2021_2022__E7_98_A6_E8_BA_AB_E5_89_8D_E5_c67_469825.htm (4) 这

里function_invoker是用的type-erase手法，具体可参见我以前写的boost.any源码剖析，或上篇讲auto的，或《C Template Metaprogramming》（内有元编程慎入！）。type-erase手法是像C 这样的弱RTTI支持的语言中少数真正实用的手法，某种程度上设计模式里面的adapter模式也是type-erase的一个变种。如果还觉得不够的话，可以参考variadic-templates的主页，上面的variadic templates proposal中带了三个tr1实现，分别是tuple，bind，function，当然，variadic-templates的好处远远不仅仅止于这三个实现，从本质上它提供了一种真正直接的表达意图的工具，完全避开了像下面这种horrible的workaround：templatecons(T1amp., const null_typeamp., const null_typeamp., const null_typeamp., const null_typeamp.): head(t1) {} tuple的C 98实现，代码近千行。利用variadic-templates实现，代码仅百行。和这种更horrible的workaround

```
: template_bi::bind_t::type>BOOST_BIND(boost::type, F f, A1 a1, A2 a2, A3 a3, A4 a4, A5 a5, A6 a6){typedef typename _bi::list_av_6::type list_type.return _bi::bind_t(f, list_type(a1, a2, a3, a4, a5, a6)).}
```

100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com