2022/4/7 12:32 (3 封私信 / 80 条消息) 为什么常量左值引用可以绑定到右值? - 知乎 知乎 首页 会员 发现 等你来答 和体育生谈恋要什么感受 Q 提问 👂 💽

为什么常量左值引用可以绑定到右值? 右值并不能现到地址,所以左值引用不能频定到右值,但是常量左值引用为什么就可以呢?其中的数据在内存中的存储状况是怎么样的一个情况? 

关注者 被浏览 49 15,192

默认排序 ≎

29 人赞同了该回答 C++ 2.0 前,引用没有左右值的限制。 The reason to allow references to be initialized by non-Ivalues was to allow the distinction between call-by-value and call-by-reference to be a detail specified by the called function and of no interest to the caller. For **const** references, this is possible; for non-**const** references it is not. For Release 2.0 the definition of C++ was changed to reflect this.

允许引用绑定非左值的初衷在于「让传值还是传引用成为函数本身的细节,调用者不用去关心」, 这一点,在 const 引用上可以达到,而非 const 引用调用者没法不关心传值还是传引用,所以干脆 禁止掉了。C++ 20 特意为此在语言方面作出了调整。 其实不仅是无法实现初衷,非 const 引用绑定非左值还会让 C++ 变「坑」,Bjarne 意识到了这样 做的问题: I made one serious mistake, though, by allowing a non-const reference to be initialized by a non-Ivalue. For example: void incr(int& rr) { rr++; }

void g()

Because of the difference in type the int& cannot refer to the double passed so a temporary was generated to hold an int initialized by ss's value. Thus, incr() modified the temporary, and the result wasn't reflected back to the calling function. 如果允许任何引用绑定非左值,会带来代码合法,但是逻辑错误的很难察觉的「坑」。 除了上述「初衷」外,支持绑定非左值还有一个硬性需求: It is important that const references can be initialized by non-Ivalues and Ivalues of type that require conversion. In particular, this is what allows a Fortran function to be called with a constant: extern "Fortran" float sqrt(const float&);

void f()
{
 sqrt(2); // call by reference
} 为了支持 Fortran 足够服人了吧? 以上引用出自《C++设计与演化<sup>Q</sup>》。 ▲ 赞同 29 ▼ ● 3 条评论 **4** 分享 ★ 收藏 ● 喜欢 ··· 收起 ^

因为以前根本就没有左值引用的说法。所有的blah改都叫引用,命名上根本不关心绑的是左值还是 右值。 所以这个规则应该叫"常引用可以绑定到右值",这样听起来就没啥问题了。 ▲赞同7 ▼ ●3条评论 4分享 ★收藏 ●喜欢 …

看来题主还是懒得看我答案啊,真的挺无奈的。我挺不喜欢帮人研究,不过答了就要敬业: 例子1

const int& a = 53; 009A1808 mov dword ptr [ebp-18h],35h // 53 被放到了栈里 009A180F lea eax.[ebp-18h] 009A1812 mov dword ptr [a],eax

int main() {
 std::cout << a << std::endl;</pre>

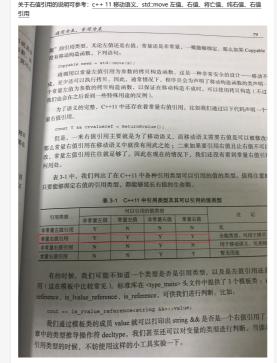
std::cout << a << std::endl; 010625C0 push offset std::endl<char,std::char\_traits<char> > (01061064h) 010625C5 mov edi,esp 010625C7 mov eax,dword ptr [a (0106A138h)] // 53 被放在了堆上

自己写一句然后看看反汇编代码。

▲ 赞同 15 ▼ ● 4 条评论 **4** 分享 ★ 收藏 ● 喜欢 ··· 收起 ^ 张<del>东升</del> 程序员/c++/windows/lua 8 人赞同了该回答 常量左值引用 在c98中被称为"万能"引用类型

他可以接受: 非常量左值, 常量左值, 右值对其初始化例如: const int e = 100; const int & c = d;//接受一个非常量左值 const int & v = e;//接受一个常量左值 const int & z = 3 + 4;//接受一个右值

关于右值引用的说明可参考: c++ 11 移动语义、std::move 左值、右值、将亡值、纯右值、右值 引用



发布于 2017-10-25 19:03 ▲ 赞問8 ▼ ● 3 条评论 4分享 ★ 收藏 ● 高欢 … 收起 ∧ kyay006 详见《C++ Primer Plus》(第六版) 262页 ▲ 赞同 1 ▼ ● 添加评论 4 分享 ★ 收蔵 ● 喜欢 … 不想写的。但是看了很多类似的问题。别的答案一些垃圾半吊子 扯了半天 什么 const 引用安全, , 真是吐了, 人要有自己思想啊。还是本答案靠进回答多编辑于 2019-09-01 18:50

1/2

▲ 赞同 ▼ ● 添加评论 增分享 ★ 收藏 ● 喜欢 …

/ 写图答 1 个回答被折叠 (为什么?)

如何最简单、通俗地理解C++的常量? 6 个回答 常量的值类型是否属于右值呢? 6 个回答 这三种定义C++常量的方式孰优孰劣? 9 个回答 为什么C++无法修改全局静态常量? 10 个回答 相关推荐 C++标准库 (第 2 版) (德)尼古拉约祖蒂斯 67 人读过 Effective STL 中文版: 50 条有效使用 STL 的经验... 36 人读过 Boost 程序库完全开发指 南: 深入 C++ J 准 J 标... 62 人读过 刘看山·知乎指南。知乎协议·知乎隐私保护指引应用·工作·申请开通知乎机构号 侵权举报。网上有害信息举报专区 京(ICP 至 110745 号 京 ICP 备 13052560 号 - 1 ● 京公网名名 1101082020088 号 互联网络品信息服务资格证书 (京) - 非经营性 - 2017 - 0067 服务执线: 400-919-0001 违法和不良信息举报: 010-82716601 举服婚籍: jubao@zhihu.com 儿童色情信息举报专区 信息安全漏洞反馈专区 内容处业人用违法违规行为举报 证酬中心 - Investor Relations 联系我们 © 2022 知乎

https://www.zhihu.com/question/40238995

2/2

https://www.zhihu.com/question/40238995