图形用户界面, 文本, 应用程序, 信件, 电子邮件

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

通过合理安排类或结构体中数据成员的声明顺序，从具有最大对齐要求的类型开始定义，可以减少编译器插入的填充字节，从而减小对象的内存占用。

当我们讨论缓存友好性的时候，还应该提到的是，将经常一起使用的多个数据成员彼此相邻放置可能是有益的。

RAII 是一种著名的 C++ 技术，即资源获取即初始化（Resource Acquisition Is Initialization），其中资源的生命周期由对象的生命周期控制。

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

小对象优化是指在字符串较小的情况下，在栈上分配内存，而不是在堆上

文本

AI 生成的内容可能不正确。

在 C++17 中，通过引入额外的间接层解决了使用不同分配器时最终得到不同类型的问题；std::pmr 命名空间下的所有标准容器都使用相同的分配器，即 std::pmr::polymorphic\_allocator，它将所有的分配 / 释放请求转发给一个内存资源类。所以，我们不必编写新的自定义内存分配器，而是可以使用通用的多态内存分配器 std::pmr::polymorphic\_allocator，并编写新的自定义内存资源，在构造时将这些资源传递给多态分配器。

多态内存分配器其实就是创建了一个栈上的内存池，然后这个内存池可以存储任意类型的对象，缓冲区用完后，才会在堆上分配内存