C++八股文

叶茂林 2024年1月12日星期五

**New、delete和malloc、free的区别？**

New和delete是C++的运算符，malloc和free是C的标准库里面的函数，malloc和free用来分配和释放内存，new和delete除了完成分配和释放内存还会自动调用对象的构造函数和析构函数。

**多次delete同一个对象会怎么样？**

Delete对象会调用对象的析构函数进行内存的释放，指向该对象的指针会变成野指针，如果不将野指针赋值为空指针的话，再次delete会出现未定义的行为。

**Delete空指针会怎么样？**

Delete空指针是安全的，delete内部有对指针判空，不会怎么样。

**C++什么时候会出现内存泄漏？**

申请的内存用完没有释放，发生异常跳过释放内存的代码，容器用完没有释放，delete与delete[]用混，还有就是share指针相互引用导致析构的时候引用无法降为0出现的内存泄露。

**说一下C++的四种智能指针？**

C++的四种智能指针包括C++11新增的unique\_ptr、share\_ptr、weak\_ptr和C++11抛弃使用的auto\_ptr。

auto和unique\_ptr一样是独占所有权的，保证同一时刻只有一个智能指针可以指向该对象，auto在转换资源所有权的时候是通过reset和release来重载赋值运算符实现的，先是release将auto本指针赋值为了0，然后reset将资源转换给新的指针，这个时候如果再通过原来的指针进行资源的访问就会出现访问空指针的问题，会触发未定义行为，而unique不允许直接复制，需要move移动赋值。

Share\_ptr是共享式的，允许多个智能指针指向相同的资源，每次构造和拷贝和赋值的时候引用计数会加一，当析构的时候会引用减一，引用为0的时候会释放资源。

然而当share指针相互引用的时候会出现析构的时候引用无法降为0的问题，weak\_ptr就是用来解决这个问题的，weak指针是弱引用，不会增加资源的引用计数。