

C 中的指针(三)智能指针 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/134/2021\\_2022\\_C\\_\\_\\_E4\\_B8\\_AD\\_E7\\_9A\\_84\\_E6\\_c97\\_134370.htm](https://www.100test.com/kao_ti2020/134/2021_2022_C___E4_B8_AD_E7_9A_84_E6_c97_134370.htm) Smart Pointer是C 中的一个题目，要说清楚他的所有好处很需要费点力气。我就一个功能一个功能的说。有我理解不透的地方希望大家指点。

1.copy-to-write 当生成一个C object的时候如果这个class很大，这个object会占用很多空间。那么每生成一个就占用一片空间，这样会占用很多系统资源。同时降低效率。一个解决方法就是对用拷贝构造函数生成的object，让他不存储数据，而只存储一个指向原来object数据的指针。这样空间就节省了很多。但问题在于这样两个object完全联结在了一起。如果修改了其中一个，另一个也跟着变了。所以这种方法不可取。这里讲的copy-to-write技术就是解决这类问题的方法。当通过引用一个已有object去拷贝构造新object时，新object只有一个指向已有object的指针。这两个object共享数据。直到其中一个需要修改数据的时候，再把这两块数据分离。这里举一个最简化的例子。假设一个class叫CLargeObject，里面存有很多数据。我们用一个inner class来把所有数据放在一起，叫CData。CData里面存有大量数据，例如一个数据库。这里用最简单的模型来表示，假设只有一个整数int m\_nVal. CData里面需要包含另一个变量。叫作索引数目(reference count)。它记录了指向这个CData object的来自CLargetObject类的指针各数。也就是说，总共有多少CLargeObject的object正在引用着当前的CData object。 class CLargeObject { private: struct CData { private: int m\_nVal. int m\_nReferenceCount. } }. 对于每

个CLargeObject的object，我们用一个CData类的指针来指向其数据。CData \*m\_pData. CLargeObject至少有两个构造函数。第一个是标准的构造函数，初始化其数据。这时数据是唯一的，所以必须新生成一个CData的object来存储数据。

```
CLargeObject::CLargeObject(int nVal) { m_pData = new  
Data(nVal). } 而对于CData类的构造函数而言，初始化他的  
CLargeObject是第一个指向他的，这一时刻索引数  
目m_nReferenceCount是1。 CLargeObject::Data::Data(int nVal) :  
m_nVal(nVal), m_nReferenceCount(1) {} CLargeObject的第二个  
构造函数是拷贝构造(copy constructor)。这样生成的object不  
需要有新的数据，和已有的object共享数据就可以了。这是索  
引数目需要加1。表示又有一个object指向当前的CData了。
```

```
CLargeObject::CLargeObject(const CLargeObject amp.  
CLargeObject::operator = (const CLargeObject& amp. ob) // copy  
assignment { ob.m_pData->m_nReferenceCount . if
```

```
(--m_pData->m_nReferenceCount == 0) 0delete m_pData.
```

```
m_pData = ob.m_pData. return *this. } 再来看看如何
```

对CLargeObject里面的数据进行真正的修改。这样就一定需要对当前的object独立操作了，否则就影响到了其它指向同一块数据的CLargeObject。这样CData类需要一个新的函数，生成只用于当前CLargetObject的数据。如果当前的引用数目是1，那么当然这个CData就是只用于这个CLargeObject的了。否则就重新new一个CData返回。 Data\*

```
CLargeObject::CData::get_own_copy() // clone if necessary { if
```

```
(m_nReferenceCount==1) return this. m_nReferenceCount--.
```

```
return new Data(m_nVal). } CLargeObject修改前用这个函数得
```

到唯一的object，然后对它赋值。 void CLargeObject::SetVal(int nNewVal) { m\_pData = m\_pData->get\_own\_copy(). m\_pData->m\_nVal = nNewVal. } 对于所有可能改变CData值的操作，都需要用这种方法。 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](http://www.100test.com)