参考 – 如何释放 Vector 容器闲置的空间

版权信息

1. 本文可能的出处：
   A、轩辕高端 IT 培训中心技术人员自主撰写
   B、相关书籍摘录
   C、互联网上摘录
   D、其他人员或组织赠予
2. 版权声明：

3. 原文作者：
   作者：程磊 kwarph@gmail.com
   译者：
4. 原文链接

一、std::vector 的容量操作

C++ STL 容器 vector 对于容量的操作是只增不减，如下面的代码：

```
vector<int> v;
v.push_back(12); // capacity: 1
v.push_back(22);// capacity: 1 * 2 = 2
v.push_back(32);// capacity: 2 * 2 = 4
cout << v.capacity() << endl; // 4

v.insert(v.begin(), 12, 86);
cout << v.capacity() << endl; // 15

// 删除第 3 个到倒手第 2 个元素（不含）之间的所有元素
v.erase(v.begin() + 2, v.end() - 2);
cout << v.size() << endl; // size: 4
cout << v.capacity() << endl; // 15 <--- #1

v.clear();
cout << v.capacity() << endl; // 15 <--- #2

v.reserve(0);
cout << v.capacity() << endl; // 15 <--- #3
```

由上面的代码可以看出，不论是删除 vector 中的元素(#1)、甚至 clear 整个容器的内容(#2)、或显式将 capacity 保留为 0(#3)，都无法缩减 vector 中闲置的空间。

二、std::vector 复制构造不会复制 capacity

如下列代码：

```
vector<int> v2;
v2.push_back(12);
v2.push_back(28);
cout << v2.capacity() << endl; // 2

v2.reserve(120);
cout << v2.capacity() << endl; // 120
cout << v2.size() << endl; // 2

vector<int> v3(v2);
cout << v3.capacity() << endl; // 2  <-- #1
cout << v3.size() << endl; // 2
```

如上所示，v3 的 capacity 只是 2(#2)，即 v2 中的元素个数。

三、通过复制构造和 swap 来释放 vector 容器闲置的内存空间

如下代码：

```
vector<int> v2;
v2.push_back(12);
v2.push_back(28);
cout << v2.capacity() << endl; // 2

v2.reserve(120);
cout << v2.capacity() << endl; // 120
cout << v2.size() << endl; // 2

vector<int> (v2).swap(v2); // <-- #1
cout << v2.capacity() << endl; // 2  <-- #2
```

为什么可以缩减 v2 的 capacity？

1. vector<int> (v2)调用 vector 的复制构造函数，用 v2 中的元素来构造一个新的、临时对象（无名对象）；
2. 由于是复制构造，所以新的、临时对象的 capacity 是 v2 的元素的个数，所以为 2；
3. 由于成员函数 swap()交换两个容器的一切：包括所有迭代器、size、所有元素甚至 capacity；
4. 经过 swap()后，v2 的 capacity 变成新的、临时对象的 capacity，也即 2，对应的：临时对象的 capacity 变成 120;

5. 由于 vector<int> (v2)创建的临时对象在 vector<int> (v2).swap(v2);这个语句结束后销毁，至此 v2 的 capacity 为 2，原先闲置的空间(120-2 个元素的空间)被释放（随着临时容器对象的销毁而释放）。

同理，完全清除一个 vector 的所有存储：

```
vector<int> v2;
v2.push_back(12);
v2.push_back(28);
cout << v2.capacity() << endl; // 2

v2.reserve(120);
cout << v2.capacity() << endl; // 120
cout << v2.size() << endl; // 2

vector<int> ().swap(v2); // <-- #1
cout << v2.capacity() << endl; // <-- #2
```

1. #1 首先创建一个临时（空）容器，然后与 v2 进行 swap。