暨南大学本科实验报告专用纸(附页)

基于增长数组的`vector`

课程名称_____数据结构_____成绩评定_____
实验项目名称__基于增长数组的`vector`__指导老师___干晓聪___
实验项目编号__02__实验项目类型__设计性__实验地点_数学系机房_
学生姓名___郭彦培___学号___2022101149___
学院_信息科学技术学院_系_数学系_专业_信息管理与信息系统_
实验时间_2024 年 6 月 13 日上午_~__2024 年 7 月 13 日中午___

1. 实验目的

实现基于增长数组的类 STL `vector`类，提供尾端插入与随机访问和迭代器。

2. 实验环境

计算机：PC X64

操作系统：Windows + Ubuntu20.0LTS

编程语言：C++：GCC std20

IDE：Visual Studio Code

3. 程序原理

增长数组会周期性的申请连续的内存，并将以往的数据移动到新申请的内存中。其倍增的特性保证了其均摊的插入复杂度在$\mathcal{O}(1)$，其连续的性质保证了随机访问的速度。

4. 程序代码

4.1. memDeleteTest.cpp

```cpp
#include <iostream>
#include <new>
#include <stdlib.h>
using namespace std;

class testClass{
public:
  int a = 0;
  testClass(){a=1;};
  ~testClass(){cout << "Distroy TestClass\n";};
};
int main()
{
  testClass * arr = new testClass[10];
  cout << "Finish Alloc\n";
  for(int i = 0;i < 10;i ++)
    arr[i].~testClass();
  if(arr)
    //delete[] arr;
    ::operator delete[](arr);
  else cout << "nullPtr\n";
  cout << "Finish Delete\n";
  return 0;
}
```

4.2. _PRIV_TEST.cpp

```cpp
#define DS_TOBE_TEST vector

#define _PRIVATE_DEBUG

#include "Dev\02\myVector.h"

#include <iostream>
#include <math.h>
#include <vector>

using namespace std;

using TBT =  myDS::DS_TOBE_TEST<int>;

void accuracyTest() {//结构正确性测试

    TBT tc = TBT();
    for(;;)
    {
```

```
        string op;
        cin >> op;
        if(op == "clr") { //清空
            tc.clear();
        } else if(op == "q") //退出测试
        {
            return;
        } else if(op == "pb")//push_back
        {
            int c;
            cin >> c;
            tc.push_back(c);
        // } else if(op == "pf")//push_frount
        // {
        //     int c;
        //     cin >> c;
        //     tc.push_frount(c);
        } else if(op == "at")//随机访问
        {
            int p;
            cin >> p;
            cout << tc[p] << "\n";
        // } else if(op == "delEL")//删除所有等于某值元素
        // {
        //     int p;
        //     cin >> p;
        //     cout << tc.erase(p) << "\n";
        // } else if(op == "delPS")//删除某位置上的元素
        // {
        //     int p;
        //     cin >> p;
        //     cout << tc.erase(tc.get(p)) << "\n";
        } else if(op == "iterF") //正序遍历
        {
            // tc.innerPrint();
            cout << "Iter with index:\n";
            for(int i = 0;i < tc.size();i ++) cout << tc[i] << " ";cout << "\n";
            cout << "Iter with begin end\n";
            for(auto x = tc.begin();x != tc.end();x ++) cout << (*x) << " ";cout << "\n";
            cout << "Iter with AUTO&&\n";
            for(auto x:tc) cout << x << " ";cout << "\n";
        } else if(op == "iterB") //正序遍历
        {
            // tc.innerPrint();
            cout << "Iter with index:\n";
            for(int i = 0;i < tc.size();i ++) cout << tc[tc.size()-1-i] << " ";cout << "\n";
```

```
            cout << "Iter with begin end\n";
            for(auto x = tc.rbegin();x != tc.rend();x ++) cout << (*x) << " ";cout << "\n";
            // cout << "Iter with AUTO&&\n";."\n";
        } else if(op == "mv")//单点修改
        {
            int p;
            cin >> p;
            int tr;
            cin >> tr;
            tc[p] = tr;
        } else if(op == "")
        {

        } else {
            op.clear();
        }
    }
}



void memLeakTest() {//内存泄漏测试
    TBT tc = TBT();
    for(;;){
        tc.push_back(1);
        tc.push_back(1);
        tc.push_back(1);
        tc.push_back(1);
        tc.clear();
    }
}

signed main()
{
    accuracyTest();
    // memLeakTest();
}
```

5. 测试数据与运行结果

运行上述`_PRIV_TEST.cpp`测试代码中的正确性测试模块，得到以下内容：

```
*
 * @file myVector.h
 * @brief A Memory-contiguous, variable-length array
 * @details
```

```
 * 不知道该写什么，反正就是 vector，正常用就行了
 * @author github.com/GYPpro
 * @version 0.2.0
```

```
  pb 1
  pb 2
  pb 3
  pb 4
  iterF
  pb 9
  iterB
  clr
  iterF
  pb 1
  iterF

  pb 1
  pb 2
  pb 3
  pb 4
  iterF
Iter with index:
1 2 3 4
Iter with begin end
1 2 3 4
Iter with AUTO&&
1 2 3 4
  pb 9
  iterB
Iter with index:
9 4 3 2 1
Iter with begin end
9 4 3 2
  clr
  iterF
Iter with index:

Iter with begin end

Iter with AUTO&&
```

```
  pb 1
  iterF
Iter with index:
1
Iter with begin end
1
Iter with AUTO&&
1
```

可以看出，代码运行结果与预期相符，可以认为代码正确性无误。

运行`_PRIV_TEST.cpp`中的内存测试模块，在保持 CPU 高占用率运行一段时间后内存变化符合预期，可以认为代码内存安全性良好。