

## **std::Vector Выделение памяти в векторе**

- 1) С помощью функции capacity выяснить как изменяется занимаемый вектором объем
- 2) Как изменяется емкость вектора, если задать начальный объем.
- 3) Как изменяется объем, если у ОСки не хватает ресурсов

### **Ход работы**

#### **Как изменяется объем вектора при добавлении новых элементов**

Был проведен следующий эксперимент:

```
std::vector<int> test1;
std::cout << "Changing of capacity" << std::endl;
for (int i = 0; i < 10; i++)
{
    test1.push_back(i);
    std::cout << "Capacity of vector: " << test1.capacity() << " Size of vector: "
```

Рис. 1. Код первого теста

Для вектора размером меньше 1066 верно следующее утверждение, что если не задавать его изначальный размер, то если не хватает памяти он будет увеличиваться в 1.5 раза.

#### **Если объем был зарезервирован**

Был проведен следующий эксперимент:

```
std::vector<int> test3;
test3.reserve(100);
for (int i = 0; i < 2000; i++)
{
    test3.push_back(1);
    std::cout << "Capacity of vector: " << test3.capacity() << " Size of vector: "
```

Рис. 2. Код второго теста

До заполнения изначально выделенного объема количество памяти выделенной вектору не изменяется, однако после достижения этого предела элементы добавляются стандартным образом, то есть выделяемая память увеличивается в 1.5 раза.

#### **Максимальный объем**

Был проведен следующий тест: интовый вектор заполнялся единицами:

Код эксперимента:

При достижении максимально возможной памяти код зависает, но ошибка не выводится. Параллельно с расчетом был открыт диспетчер задач и в момент когда код завис было видно, что оперативная память полностью заполнена.

```
Capacity: 1551591828 Size: 1550700000  
Capacity: 1551591828 Size: 1550800000  
Capacity: 1551591828 Size: 1550900000  
Capacity: 1551591828 Size: 1551000000  
Capacity: 1551591828 Size: 1551100000  
Capacity: 1551591828 Size: 1551200000  
Capacity: 1551591828 Size: 1551300000  
Capacity: 1551591828 Size: 1551400000  
Capacity: 1551591828 Size: 1551500000
```

Рис. 3. Последний вывод теста

```
std::vector<int> test2;  
  
for (long long int i = 0; i < 1000000000000000; i++)  
{  
    if (i % 100000 == 0) {  
        std::cout << "Capacity: " << test2.capacity() << " Size: " << test2.size()  
    }  
    test2.push_back(1);  
}
```

Рис. 4. Тест с переполнением памяти