

问题

请说说new/delete与malloc/free的区别？

区别

- new和delete是C++的**关键字**，是一种操作符，**可以被重载**
- malloc和free是**C语言的库函数**，并且**不能重载**
- malloc使用时需要自己显示地计算内存大小，而new使用时由编译器自动计算

```
1 | int *q = (int *)malloc(sizeof(int) * 2);    //显示计算内存大小
2 | int *p = new int[2];                      //编译器会自动计算
```

- malloc分配成功后返回的是**void*指针**，需要强制类型转换成需要的类型；而new**直接就返回了对应类型的指针**
- new和delete使用时会分别调用构造函数和析构函数，而malloc和free只能申请和释放内存空间，不会调用构造函数和析构函数

代码示例

注意：**delete和free被调用后，内存不会立即回收，指针也不会指向空**，delete或free仅仅是告诉操作系统，这一块内存被释放了，可以用作其他用途。但是由于没有重新对这块内存进行写操作，所以内存中的变量数值并没有发生变化，这时候就会出现野指针的情况。因此，释放完内存后，应该把指针指向NULL。

```
1 | int main() {
2 |     int *q = (int *)malloc(sizeof(int) * 2);
3 |     int *p = new int[2];
4 |     cout << "p = " << p << "   q = " << q << endl;
5 |     free(p);    //指针还没指向空
6 |     delete q;   //同上
7 |     cout << "p = " << p << "   q = " << q << endl;
8 |     return 0;
9 | }
10 | //上面程序运行的结果，可见第二次打印的时候p和q指针还没指向空
11 | //这里不知为何编译器第二次打印的q和第一次不一样
12 | //  p = 000000000039B9B0   q = 000000000039B960
13 | //  p = 000000000039B9B0   q = 0000000000008123
```

参考资料

[new 和 malloc free 和 delete 的区别](#)

[new/delete与malloc/free的区别与联系详解！](#)