

extra5. Выравнивания. Оператор alignof и спецификатор alignas, их действие и пример использования. Основные правила выравнивания стандартных типов. Особенности выравнивания при выделении памяти стандартной функцией operator new. Функция std::aligned_alloc, ее предназначение.

Alignment - минимальная степень двойки, такая что только с адресов кратных этой степени двойки можно класть объект этого типа - выравнивание.

Выравнивание структуры - выравнивание наибольшего поля

```
1 struct S {
2     int x;
3     double y;
4     int z;
5 }
6
7 int main() {
8     std::cout << sizeof(S);
9 }
```

Будет 24 байта: 4 на инт, 4 на выравнивание дабла, 8 на дабл, 4 на инт и 4 на выравнивание структуры (чтобы заканчивалась делящимся на 8).

`std::cout << alignof(S);` - как раз минимальная степень двойки, чтобы положить. Здесь 8 (потому что дабл).

Можно написать:

```
1 struct alignas(8) S {
2     int x;
3     int y;
4 };
```

Проблема может быть такая: когда в аллокаторе вызывается `new(count * sizeof(T))` может быть не учтено выравнивание. Правда `malloc` умная функция и выделит с учетом выравнивания для стандартных типов (обычно это 8 байт, но мб и больше) - `std::max_align_t`.

Но проблема может быть если мы собственноручно сделали `alignas` больше чем максимальный в стандарте, и тогда будет UB.

Есть `aligned_alloc` - в который выравнивание: `aligned_alloc(alignof(S), n)` - в операторе `new`.

Это бывает нужно если есть мощный комп и за одну процессорную инструкцию можно сделать действие - за раз 32 байтное число, или сделать операцию сразу с 4 8-байтными числами. Или наоборот для `char` можно ходить выравнивание 1.