

Семинар 12: Архитектура

Igor Zhirkov

16 ноября 2021 г.

Содержание

1 Изначальный код 1

2 Задание 2

На этом семинаре вы можете делать одно из двух заданий.

Первый вариант это развитие принтера, который вы увидели на прошлом семинаре. Добавьте туда:

- вычислитель, который умеет вычислять формулы. Сделайте его расширяемым, так же, как сделан принтер. Подсказка: в отличие от принтера, который не имеет состояния, для вычисления потребуется накапливать состояние, поэтому функции будут иметь больше аргументов, нежели принтеры для узлов.
- парсер, который сможет распознавать верные формулы из сложения, умножения, деления и вычитания. Не забудьте про скобки!

Второй вариант это реализация несложной структуры данных как абстрагированного модуля с непрозрачным типом. Ему посвящена эта заметка.

1 Изначальный код

У вас есть код внутри одной функции `main`, который реализует расширяемый массив (вектор). Изучите его.

```
#include <inttypes.h>
#include <malloc.h>
#include <stdio.h>
```

```
int main() {
    int64_t *array = malloc(sizeof(int64_t) * 5);
```

```

// вместимость -- сколько памяти выделено
size_t capacity = 5;
// количество -- сколько памяти по факту используется из выделенной.
size_t count = 0;
// заполните массив квадратами чисел от 0 до 100
// если не хватает места, расширяем в два раза
for (size_t i = 0; i <= 100; i++) {
    if (count == capacity) {
        array = realloc(array, sizeof(int64_t) * capacity * 2);
        capacity = capacity * 2;
    }
    array[count++] = i * i;
}

for (size_t i = 0; i < 100; i++) {
    printf("%" PRId64 " ", array[i]);
}
return 0;
}

```

Расширяемый массив в отличие от обычного имеет нефиксированный размер, в конец такого массива можно добавлять элементы. Как это работает:

- Мы выделяем память с запасом.
- Мы храним два дополнительных числа: количество выделенных слотов под элементы и количество заполненных слотов в массиве.
- Пока у нас хватает выделенных слотов, мы просто дописываем элементы в массив, увеличивая количество занятых слотов.
- Если слотов перестало хватать, то увеличиваем количество слотов в 2 раза. Для этого используем `realloc` чтобы скопировать выделенную память в расширенный участок.

Вопрос Прочитайте раздел `man malloc` про функцию `realloc`.

2 Задание

Ваша задача — выделить из этого кода как минимум модуль с реализацией вектора, снабжённый заголовочным файлом.

- Вектор должен быть реализован как непрозрачная структура данных.

- Доступ к его элементам должен быть контролируем и осуществляться через `getter` и `setter`.
- Постарайтесь максимально переиспользовать код и ничего не дублировать.
- Вывод вектора реализуйте как отдельную функцию, которая принимает `FILE*`, в который нужно вывести его содержимое. Эту функцию можно также разбить на функцию `foreach` и принтер одного элемента.
- Стремитесь сделать настолько маленькие функции, насколько возможно.

В результате должна получиться программа из нескольких файлов, которая делает то же самое, но в которой `main` содержит только заполнение вектора числами и вызов функции, печатающей его в `stdout`.

Прочитайте заметку [Правила хорошего стиля] — ваш код должен им соответствовать.