

Работа 3. Изучение особенностей размещения объектов в динамической памяти

Теоретический материал

Для выполнения работы вам будет необходимо использовать материал по следующим темам:

1. Массивы и указатели (материал из предыдущей работы)

- Общая информация о работе с массивами: <https://metanit.com/c/tutorial/2.13.php>
- Общий материал по указателям: <https://metanit.com/c/tutorial/5.1.php>
- Связь указателей и массивов: <https://metanit.com/c/tutorial/5.5.php>

2. Динамическое выделение памяти

- Функции для работы с динамической памятью(функции `malloc`, `realloc`, `free`): <https://metanit.com/c/tutorial/5.8.php>

3. Работа с объектами в памяти

- Работа с памятью (функция `memcpy`): <https://metanit.com/c/tutorial/8.3.php>

ЗАДАНИЕ:

1. Создание массивов

Создайте два массива А и В, каждый объемом от 100 МБ до 1 ГБ, используя функцию `malloc`. Размер каждого массива вводится пользователем («150М», «233М»)

2. Увеличение размера

Увеличьте размеры массивов в 2 раза, используя функцию `realloc`.

Измерьте время, затраченное на выполнение этой операции, и выведите его на экран. Сравните значения указателей (адресов начала массивов до и после увеличения).

3. Копирование фрагмента массива

Пользователь вводит **начальную позицию** и **длину** субмассива для копирования (не забываем про проверку на превышение размеров и начало!).

Увеличить массив A на соответствующую **длину**.

Переместить (скопировать) субмассив из **A** начиная с начальной позиции в конец массива **A**.

Скопировать часть массива **B** в **A** (длина соответствует введённой пользователем)

Измерять время выполнения операций.

4. Уменьшение массивов

Уменьшить размеры массивов **A** и **B** в **2** раза используя **realloc**. Замерять время выполнения и сравнить значения указателей (адреса начал массивов до и после, начало A до с начало A после и т.д.)

5. Очистка и доступ

Очистить память используя **free**. После очистки попытаться прочесть любой элемент из массива **A**. Объяснить результат.

6. **Продвинутый уровень

Создать 2 новых массива произвольной длины.

Увеличивайте размер массива и заполняйте случайными данными следующими способами:

А) «Малые» порции – порядка 1КБ за раз

Б) «Большие» порции – порядка 1МБ за раз

Точный размер порции задаете самостоятельно.

Производите до 10-ти операций, измеряете время выполнения каждой операции, сводите в таблицу и получаете среднее значение.

При замере времени выполнения (производительности) не включайте функции, в которых производится пользовательский ввод! Так как он вносит элемент случайной задержки!

Содержание отчёта:

Код

Измерения времени и адреса указателей для каждого пункта.

Ваши выводы по результатам пункта **5** и **6**** (поменьше воды!).

Подсказки:

Для замера времени выполнения используйте функции из стандартной библиотеки времени, например:

```
#include <time.h>
```

```
clock_t start, end;
```

```
double elapsed_time;
```

```
// Засаекаем время до вызова функции
```

```
start = clock();
```

```
/* Измеряемая операция */  
// Засекаем время после завершения функции  
end = clock();  
  
// Вычисляем разницу времени в миллисекундах  
elapsed_time = (double)(end - start) / CLOCKS_PER_SEC * 1000;
```

В случае, если исследуемый блок кода выполняется слишком быстро, и разница между end и start равна или близка к нулю, попробуйте модифицировать алгоритм так, чтобы искомый фрагмент кода выполнялся много раз подряд с одними и теми же параметрами, и измерьте суммарное время. При этом обратите внимание, чтобы в настройках компиляции не использовалась оптимизация (обычно для этого надо указать ключ -O0 в командной строке gcc, для других компиляторов возможны другие варианты, уточняйте в руководстве по среде разработки)

Параметры и условия для каждого варианта

1. **Вариант 1**
 - Начальные размеры массивов: 1 МБ, 100 МБ
 - Полиси увеличения: 1 МБ за раз, удвоение размера
2. **Вариант 2**
 - Начальные размеры массивов: 10 КБ, 500 МБ
 - Полиси увеличения: 512 байт за раз, 1.5x от текущего размера
3. **Вариант 3**
 - Начальные размеры массивов: 5 МБ, 50 МБ
 - Полиси увеличения: 256 байт за раз, 5 МБ за раз
4. **Вариант 4**
 - Начальные размеры массивов: 10 МБ, 300 МБ
 - Полиси увеличения: 500 КБ за раз, 2.5x от текущего размера
5. **Вариант 5**
 - Начальные размеры массивов: 100 КБ, 400 МБ
 - Полиси увеличения: 2 КБ за раз, удвоение размера
6. **Вариант 6**
 - Начальные размеры массивов: 100 КБ, 10 МБ
 - Полиси увеличения: 1 КБ за раз, 100 КБ за раз
7. **Вариант 7**
 - Начальные размеры массивов: 500 КБ, 50 МБ
 - Полиси увеличения: 128 байт за раз, 3x от текущего размера
8. **Вариант 8**
 - Начальные размеры массивов: 25 МБ, 250 МБ
 - Полиси увеличения: 512 байт за раз, 4 МБ за раз
9. **Вариант 9**
 - Начальные размеры массивов: 2 МБ, 100 МБ

- Полиси увеличения: 2 КБ за раз, 20 МБ за раз

10. Вариант 10

- Начальные размеры массивов: 1 КБ, 10 МБ
- Полиси увеличения: 128 байт за раз, удвоение размера