



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4 **«Работа со стеком»**

Студент Городский Юрий Николаевич

Группа ИУ7 – 32Б

Оглавление

Условие задачи.....	3
Техническое задание.....	3
Функции программы.....	3
Аварийные ситуации:.....	4
Обращение к программе.....	4
Структуры данных.....	4
Тесты.....	5
Замерный эксперимент.....	8
Контрольные вопросы.....	10
Вывод.....	11

Условие задачи

Цель работы: реализовать операции работы со стеком, который представлен в виде статического массива и в виде односвязного линейного списка; оценить преимущества и недостатки каждой реализации; получить представление о механизмах выделения и освобождения памяти при работе со стеком.

Элементами стека являются адреса памяти. При реализации массивами - их вводить, при реализации списком – брать адрес выделенной памяти под элемент.

Техническое задание

Входные данные:

1. **Номер команды:** целое число в диапазоне от 0 до 11
2. **Стек-список:** вводится количество элементов, стек заполняется автоматически адресами выделенных элементов
3. **Стек-массив:** Вводится количество элементов, вводится значение каждого элемента

Выходные данные:

1. Состояние стека — списка: таблица с элементами стека, таблица свободных областей
2. Состояние стека — статического массива
3. Файлы с результатами временных замеров.

Функции программы

1. Ввести стек - список
2. Добавить элемент в стек - список
3. Удалить элемент из стека - списка
4. Вывести состояние стека - списка
5. Очистить стек - список
6. Ввести стек - статический массив
7. Добавить элемент в стек - статический массив
8. Удалить элемент из стека - статического массива

9. Вывести состояние стека - статического массива
10. Очистить стек - статический массив
11. Замерный эксперимент
12. Выйти

Аварийные ситуации:

1. Некорректный ввод команды (введено не число или число не находится в диапазоне от 0 до 11): сообщение «Неверная команда».
2. Введено не число при вводе размеров стека-списка или стека — статического массива: сообщение «Ошибка ввода»
3. Введено неверное количество элементов в стеке (число < 0 или число $>$ указанного в интерфейсе максимума в 10000 элементов): сообщение «Ошибка диапазона данных»
4. Не удалось открыть файл при записи замерного эксперимента: сообщение «Ошибка работы с файлом».
5. Переполнение стека: сообщение «Ошибка: Стек заполнен»

Обращение к программе

Запуск через терминал (./app.exe).

Структуры данных

```
// Элемент стека на односвязном списке
typedef struct list_stack_node_t
{
    size_t node_num; // Индекс в стеке
    void *value; // Значение
    struct list_stack_node_t *previous; // Указатель на предыдущий элемент
} list_stack_node_t;

// Стек на односвязном списке
typedef struct
{
    size_t max_nodes_num; // Максимальное допустимое кол-во элементов в стеке
    list_stack_node_t *top; // Вершина стека
} list_stack_t;
```

```
static list_stack_t free_stack = {0, NULL}; // Стек свободных адресов
```

```
// Стек - статический массив
typedef struct
{
    void *arr[STATIC_STACK_SIZE]; // Массив значений
    void **start; // Начало массива
    void **end;    // Конец массива
    void **cur;    // Текущий элемент массива
} static_stack_t;
```

Тесты

Таблица 1: Негативные тесты

Описание	Ввод	Вывод
Удаление из пустого стека-списка	3	Ошибка: Пустой стек
Удаление из пустого стека-массива	8	Ошибка: Пустой стек
Добавление в заполненный стек-список	1 10000 2 ab2	Ошибка: Стек заполнен
Количество элементов при вводе стека-списка < 0	1 -1	Ошибка диапазона данных
Количество элементов при вводе стека-списка > 10000	1 10001	Ошибка диапазона данных
Количество элементов при вводе стека-массива < 0	6 -1	Ошибка диапазона данных
Количество элементов при вводе стека-массива > 10000	6 10001	Ошибка диапазона данных
При вводе стека-массива введен некорректный адрес	6 3 nn	Ошибка ввода
При добавлении элемента в стек-массив введен некорректный адрес	7 nn	Ошибка ввода

При добавлении элемента в стек-список введен некорректный адрес	2 nn	Ошибка ввода
---	---------	--------------

Таблица 2: Пример работы со стеком-списком

Описание	Вход	Выход
Ввод стека-списка	1	Операция завершена
	4	...
	4	Стек на списке
		Адрес элемента Значение эл-та n Предыдущий
		0x555b7411db20 0x555b7411db20 3 0x555b7411db00
		0x555b7411db00 0x555b7411db00 2 0x555b7411dae0
		0x555b7411dae0 0x555b7411dae0 1 0x555b7411dac0
		0x555b7411dac0 0x555b7411dac0 0 (nil)
		Стек свободных областей
		Пустой стек
	Операция завершена	
Добавление элемента стека-списка	2	Операция завершена
	3	...
	4	Стек на списке
		Адрес элемента Значение эл-та n Предыдущий
		0x555b7411db40 0x3 4 0x555b7411db20
		0x555b7411db20 0x555b7411db20 3 0x555b7411db00
		0x555b7411db00 0x555b7411db00 2 0x555b7411dae0
		0x555b7411dae0 0x555b7411dae0 1 0x555b7411dac0
		0x555b7411dac0 0x555b7411dac0 0 (nil)
		Стек свободных областей
	Пустой стек	
	Операция завершена	
Удаление элемента стека-списка	3	Удаленный элемент: 0x555b7411db40
	4	Операция завершена
		...
		Стек на списке
		Адрес элемента Значение эл-та n Предыдущий
		0x555b7411db20 0x555b7411db20 3 0x555b7411db00
		0x555b7411db00 0x555b7411db00 2 0x555b7411dae0
		0x555b7411dae0 0x555b7411dae0 1 0x555b7411dac0
		0x555b7411dac0 0x555b7411dac0 0 (nil)
		Стек свободных областей
	Адрес элемента Значение эл-та n Предыдущий	
	0x555b7411db40 (nil) 0 (nil)	
	Операция завершена	
Очистка стека-списка	5	Операция завершена
	4	...
		Стек на списке
		Пустой стек
		Стек свободных областей
		Адрес элемента Значение эл-та n Предыдущий
		0x555b7411dac0 (nil) 4 0x555b7411dae0
		0x555b7411dae0 (nil) 3 0x555b7411db00
		0x555b7411db00 (nil) 2 0x555b7411db20
		0x555b7411db20 (nil) 1 0x555b7411db40

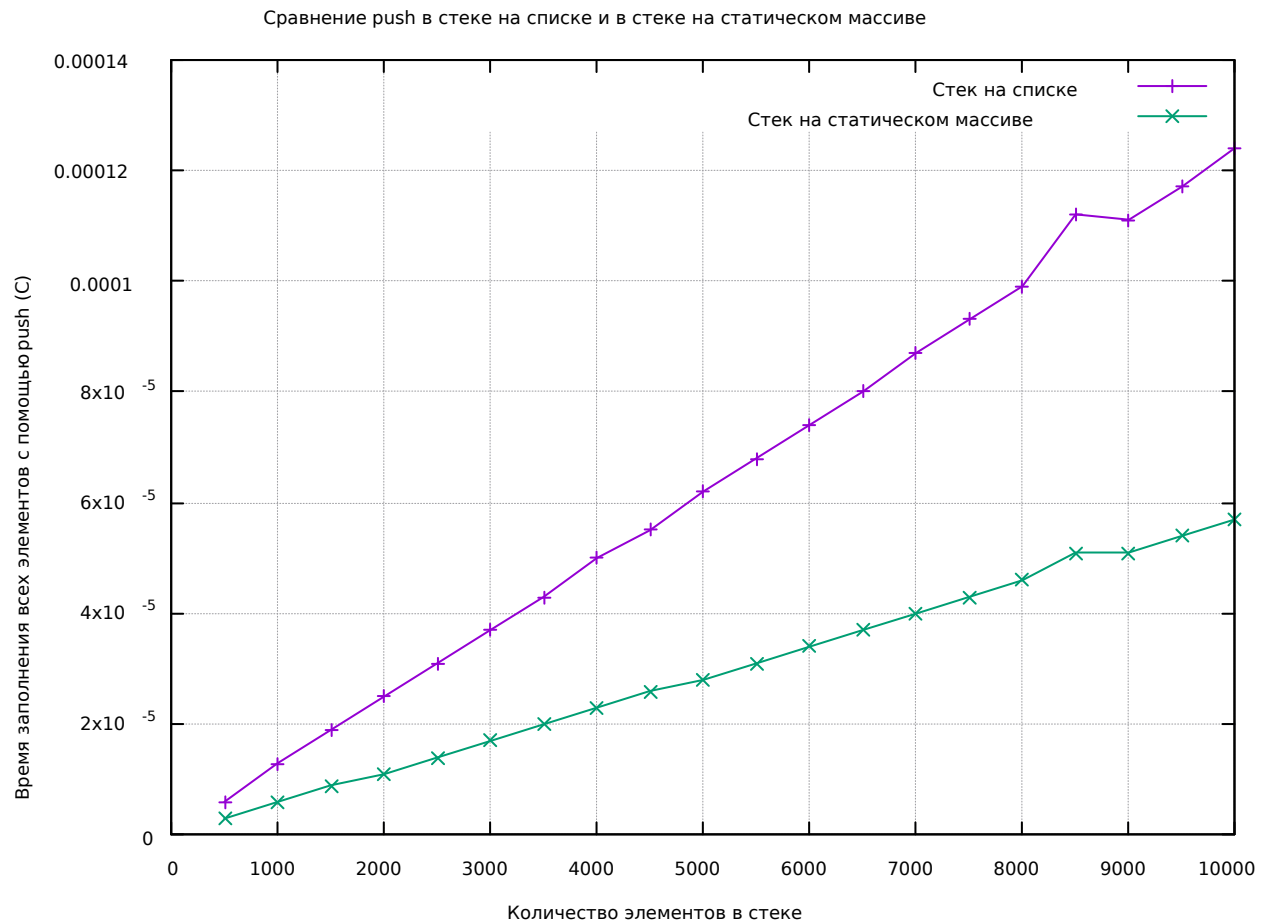
		0x555b7411db40 Операция завершена	(nil) 0 (nil)
--	--	--	-------------------

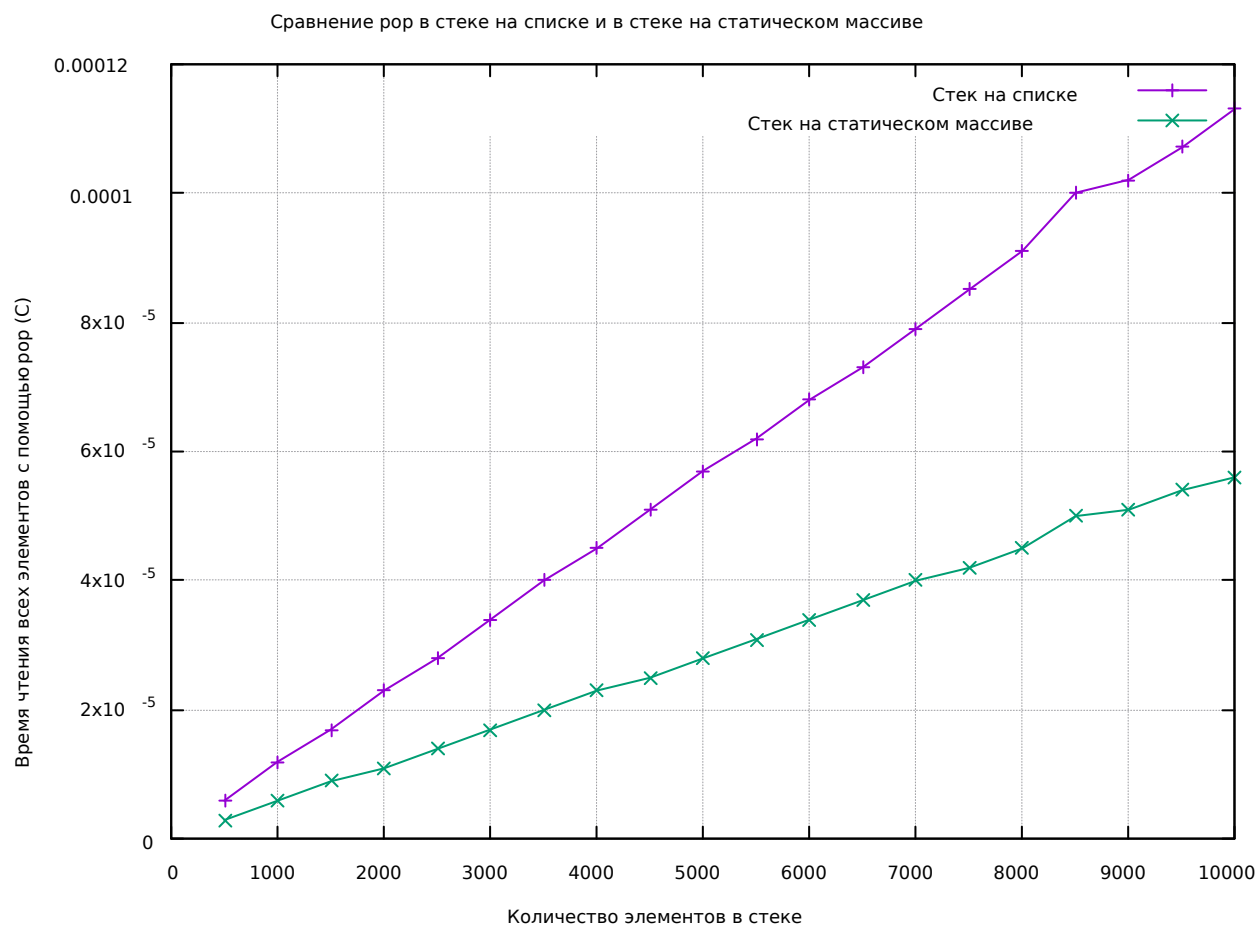
Таблица 3: Пример работы со стеком-массивом

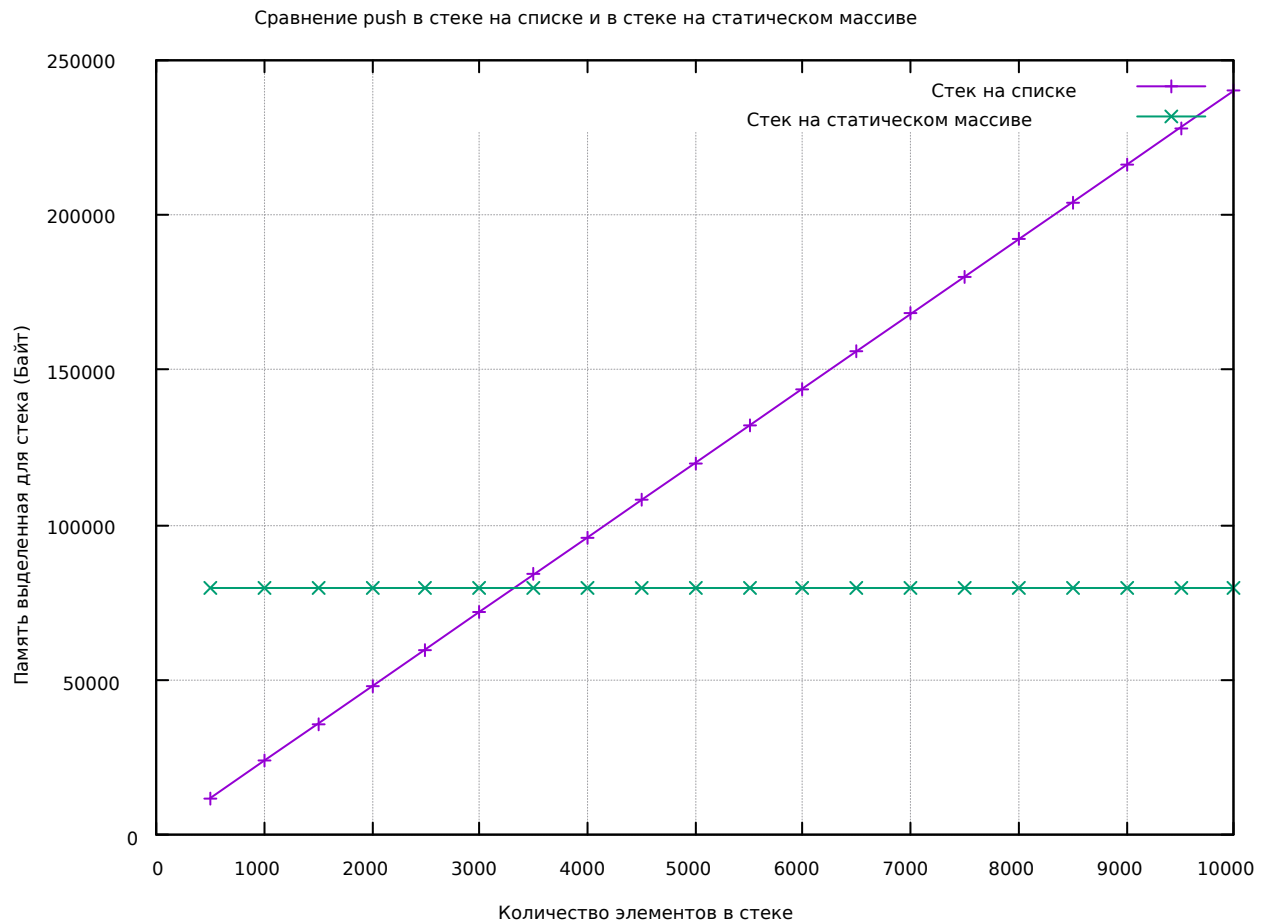
Описание	Вход	Выход
Ввод стека- массива	6 4 1 2 3 4 9	Операция завершена ... Стек - статический массив Адрес элемента значение элемента 0x7fff0a9610f8 0x4 0x7fff0a9610f0 0x3 0x7fff0a9610e8 0x2 0x7fff0a9610e0 0x1 Операция завершена
Добавлени е элемента стека- массива	7 5 9	Операция завершена ... Стек - статический массив Адрес элемента значение элемента 0x7fff0a961100 0x5 0x7fff0a9610f8 0x4 0x7fff0a9610f0 0x3 0x7fff0a9610e8 0x2 0x7fff0a9610e0 0x1 Операция завершена
Удаление элемента стека- массива	8 9	Операция завершена ... Стек - статический массив Адрес элемента значение элемента 0x7fff0a9610f8 0x4 0x7fff0a9610f0 0x3 0x7fff0a9610e8 0x2 0x7fff0a9610e0 0x1 Операция завершена
Очистка стека- массива	10 9	Операция завершена ... Пустой стек Операция завершена

Замерный эксперимент

Замеры проводились для добавления и удаления всех элементов. В каждом замере проводилось 1000 измерений и бралось среднее значение. Количество элементов: от 500 до 10000 с шагом в 500 элементов







Контрольные вопросы

1. Стек – это структура данных, которая работает по принципу "последний пришёл – первый вышел" (LIFO).
2. При реализации стека на статическом массиве его размер зависит от установленного максимального количества элементов. При реализации стека на списке его размер динамичен – для каждого элемента выделяется новая область памяти.
3. При удалении элемента со стека: В случае массива: его длина уменьшается на 1, но сам элемент остаётся в ячейке памяти – он будет перезаписан следующим добавленным элементом. В случае списка: адрес удалённого элемента помещается в массив освобождённых адресов памяти, а узел списка стирается из памяти.
4. Создаётся копия стека, в которую помещаются все элементы. Элементы стека поочерёдно извлекаются из стека и выводятся на экран. Затем элементы возвращаются в стек в исходном порядке.
5. Разница между подходами минимальна на небольших данных. При большем количестве данных, реализация стека на массиве выигрывает по всем параметрам, за исключением удаления элементов (и только при больших данных, более 3000). Если известно, сколько максимум данных должно храниться в стеке – следует использовать массив. Если нужен динамический стек – следует использовать связный список.

Вывод

Реализация стека на массиве выигрывает в большинстве случаев по скорости и памяти. При реализации элементов стека из задания по памяти выигрыш начинается с 3200 элементов, а по скорости стек на статическом массиве всегда быстрее стека на списке. Поэтому, если максимальное количество записей известно заранее, стоит использовать именно эту реализацию. Для динамического стека подойдёт односвязный список, но за это придётся пожертвовать памятью, так как память выделяется под каждый узел.