



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 Тузов Даниил Александрович

Группа ИУ7 – 32Б

Преподаватель Барышникова Марина Юрьевна
Силантьева Александра Васильевна

Оглавление

<u>ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ.....</u>	<u>3</u>
<u>ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....</u>	<u>3</u>
<u>ОПИСАНИЕ СТРУКТУР ДАННЫХ.....</u>	<u>4</u>
<u>ОПИСАНИЕ АЛГОРИТМА.....</u>	<u>4</u>
<u>ОПИСАНИЕ ФУНКЦИЙ.....</u>	<u>5</u>
<u>НАБОР ТЕСТОВ.....</u>	<u>6</u>
<u>ЗАМЕРЫ ВРЕМЕНИ РАБОТЫ.....</u>	<u>7</u>
<u>ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ.....</u>	<u>7</u>
<u>ВЫВОД.....</u>	<u>8</u>

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Создать программу работы со стеком, выполняющую операции добавление, удаления элементов и вывод текущего состояния стека. Реализовать стек: а) статическим массивом (дополнительно можно реализовать динамическим массивом); б) списком. Все стандартные операции со стеком должны быть оформлены подпрограммами. При реализации стека списком в вывод текущего состояния стека добавить просмотр адресов элементов стека и создать СВОЙ список или массив свободных областей (адресов освобождаемых элементов) с выводом его на экран.

Элементами стека являются слова. Распечатайте слова в обратном порядке, каждое слово в перевернутом виде.

ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Входные данные:

Пользователю предлагается выбрать действие: добавить в стеки, удалить из стеков, вывести стек-массив, вывести стек-список, вывести результат по стек-массиву, вывести результат по стек-списку, закончить работу. При выборе добавления пользователю предлагается ввести слово длиной до 40 символов

Выходные данные:

В консоль выводится распечатанные в обратном порядке перевернутые слова и время выполнения каждого действия

Описание программы:

Программа имитирует работу стека двумя способами: с помощью массива, с помощью списка

Способ обращения к программе:

Взаимодействие через консоль.

Аварийные ситуации:

1. Любой некорректный ввод данных в программе
Код ошибки — 1
2. Ошибка длины введенного слова
Код ошибки — 2

3. Ошибка при удалении из пустого стека
Код ошибки — 3
4. Ошибка переполнения, если в стеке уже 100 элементов
Код ошибки — 4
5. Ошибка работы с памятью
Код ошибки — 5

ОПИСАНИЕ СТРУКТУР ДАННЫХ

Стек в виде массива слов

```
char arr[N][WORD_LEN + 1]
```

Стек в виде списка слов

word – слово, которое хранится в узле списка

next – следующий узел списка

```
typedef struct stack_list  
{  
    char word[WORD_LEN + 1];  
    struct stack_list *next;  
} stack_list_t;
```

ОПИСАНИЕ АЛГОРИТМА

1. Пользователь выбирает действие
2. Согласно выбранному действию вызывается соответствующая функция
3. Выполняется логика этой функции
4. Если во время выполнения функции возникла ошибка, то выводится сообщение, поясняющее эту ошибку
5. Если ошибки не возникло, возвращаемся к 1 пункту

ОПИСАНИЕ ФУНКЦИЙ

`void description();` – функция выводит описание работы программы

`int read_str(char str[WORD_LEN + 1]);` – функция, которая считывает слово из консоли и записывает его в str. Возвращает код ошибки

`void print_str(char *str);` – функция, которая печатает строку str в консоль

`void reverse_print_str(char *str);` – функция, которая печатает перевернутую строку str в консоль

`int pop_stack_arr(char stack[][WORD_LEN + 1], size_t *size);` – функция удаляет из стек-массива stack размера size верхний элемент. Возвращает код ошибки

`int push_stack_arr(char stack[][WORD_LEN + 1], size_t *size, char *str);` – функция добавляет в стек-массив stack размера size элемент str. Возвращает код ошибки

`void print_stack_arr(char stack[][WORD_LEN + 1], size_t size);` – функция печатает стек-массив stack размера size

`void print_res_stack_array(char stack[][WORD_LEN + 1], size_t size);` – функция печатает результат, записанный в стек-массиве stack размера size

`void free_stack_list(stack_list_t *head);` – функция освобождает память из под стек-списка head

`int pop_stack_list(stack_list_t **head, size_t *size);` – функция удаляет верхний элемент из стек-списка head размера size. Возвращает код ошибки

`int push_stack_list(stack_list_t **head, size_t *size, char *str);` – функция добавляет в стек-список head размера size. Возвращает код ошибки

`void print_stack_list(stack_list_t *head);` – функция печатает стек-список head

`void print_res_stack_list(stack_list_t *head);` – функция печатает результат по стек-списку head

НАБОР ТЕСТОВ

№	Описание теста	Ввод пользователя	Ответ программы
1	Выбор действия	1	Добавить элемент в стеки
2	Выбор действия	2	Удалить элемент из стека
3	Выбор действия	3	Вывести стек-массив
4	Выбор действия	4	Вывести стек-лист
5	Выбор действия	5	Вывести результат по стек-массиву
6	Выбор действия	6	Вывести результат по стек-листу
7	Выбор действия	0	Выход из программы
8	Выбор действия	7	Ошибка ввода действия
9	Ввод слова	Пустой ввод	Ошибка длины слова
10	Ввод слова	Строка длиной более 40 символов	Ошибка длины слова
11	Ввод слова	Hello	Слово успешно введено
12	Добавление в стек	Размер стека 100	Ошибка переполнения стека
13	Добавление в стек	Размер стека 0-99	Успешно добавлено
14	Удаление из стека	Размер стека 0	Ошибка удаления из пустого стека
15	Удаление из стека	Размер стека больше 0	Успешно удалено
16	Вывод стека/результата	Размер стека 0	Предупреждение, что стек пуст
17	Вывод стека	Hello World	dlorW olleH

ЗАМЕРЫ ВРЕМЕНИ РАБОТЫ

В ходе эксперимента для получения результатов проводилось 100 замеров по времени. Усредненный результат представлен в таблице

	Удаление		Добавление	
	М	С	М	С
Время выполнения операции	0.0036	0.0111	0.0012	0.0022

*Значения времени в таблице представлены в миллисекундах

** М – массив, С – список

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое стек?

Стек – это последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с одной стороны – с его вершины.

2. Каким образом и сколько памяти выделяется под хранение стека при различной его реализации?

При хранении стека в виде массива выделяется память только под сами значения элементов стека. В случае моей задачи 40 байт на хранение слова. При хранении стека в виде списка выделяется память под хранение значения элемента и указатель на следующий элемент. В случае моей задачи 40 байт + 8 байт. Соответственно реализация стека методом списка проигрывает по памяти.

3. Каким образом освобождается память при удалении элемента стека при различной реализации стека?

При реализации через статический массив значение элемента затирается. При реализации через список освобождается динамически выделенная память

4. Что происходит с элементами стека при его просмотре?

При просмотре стека элементы поочередно достаются из стека, обрабатываются и помещаются обратно в стек в том же порядке

5. Каким образом эффективнее реализовывать стек? От чего это зависит?

По времени эффективнее использовать статический массив. По памяти тоже эффективнее использовать статический массив.

ВЫВОД

В рамках этой лабораторной работы я научился работать со стеком. Я реализовал работу со стеком двумя способами: с помощью статического массива и с помощью связанного списка. Сравнив время добавления в стек и удаления из него я пришел к выводу:

1. По памяти эффективнее использовать стек-массив. (Стек-лист занимает на 20% больше памяти в случае моей задачи)
2. По времени эффективнее использовать стек-массив. (Для стек-массива производится меньше операций)
 - Для добавления выигрыш в 2 раза
 - Для удаления порядка 3 раз