



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 Ковалец Кирилл

Группа ИУ7 – 33Б

2020 г.

Описание условия задачи

Создать программу работы со стеком, выполняющую операции добавление, удаления элементов и вывод текущего состояния стека. Реализовать стек:

а) 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.

Описание технического задания

Распечатать убывающие серии последовательности целых чисел в обратном порядке.

Входные данные:

Номер команды, отвечающий за определённое действие.

Команды:

1. Добавить элементы в стек;
2. Добавить случайные элементы в стек;
3. Удалить элементы из стека;
4. Вывести текущее состояние стека;
5. Распечатать убывающие серии последовательности целых чисел в обратном порядке;
0. Выйти из программы.

Выходные данные:

1. Результат выполнения определённой команды;
2. Результат сравнения обработки 2-х типов стеков по времени и памяти.

Обращение к программе:

Запускается через терминал командой `make run`

Сообщения при аварийных ситуациях:

1. Не удалось прочитать тип стека;
2. Не удалось прочитать номер команды;
3. Номер команды должен быть ≥ 0 и ≤ 5 ;
4. Не удалось прочитать кол-во добавляемых элементов;
5. Кол-во добавляемых элементов не может быть отрицательным;
6. Произошло переполнение стека;
7. Не удалось прочитать элемент стека;

8. Не удалось прочитать кол-во удаляемых элементов;
9. Кол-во удаляемых элементов не может быть отрицательным;
10. В стеке нет столько элементов;
11. Не удалось выделить память.

Описание структуры данных

stack_array_t - структура, содержащая информацию о стеке, реализованного с помощью массива.

```
typedef struct stack_array
{
    int *ptr;
    int size;
} stack_array_t;
```

Поля структуры:

- 1) int *ptr – указатель на текущий элемент стека;
- 2) int size – кол-во заполненных элементов стека.

elem_stack_list_t - структура, содержащая информацию о текущем элементе стека, реализованного с помощью односвязного списка.

```
typedef struct elem_stack_list
{
    int elem;
    int numb_elem;
    struct elem_stack_list *next;
} elem_stack_list_t;
```

Поля структуры:

- 1) int elem –текущий элемент стека;
- 2) int numb_elem – кол-во заполненных элементов стека;
- 3) struct elem_stack_list *next – указатель на следующий элемент стека.

Описание алгоритма

1. Выводится меню программы (каждой команде присвоен номер);
2. Пользователь вводит номер команды, который отвечает за определённое действие;
3. Ввод осуществляется до того момента, пока не будет введён 0, являющийся признаком выхода из программы.

Набор тестов

№	Название теста	Входные данные	Результат
1	Тип стека - число	k	Не удалось прочитать тип стека
2	Номер команды - число	k	Не удалось прочитать номер команды
3	Номер команды ≥ 0 и ≤ 5	8	Номер команды должен быть ≥ 0 и ≤ 5
4	Кол-во добавляемых элементов в стек - число	k	Не удалось прочитать кол-во добавляемых элементов
5	Кол-во добавляемых элементов в стек > 0	-1	Кол-во добавляемых элементов не может быть отрицательным
6	Максимальный размер стека - 10000	В стеке содержалось 5 элементов, попытались добавить ещё 9996	Произошло переполнение стека
7	Элемент стека – целое число	k	Не удалось прочитать элемент стека
8	Кол-во удаляемых элементов - число	k	Не удалось прочитать кол-во удаляемых элементов
9	Кол-во удаляемых элементов в стек > 0	-1	Кол-во удаляемых элементов не может быть отрицательным
10	Попытка удалить элементов больше, чем есть в стеке	В стеке содержалось 5 элементов, попытались удалить 6	В стеке нет столько элементов
11	Добавление элементов в стек	Добавление 5 элементов в незаполненный стек	Элементы успешно добавлены в стек
12	Добавление элементов в заполненный стек	Попытка добавить 1 элемент	Стек заполнен
13	Удаление элементов из стека	Удаление 5 элементов из заполненного стека	Элементы успешно удалены из стека
14	Удаление элементов из пустого стека	Попытка удалить 1 элемент	Стек пуст

15	Запрос на печать массива, содержащего адреса освобождённых элементов стека (список)	Номер команды – 1 (появляется при печати стека)	Печать массива
16	Распечатать убывающие серии последовательности целых чисел в обратном порядке	Элементы стека: -67 -7 -98 -39 41 -91 -16 -2 85 11	 -67 -7 -98 -39 41 -91 -16 -2 85
17	Вывести текущее состояние стека	Команда 4	Печать элементов стека в столбец (для списка выводится адрес каждого элемента стека)
18	Выход из программы	Команда 0	Выход из программы

Оценка эффективности

Сортировка каждой таблицы будет измеряться в тактах процессора (процессор со средней частотой 2.3GHz).

Сравнение эффективности

Добавление элементов (в тиках)

Размер	Массив	Список
10	1219	3484
100	4497	21724
500	12401	72099
1000	25060	135390

Удаление элементов (в тиках)

Размер	Массив	Список
10	119	2597
100	1038	18789
500	5060	70534
1000	8319	131583

Печать убывающих серий последовательности целых чисел в обратном порядке (в тиках)

Размер	Массив	Список
10	7359	5240
100	32620	28362
500	128051	118246
1000	232917	215386

Сравнение памяти

Размер	Массив	Список
10	40016	160
100	40016	1600
500	40016	8000
1000	40016	16000

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. Стек функционирует по принципу: LIFO - последним пришел — первым ушел,

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

Если хранить стек как список, то память выделяется в куче. Если хранить как массив — либо в куче, либо на стеке (зависит от того, динамически или статический массив используется). Для каждого элемента стека, который хранится как список, выделяется на 4 или 8 байт (если брать современные ПК) больше, чем для элемента стека, который хранится как массив.

Данные байты использованы для хранения указателя на следующий элемент списка. (из-за этого либо 4 либо 8 байт)

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

Если хранить стек как список, то верхний элемент удаляется при помощи операции освобождения памяти для него и смещением указателя, который указывает на начало стека.

Если хранить стек как массив, то смещается только указатель на начало стека.

4. Что происходит с элементами стека при его просмотре?

Элементы стека удаляются, так как каждый раз достается верхний элемент стека, чтобы посмотреть следующий.

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

Стек эффективнее реализовать с помощью массива, так как он выигрывает в количестве занимаемой памяти (если массив динамический) и во времени обработки стека (добавлении и удалении элементов).

Хранение с помощью списка может выигрывать, если только стек реализован с помощью статического массива, так как в данном случае размер памяти под список ограничен размером оперативной памяти (хранится в куче), а для статического массива — ограничена размером стека.

Вывод

Если стек реализован статическим массивом, то добавление в него новых элементов будет происходить быстрее (\sim в 3-5 раз), чем в стек, реализованный списком. Это связано с тем, что для хранения стека в виде списка требуется выделить память для указателей на следующие элементы списка. Удаление элементов из массива происходит в 13-20 раз быстрее, чем из списка, так как при этом затрачивается время на очищение памяти, динамически выделенной под элемент стека.

Для вывода убывающих серий последовательности целых чисел в обратном порядке стек-список справился быстрее (\sim на 8 – 15%) списка массива.

Вариант хранения стека в виде списка может выигрывать в том случае, если стек реализован статическим массивом (массив должен быть заполнен менее, чем на 25%). Так же, если не известен размер стека, то в таком случае стоит использовать списки (или динамические массивы).