МГТУ им. Н. Э. Баумана

ИУ 7 – 32

Отчет о лабораторной работе №4

## «Работа со стеком»

Вариант №8

Исаев Д.С.

# Цель работы

### Цель работы: реализовать операции работы со стеком, который представлен в виде массива (статического или динамического) и в виде односвязного списка, оценить преимущества и недостатки каждой реализации, получить представление о механизмах выделения и освобождения памяти при работе с динамическими структурами данных.

# Условие 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 свободных областей (адресов освобождаемых элементов) с выводом его на экран.

Проверить правильность расстановки скобок трех типов (круглых, квадратных и фигурных) в выражении.

# Входные данные

Текст, содержащий скобки.

**Âûõîäíûå äàííûå:**

Результат – верно ли расставлены скобки в тексте.

**Ïðèìåð:**

**Âõîäíûå äàííûå**

1. (())]
2. (()[{}edge])

**Âûõîäíûå äàííûå**

1. Неверно
2. Верно

**Описание**

**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: последним пришел – первым ушел, Last In – First Out (LIFO).

Программная реализация стека возможна на основе различных структур данных, например, с использованием статических или динамических одномерных массивов (векторов) и линейных списков.

При работе со стеком доступен только его верхний элемент, который адресуется специальным указателем стека PS (Pointer Stack). Причем классическая реализация стека предполагает, что просмотреть содержимое стека без извлечения (удаления) его элементов невозможно.

В современной реализации при просмотре верхнего элемента стека, он не удаляется.

**Реализация**

**Массив**

Принцип организации стека предполагает, что в текущий момент времени доступен элемент, индекс которого хранится в указателе стека PS. Граничные адреса стека – это параметры физической структуры. Физическая структура обычно дополняется дескриптором стека, в котором хранится имя стека, адрес нижней границы стека, адрес верхней границы стека (или максимальное количество элементов в нем), указатель стека (номер последнего записанного элемента) и описание (тип) элемента стека.

В своей программе я использовал динамический массив и выделял память небольшими порциями, по мере необходимости. Это более экономно по памяти чем статический массив, однако иногда требует долгой реструктуризации.

**Список**

До начала работы указатель стека показывает на нулевой, физически отсутствующий адрес (т. е. указатель - пустой). При включенииэлемента в стек сначала происходит выделение области памяти, адрес которой записывается в указатель стека, а затем по значению этого указателя в стек помещается информация.

При исключенииэлемента сначала по указателю стека считывается информация об исключаемом элементе, а затем указатель смещается к предыдущему элементу. После чего освобождается память, выделенная под элемент. Если указатель имеет значение нулевого адреса, то стек пуст.

При удалении элемента память освобождается, т.е. она может быть снова выделена для других переменных.

Память в данной программе при реализации стека списком выделяется при каждом новом добавлении элемента.

**Результаты**

В результате работы программы были получены следующие данные:

Количество сортировок: 100000

Время обработки:

* Списком – 5.17 с
* Массивом – 4.89 с

Затраты памяти:

* Списком – 6 байт на элемент
* Массивом – 2 байта н элемент

В результате наблюдения за освобождаемой памятью выснилось, что освободившаяся памтья почти всегда сразу же повторно используется при выделении памяти, что очень хорошо и не способствует фрагментации памяти.

Результаты показывают, что использование массива для поставленной задачи и эффективнее по быстродействию и экономнее по памяти.

Стек эффективнее реализовывать массивом, однако в редких случаях – когда ограничения на размер массива критичны, стоит использовать список.

**Вывод:**

Из результатов работы программы видно, что в данной технической задаче выгоднее использовать массив для реализации стека вместо списка.