

Лабораторная работа №2 по дисциплине «Типы и структуры данных»

Работа со стеком

Говязин Сергей Владимирович, ИУ7-31 (вариант 7).

Используется динамический массив и список.

Условие задачи:

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.

Исходные данные:

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.

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

Ввод данных:

Пользователь для работы с программой использует консоль. При открытии программы ему предлагается памятка по доступным командам.

Вывод:

ДЕМОНСТРАЦИОННАЯ ВЕРСИЯ СТЕКА v1.0

Используется 2 реализации стека: массивом и списком

Инструкция по работе со стеком

Введите help для получения этой инструкции

Введите switch для переключения между реализациями стеком и списком

Введите exit для выхода из программы

Введите for [count > 0] для выполнения следующей команды несколько раз

Введите rand для ввода случайного значения

При работе с реализацией МАССИВОМ:

Введите push [addr] для добавления какого-либо адреса

Введите pop для взятия адреса из стека

Введите get [i] для просмотра какого-либо элемента в стеке

Введите addr [i] для просмотра адреса какого-либо элемента в стеке

Введите size для просмотра количества элементов

При работе с реализацией СПИСКОМ:

Введите push для добавления элемента

Введите pop для взятия адреса из стека

Введите get [i] для просмотра какого-либо элемента в стеке

Введите addr [i] для просмотра адреса какого-либо элемента в стеке

Введите size для просмотра количества элементов

В зависимости от выбранного пункта меню, вывод программы будет отличаться.

Возможные ошибки:

- 1) Нехватка памяти
- 2) Индексация вне границ стека
- 3) Попытка удалить элемент из пустого стека
- 4) Задание неправильной длины цикла

Обработка ошибок — вывод сообщения и завершение программы

Структура данных для стека на основе списка

```
template <typename T>
struct Node
{
    T Data;
    Node* Next;
    Node() : Next(0){}
    Node(T data) : Next(data){}
};
```

Сравнение эффективности реализаций списков:

Время выполнения программы:

Кол-во эл-тов	Время-работы стека-массива (Добавление, удаление), нс	Время работы стека-списка (Добавление, удаление), нс
10	2556, 704	4654, 2795
1000	106582, 45626	182127, 36264
10000	617080, 298554	1137545, 662921
50000	4682457, 2392090	6066693, 1911353
100000	4681523, 2443470	6975629, 2426153
1000000	28219504, 9062206	52869132, 22414853

Реализация стека на массиве более эффективна по времени в 1.5-2.5 раза.

Для хранения каждого нового элемента при реализации через список требуется на 8 байт больше места чем при добавление его реализацию на массиве, но в то же время массив содержит буффер, память на который выделяется не часто и поэтому общая память списка может быть меньше, чем у массива

Вывод

Обе реализации имеют свои плюсы и минусы. Список использует дефрагментированную память, но в большинстве случаев памяти под него требуется больше, чем под реализацию массивом. Список работает немного медленнее, чем массив, но это проявляется при добавлении значительного количества элементов.

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?

Если стек основан на массиве, то для его хранения отводится непрерывный ограниченный блок памяти, характеризуемый заранее указанным количеством элементов. Этот блок памяти может расширяться и сужаться динамически (при динамической реализации)

В случае реализации стека на списке, память доступная для хранения элементов ограничена только объемом свободной оперативной памяти. Память выделяется каждый раз при добавлении нового элемента.

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

При реализации в виде списка, то при удалении очередного эл-та происходит освобождение области памяти, которую он занимал.

Если стек реализован в виде массива, то память освобождается после окончания работы со стеком: освобождается весь массив.

4. Что происходит с элементами стека при его просмотре?

Для того, чтобы просмотреть элемент, его необходимо извлечь из стека (операция Pop)

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

Эффективность определяется типом хранимых элементов и их количеством. Большие данные удобнее хранить в списке, а маленькие — в массиве.

