|  |  |
| --- | --- |
|  | **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

**Отчёт по лабораторной работе № 4 «Работа со стеком»**

Студент Шелия София Малхазовна

Группа ИУ7 – 35Б

*2020 г.*

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

1. Описание условия задачи.

**Общее задание**: 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.

*Специальное задание*

**Вариант 1**

Распечатайте убывающие серии последовательности целых чисел в обратном порядке.

1. Техническое задание

**Исходные данные и результат**

**Ввод:**

1. Пользователь вводит цифру (от 0 до 4), в зависимости от выбранного пункта меня.

* 1 - Добавление элемента в стек.

Пользователь вводит целое число – элемент, который он хочет добавить. Максимальное количество элементов в стеке – 100.

* 2 – Удаление элемента из стека.
* 3 – Печать текущего состояния стека
* 4 – Выполнение задания
* 0 – Завершение программы

Пользователь выбирает с каким стеком будет работать программа (1 – стек, с которым пользователь работал до этого, 2 – новый стек с рандом. значениями)

Если пользователь выбирает 2-ой вариант, то он также вводит целое число (не меньше 1 и не больше 1000) – количество элементов в стеке.

**Вывод: (для пунктов 1 - 4)**

1. Программа добавляет элемент в стек, представленный в виде статического массива и связного списка, а потом выводит их, а также выводит адреса элементов связного списка и свободной области.
2. Программа удаляет элемент из стека, представленного в виде статического массива и связного списка, а потом выводит их, а также выводит адреса элементов связного списка и свободной области.
3. Программа выводит текущее состояние связного списка и статического массива, а также адреса элементов связного списка и свободной области.
4. Программа выводит текущее состояние связного списка и статического массива, а также убывающие серии последовательности целых чисел в обратном порядке для обоих реализаций стека.

**Описание задачи, реализуемой программой**

1. Программа создает два стека, реализованных в виде статического массива и связного списка.
2. Программа производит операции добавления/удаления элемента одновременно для обеих реализаций стека.
3. Программа выводит текущее состояние стека в обеих реализациях, а также адреса элементов связного списка и свободной области.
4. Программа распечатывает убывающие серии последовательностей целых чисел в обратном порядке для обеих реализаций стека.

**Способ обращения к программе**

Запуск приложения возможен через терминал MSYS2, а именно.

1. gcc -std=c99 -Wall -Werror -c \*.c
2. gcc -o main.exe \*.o
3. ./main.exe

**Описание возможных аварийных ситуаций и ошибок пользователя**

**Ошибки пользователя при выборе пункта меню**

* Пустой ввод
* Вводятся посторонние символы (например, буквы)
* Вводится отрицательное число
* Вводится не целое число
* Вводится отсутствующий пункт меню

**Ошибки пользователя при вводе элемента для добавления**

* Пустой ввод
* Вводятся посторонние символы (например, буквы)
* Вводится не целое число

**Ошибки пользователя при вводе количества элементов для нового стека**

* Пустой ввод
* Вводятся посторонние символы (например, буквы)
* Вводится не целое число
* Вводится отрицательное число
* Вводится число больше того, которое было запрошено

**Аварийные ситуации.**

* Попытка добавления элемента в переполненный стек
* Попытка удаление элемента из пустого стека

**3. Описание внутренних структур данных**

Для хранения стека в виде статического списка я использовала следующую структуру:

typedef int Data;

typedef struct{

Data data[N]; - массив элементов стека

int n; - количество элементов стека

}Stack\_sa;

Для хранения стека в виде связного списка я использовала следующую структуру:

typedef int Data;

typedef struct **s\_stack**{

data; - значение элемента

s\_stack \*next; - указатель на следующий элемент

}Stack\_ll;

**4. Описание алгоритма**

Моя задача: «Распечатайте убывающие серии последовательности целых чисел в обратном порядке.»

Для ее решения мною были написаны две функции – одна для стека, реализованного в виде статического массива, а другая для стека в виде связного списка.

1. Стек, реализованный с помощью статического массива

* Программа создает дополнительный стек для хранения промежуточных результатов. В него помещается первый элемент исходного стека.
* Далее начинается цикл, который проходит по всем элементам исходного стека
* Если элемент, находящийся в начале промежуточного стека меньше текущего элемента исходного стека, то в промежуточный стек добавляется данный элемент исходного стека.
* Если это условие не выполняется и размер промежуточного стека больше 1, то все элементы промежуточного стека распечатываются, затем удаляются, и в промежуточный стек помещается текущий элемент исходного стека.
* Иначе (то есть убывающей последовательности не было) из промежуточного стека удаляется элемент и добавляется текущий элемент исходного стека.
* После завершения цикла проверяется длина промежуточного стека, если она больше 1, то элементы стека распечатываются
* В завершении функции из промежуточного стека удаляются все оставшиеся значения

1. Стек, реализованный с помощью связного списка

Алгоритм для связного списка полностью аналогичен алгоритму для статического массива, за исключением особенностей, связанных с данной реализацией стека.

**5. Тесты**

**Негативные тесты**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Результат** | **Условие проверки** |

|  |  |  |
| --- | --- | --- |
| **Для выбора пункта меню** | | |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: 4 | Некорректный ввод. | Пустой ввод |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: впав | Некорректный ввод. | Вводятся посторонние символы (например, буквы) |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: -5 | Некорректный ввод. | Вводится отрицательное число |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: 10 | Некорректный ввод. | Вводится отсутствующий пункт меню |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: 2.1 | Некорректный ввод. | Вводится не целое число |
| **Для ввода значения элемента** | | |
| Введите целое число - значение элемента для добавления: | Некорректный ввод. Введите целое число: | Пустой ввод |
| Введите целое число - значение элемента для добавления: 1ава | Некорректный ввод. Введите целое число: | Вводятся посторонние символы (например, буквы) |
| Введите целое число - значение элемента для добавления: 34.12 | Некорректный ввод. Введите целое число: | Вводится не целое число |

|  |  |  |
| --- | --- | --- |
|  | Стек пустой, удаление невозможно | Попытка удаления элемента из пустого стека |
|  | Стек переполнен, добавление невозможно (максимальное количество элементов - %d) | Попытка добавления элемента в переполненный стек |

**Позитивные тесты**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Результат** | **Условие проверки** |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стека  4 - Выполнение задания  Выбор: 1  Введите целое число - значение элемента для добавления: 1 | **До:**  Элементы стека (статический массив): -> 2 3  Элементы стека (связный список): -> 2 3  **После:**  Элементы стека (статический массив): -> 1 2 3  Элементы стека (связный список): ->1 2 3 | Добавление элемента в стек |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: 2 | **До:** Элементы стека (статический массив): -> 1 2 3  Элементы стека (связный список): -> 1 2 3  **После:**  Элементы стека (статический массив): -> 2 3  Элементы стека (связный список): -> 2 3 | Удаление элемента из стека |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: 4 | Элементы стека (статический массив): -> 1 2 3  Элементы стека (связный список): -> 1 2 3  Результат для массива: ->  Результат для связного списка: -> | Выполнение задания для полностью возрастающего стека |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: 4 | Элементы стека (статический массив): -> 3 2 1  Элементы стека (связный список): -> 3 2 1  Результат для массива: -> 1 2 3  Результат для связного списка: -> 1 2 3 | Выполнение задания для полностью убывающего стека |
| Выберите действие:  1 - Добавление элемента в стек  2 - Удаление элемента из стека  3 - Печать текущего состояния стеков  4 - Выполнение задания  Выбор: 4 | Элементы стека (статический массив): -> 29 8 67 27 58 77 93 71 18 42 69 42  Элементы стека (связный список): -> 29 8 67 27 58 77 93 71 18 42 69 42  Результат для массива: -> 8 29 27 67 18 71 93 42 69  Результат для связного списка: -> 8 29 27 67 18 71 93 42 69 | Выполнение задания для обычного стека |

**6. Сравнение двух реализаций стека**

*Количество итерaций 1000000*

**Время**

**Добавление элемента**

|  |  |  |
| --- | --- | --- |
| Размер стека | Статический массив | Связный список |
| 10 | 0.0000000180 | 0.0000000410 |
| 20 | 0.0000000100 | 0.0000000340 |
| 50 | 0.0000000160 | 0.0000000400 |
| 100 | 0.0000000170 | 0.0000000410 |
| 500 | 0.0000000190 | 0.0000000470 |
| 1000 | 0.0000000200 | 0.0000000480 |

**Удаление элемента**

|  |  |  |
| --- | --- | --- |
| Размер стека | Статический массив | Связный список |
| 10 | 0.0000000090 | 0.0000000270 |
| 20 | 0.0000000110 | 0.0000000220 |
| 50 | 0.0000000170 | 0.0000000400 |
| 100 | 0.0000000180 | 0.0000000260 |
| 500 | 0.0000000190 | 0.0000000230 |
| 1000 | 0.0000000200 | 0.0000000300 |

**Выполнение задания**

|  |  |  |
| --- | --- | --- |
| Размер стека | Статический массив | Связный список |
| 10 | 0.0000000830 | 0.0000006500 |
| 20 | 0.0000001580 | 0.0000012050 |
| 50 | 0.0000003860 | 0.0000029730 |
| 100 | 0.0000007650 | 0.0000061960 |
| 500 | 0.0000043980 | 0.0000316700 |
| 1000 | 0.0000092000 | 0.0000610000 |

**Занимаемая память**

|  |  |  |
| --- | --- | --- |
| Размер стека | Статический массив | Связный список |
| 10 | 4004 | 160 |
| 20 | 4004 | 320 |
| 50 | 4004 | 800 |
| 100 | 4004 | 1600 |
| 500 | 4004 | 8000 |
| 1000 | 4004 | 16000 |

**Выводы по проделанной работе**

Стек, реализованный связным списком, проигрывает по времени обработки статическому массиву, но в большинстве случаев выигрывает по памяти. Однако при приближении количества элементов к максимальному количеству элементов в стеке, связной список начинает проигрывать и по памяти.

**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?**

Если стек реализован в виде статического массива, то под него выделяется непрерывная область памяти ограниченного размера.

Если стек реализован в виде связного списка, то память выделяется постепенно, то есть при каждом добавлении элемента происходит новое выделение памяти на куче.

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

Для стека, реализованного с помощью связного списка: происходит освобождение памяти для последнего узла и смещение конца стека на один элемент.

Для стека, реализованного с помощью массива: указатель смещается назад и элемент перестает быть доступным.

**4. Что происходит с элементами стека при его просмотре?**

Они уничтожаются, потому что происходит сдвиг указателя.

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

Самым выигрышным по времени является реализация стека в виде массива, связный список имеет выигрыш по памяти лишь при небольших размерах стека, а при больших все также лидирует массив.