

Лабораторная работа №3

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

2. ТЗ:

Цель работы – приобрести навыки работы со стеком, а также реализация функций стека (удаление, добавление и тд).

Входные данные.

На входе программа получает вариант стека - 1 или 2 (массив или список) а также номера операций - 1, 2 или 3 (удаление, добавление, вывод) - производимых над стеком.

Выходные данные.

В результате программа выводит адреса ячеек памяти стека и время, затраченное на работу программы.

Обработка ошибок.

В случае переполнения стека, или если пользователь пытается удалить элемент из пустого стека, программа сообщит пользователю об ошибке.

3. СД:

В программе было использовано 2 реализации стека - массив и список. Содержимое стека - адреса памяти (стека). Структура стека для массива содержит в себе массив элементов и указатель на начало массива. Структура стека для списка содержит в себе значение элемента (адрес ячейки), а также указатель на следующий элемент (next).

Структура для списка:

```
struct NewNode {  
    struct NewNode *current;  
    struct NewNode *next;
```

```
} *Node = NULL;
```

Структура для массива:

```
typedef struct {  
int *array[5];  
unsigned head;  
} ArrayStack;
```

4. Алгоритм:

При создании стека через массив была создана структура с массивом и индексом начала массива. При создании стека через список был реализован алгоритм работы со списком (у элемента списка есть адрес и ссылка на следующий элемент списка (next)).

5. Тесты:

Функция:	Время работы (массив):	Время работы (список):	Процентное соотношение:
push (добавление)	0.0000216	0.0003390	1500 %
pop (удаление)	0.0000137	0.0001393	1000 %
print (вывод)	0.1450439	0.1697987	17 %

6. Выводы:

В результате работы были реализованы функции работы со стеком, а также были сделаны обе реализации стека - через массив и через список. Массив дает преимущество по времени, стек - по памяти. Массив использовать целесообразно при известном количестве элементов. Время вывода стека примерно одинаковое, так как на внешний вывод (печать) уходит большая часть времени. При работе программы фрагментация памяти не происходит.

7.

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

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

- Массив: Если стек реализован в виде статического или динамического массива (вектора), то для его хранения обычно отводится непрерывная область памяти ограниченного размера, имеющая нижнюю и верхнюю границу. Список: До начала работы указатель стека показывает на нулевой, физически отсутствующий адрес (т. е. указатель - пустой). При включении элемента в стек сначала происходит выделение области памяти, адрес которой записывается в указатель стека, а затем по значению этого указателя в стек помещается информация.

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

- Массив: При исключении элемента из стека сначала считываются данные, а затем происходит перемещение указателя PS к предыдущему элементу. Если указатель стека выходит за нижнюю границу массива, то стек пуст. Список: При исключении элемента 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.

4) Что происходит с элементами стека при его просмотре?

- Классическая реализация стека предполагает, что просмотреть содержимое стека без извлечения (удаления) его элементов невозможно.

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

- Массив дает преимущество по времени, стек - по памяти. Массив использовать целесообразно при известном количестве элементов стека.