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”
Вариант 6

Студент_____Гурова Наталия Алексеевна_____
фамилия, имя, отчество

Группа_____ИУ7-34Б_____

Выполнил _____ _____Гурова Н.А_____
подпись, дата *фамилия, и.о.*

Принял _____ ____Барышникова М. Ю.
___
подпись, дата *фамилия, и.о.*

2021 г.

Цель работы

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.

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.

Используя стек, определить, является ли строка палиндромом.

Входные данные

Для создания стеков необходимо ввести максимальное количество элементов в стеках. Для добавления элемента (элементов) в стеки необходимо ввести соответствующий символ (строку). Для удаления элементов необходимо ввести количество этих элементов.

Выходные данные

При выборе пункта меню номер 4 будет выведено актуальное состояние стеков, пункт 5 печатает ответ на вопрос, являются ли соответствующие стеки палиндромами, пункт номер 6 выводит освобожденные адреса, пункт номер 7 позволяет провести исследование по использованию стеками памяти и времени при различных размерностях стеков.

Способ обращения к программе

Программа может быть вызвана через консоль с помощью команды app.exe

Аварийные ситуации

Могут быть выведены такие ошибки, как:

- Неверный введенный максимальный размер стеков
- Попытка вызвать пункт меню, использующийся для обработки массива, не создав массив
- Попытка добавить элемент в уже заполненный стек
- Попытка удалить элемент из пустого стека
- Неверно введен пункт из меню

Структуры данных

Для хранения стека в виде односвязного списка использовалась следующая структура:

```
typedef struct
{
    int max_count_elems;
    int count_elems;
    Node_t *data;
} linked_list_stack_t;
```

Где:

- max_count_elems – максимальное количество элементов в стеке
- count_elems – текущее количество элементов в стеке
- data – элементы стека

Для хранения информации об элементах в стеке в виде односвязного списка использовался односвязный список, состоящий из следующих узлов:

```
typedef struct Node_t
{
    struct Node_t *next;
    char data;
} Node_t;
```

- next – указатель на следующий узел в списке
- data - значение соответствующего узла

Для хранения стека в виде статического массива использовалась следующая структура:

```
#define MAX_SIZE_OF_STACK 1000
```

```
typedef struct
{
    int max_count_elems;
    int count_elems;
    char data[MAX_SIZE_OF_STACK];
} static_stack_t;
```

- max_count_elems – максимальное количество элементов в стеке
- count_elems – текущее количество элементов в стеке
- data – элементы стека

Описание алгоритма

Данная программа предназначена консольное приложение со следующими возможными операциями, представленными в меню:

```
Menu:
    0 - EXIT
    1 - create stacks
    2 - add elems
    3 - delete elems
    4 - print stacks
    5 - check 'is stacks palindrome'
    6 - print free addresses
    7 - conduct research about memory and time
Input command:
```

Пример создания стеков:

```
Input command:1

Input max size for stack:10
Create stacks            ----> SUCCESS
```

Пример добавления элементов:

```
Input command:2
Input string:aboba
The item add to the static stack          ----> SUCCESS
The item add to the linked list stack     ----> SUCCESS
The item add to the static stack          ----> SUCCESS
The item add to the linked list stack     ----> SUCCESS
The item add to the static stack          ----> SUCCESS
The item add to the linked list stack     ----> SUCCESS
The item add to the static stack          ----> SUCCESS
The item add to the linked list stack     ----> SUCCESS
The item add to the static stack          ----> SUCCESS
The item add to the linked list stack     ----> SUCCESS
```

Пример удаления элементов:

```
Input command:3
How many elems do you want to delete?
Input count:2

From static stack you delete      ----> a
From linked list stack you delete ----> a
Address of delete elem 00000000001F1480

From static stack you delete      ----> b
From linked list stack you delete ----> b
Address of delete elem 00000000001F1460
```

Пример вывода информации о стеках:

```
Input command:4

Stacks contains 3
Free 7
Word now: abo

STATIC STACK
HEAD -->
         o
         b
         a


LINKED LIST STACK
HEAD -->
         o     address = 00000000001F1440
         b     address = 00000000001F1420
         a     address = 00000000001F1400


Used memory for static stack: 1000
Used memory for linked list stack: 48
```

Пример проверки на палиндромность:

```
Input command:5

Word now: abo
Static stack is not palindrome
Linked list is not palindrome
```

Пример вывода свободных адресов:

```
Input command:6

Addresses of all elems that you delete:
00000000001F1480
00000000001F1460
```

Пример вывода результатов исследования:

```
Input command:7

Input count of measurements that you want to do:2

Input all sizes of stack that you want to test:100 200

START------------------------------------------------------------------------------------------------------------>END
Research for size = 100     ----> SUCCESS

START------------------------------------------------------------------------------------------------------------>END
Research for size = 200     ----> SUCCESS

RESULT OF THE EXPERIMENT:
MEMORY
Size of stack  | Memory for static stack | Memory for list stack
    100                 1000                    1600
    200                 1000                    3200

TIME
Size of stack  | Add static stack | Add list stack | Del static stack | Del list stack
    100             0.00300          0.08200         0.00200         0.03200
    200             0.00900          0.07700         0.00000         0.02800

Size of stack  | Palindrome static stack | Palindrome list stack
    100               0.01160              113727.27273
    200               0.00040              454818.18182

IMPORTANT: all time results * on 1000 for clarity
```

Результаты исследования занимаемой стеками памяти и времени выполнения операций при разных размерах стеков

```
RESULT OF THE EXPERIMENT:
MEMORY
Size of stack  | Memory for static stack | Memory for list stack
     10                 1000                    160
     50                 1000                    800
    100                 1000                    1600
    250                 1000                    4000
    500                 1000                    8000
    750                 1000                    12000
   1000                 1000                    16000

TIME
Size of stack  | Add static stack | Add list stack | Del static stack | Del list stack
     10             0.00000          0.00700         0.00000         0.00600
     50             0.00300          0.02500         0.00000         0.01500
    100             0.00100          0.07400         0.00000         0.02700
    250             0.00600          0.08200         0.00300         0.01800
    500             0.00500          0.12400         0.00500         0.05400
    750             0.00100          0.05100         0.00100         0.03600
   1000             0.00400          0.03800         0.00300         0.03100

Size of stack  | Palindrome static stack | Palindrome list stack
     10               0.00140              1272.72727
     50               0.00620              26363.63636
    100               0.01280              115090.90909
    250               0.00040              741545.45455
    500               0.00070              3713181.81818
    750               0.00040              7396545.45455
   1000               0.00030              12276636.36364
```

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.

При хранении стека как массива, память удаляется при завершении программы при вызове функции free (если работа идет с динамической памятью).

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 (роль играет процент заполнения) и во времени обработки стека.

Вывод

Для реализации стека выгодно использовать массив, так как он занимает меньше памяти (для 100 элементов в 1,6 раз, для 1000 элементов уже в 16), чем список, и работает быстрее (в 6-7 раз).

Однако при использовании массива размер стека сильно ограничен. Если массив был создан статически, то его размер не изменить, если динамически – придется постоянно перевыделять память. При использовании же списка такой проблемы не возникает.