Лабораторная работа № 2

Методические указания

Работа со стеком

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.

Краткие теоретические сведения

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.

При работе со стеком необходимо отслеживать возникновение таких аварийных ситуаций, как попытки исключения элемента из пустого стека и переполнения стека. При реализации стека в виде вектора переполнение возникает вследствие ограниченности объема памяти, выделяемой под размещение массива. Однако даже если стек реализован на основе динамического списка, его размер также не безграничен, он ограничен объемом доступной программе оперативной памяти. Для отработки указанных аварийных ситуаций в программе должны быть предусмотрены дополнительные проверки на пустоту и переполнение стека и выдача соответствующих сообщений.

Любой тип данных в языке программирования определяет множество допустимых значений переменных и множество допустимых операций над ними, поэтому можно определить стек как некий **абстрактный тип данных (АТД).** Преимущество использования АТД заключается в том, что в этом случае описание

данных и операций, выполняемых над ними, отделено от их представления и реализации. Таким образом, АТД специфицирует данные и операции, которые могут быть применены к этим данным.

К основным операциям, выполняемым над стеком, относятся: включение элемента в стек и исключение элемента из стека.

Реализация стека в виде массива. 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. Перед началом работы указатель стека PS находится ниже левой (нижней) границы массива. При включении первого элемента в стек указатель PS устанавливается на начало массива и по адресу первого элемента размещается записываемое значение. При попытке добавить в стек каждый последующий элемент сначала происходит смещение указателя на длину типа данных (т.е. он перемещается к следующему элементу массива), а затем – размещение значений элементов по этим адресам. Если указатель выходит за верхнюю границу массива, то это признак того, что стек переполнен. 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.

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.

Реализация стека в виде списка. 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.

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.

При физической реализации стека в виде односвязного линейного списка дескриптор будет отличаться отсутствием верхней границы стека. В этом случае объем стека ограничивается только объемом доступной оперативной памяти.

Преимущество использования АТД заключается в том, что в программе общий алгоритм программы по работе со стеком не зависит от конкретной реализации самого стека. Однако для реализации каждой операции (включения элемента, исключения элемента, проверки на пустоту, проверки на переполнение) должна быть написана подпрограмма, зависящая от конкретной реализации структуры данных.

Задание

Разработать программу работы со стеком, реализующую операции добавления и удаления элементов из стека и отображения текущего состояния стека. Реализовать стек: а) массивом; б) списком.

Все стандартные операции со стеком должны быть оформлены отдельными подпрограммами. В случае реализации стека в виде списка при отображении текущего состояния стека предусмотреть возможность просмотра адресов элементов стека и создания дополнительного собственного списка свободных областей (адресов освобождаемой памяти при удалении элемента, который можно реализовать как списком, так и массивом) с выводом его на экран. Список свободных областей необходим для того, чтобы проследить, каким образом происходит выделение памяти менеджером памяти при запросах на нее и убедиться в возникновении или отсутствии фрагментации памяти..

Указания к выполнению работы

Интерфейс программы должен быть понятен неподготовленному пользователю. При разработке интерфейса программы следует предусмотреть:

- указание формата и диапазона вводимых данных,
- блокирование ввода данных, неверных по типу,
- указание операции, производимой программой:
  - добавление элемента в стек,
  - удаление элемента из стека,
  - вычисление (обработка данных);
- наличие пояснений при выводе результата.

Кроме того, нужно вывести на экран время выполнения программы при реализации стека списком и массивом, а также указать требуемый объем памяти.

При тестировании программы необходимо:

- проверить правильность ввода и вывода данных (в том числе, отследить попытки ввода данных, неверных по типу);
- обеспечить вывод сообщений при отсутствии входных данных («пустой ввод»);
- проверить правильность выполнения операций;
- обеспечить вывод соответствующих сообщений при попытке удаления элемента из пустого стека;
- отследить переполнение стека.

При реализации стека в виде списка необходимо:

- ограничить доступный объем оперативной памяти путем указания:
  - максимального количества элементов в стеке;
  - максимального адреса памяти, превышение которого будет свидетельствовать о переполнении стека;
- следить за освобождением памяти при удалении элемента из стека.

Содержание отчета

В отчете по лабораторной работе должен быть сделан вывод о том, в каких случаях для реализации стека целесообразно использовать массив, а в каких список. Кроме того, следует указать, какие преимущества дает та или иная реализация. Выводы должны быть подкреплены результатами сравнений объёмов памяти и времён выполнения программы при использовании списка и массива. Результаты должны быть представлены, как в количественном, так и в процентном соотношениях.

В отчете по лабораторной работе должны быть даны ответы на следующие вопросы:

1. Что такое стек?
2. Каким образом и сколько памяти выделяется под хранение стека при различной его реализации?
3. Каким образом освобождается память при удалении элемента стека при различной реализации стека?
4. Что происходит с элементами стека при его просмотре?
5. Каким образом эффективнее реализовывать стек? От чего это зависит?

Отчет должен быть представлен в электронном или печатном виде.

Список рекомендуемой литературы

1. *Вирт Н.* Алгоритмы и структуры данных: Пер. с англ.- СПб.: Невский диалект, 2001. С. 69–71, 205–235
2. *Ахо А., Хопкрофт Д., Ульман Д.* Структуры данных и алгоритмы.: Пер. с англ. М.: Издат. дом «Вильямс», 2000. С. 58–76
3. *Кормен Т., Лейзерсон Ч., Ривест Р.,* Алгоритмы: построение и анализ: Пер. с англ. М.: МЦНМО, 2001. С. 194–198