УДК 004.432

Степович-Цветкова Г.С.
кандидат экономических наук, доцент,
ФГБОУ ВО «Ивановский государственный университет»,
Россия, г. Иваново

ИСПОЛЬЗОВАНИЕ УКАЗАТЕЛЕЙ ДЛЯ РАБОТЫ С МАССИВАМИ В РЕКУРСИВНЫХ АЛГОРИТМАХ НА ЯЗЫКЕ С++

Обосновывается целесообразность динамического выделения памяти под массивы при необходимости их передачи в качестве параметров в рекурсивные функции. Рассматривается вопрос обращения к элементам массива с помощью указателей.

Ключевые слова: указатели, С++, динамические массивы, рекурсивные алгоритмы, переполнение стека.

При написании программ на языке С++ широко используется такая структура данных как массив, представляющий собой определенный одним именем набор однотипной информации заданного размера, память под который может выделяться статически – во время компиляции программы, или динамически – во время ее выполнения.

При передаче в функции в качестве параметра статического массива происходит его полное копирование, приводящее к затрате дополнительных объемов памяти, что существенно при больших количествах информации. В то же время, при работе с массивами, объявленными динамически, в функции передается лишь указатель на массив.

Динамическое выделение памяти основано на работе с указателями – переменными, в которых хранятся адреса других переменных или функций, объявляемых при помощи символа * (звездочка). Выделяет динамическое пространство памяти операция new, в зависимости от операнда может быть выделена одиночная переменная или массив. После работы с динамической памятью ее необходимо очистить операцией delete.

Для работы с указателями используются два оператора: & – взятие адреса (возвращает адрес, по которому находится объект, указанный после данной операции); * – разыменование указателя (возвращает значение объекта, на который указывает указатель, записывается перед именем указателя).

Имя массива – это указатель на его первый элемент, поэтому, если объявлен динамический массив a, то к первому элементу массива можно обратиться двумя способами: a[0] или *a.

К указателям можно применять арифметические операции. Пусть p – указатель, i – произвольное целое или символьное выражение. Тогда p+i возвращает p+i*sizeof(*p), то есть p сдвигается вправо на i размеров объекта, на который указывал p. К примеру, если p указывал на первый элемент массива типа double, p+1 будет указывать на следующий элемент массива, а не на следующий байт памяти. Аналогично используются операции вычитания, префиксных и постфиксных инкремента и декремента.

Таким образом, с помощью арифметических операций над указателями можно обратиться к любому элементу массива. Так, в результате выполнения оператора x=*(a+2) в переменную x будет записан элемент массива a с индексом 2 (a[2]), так как сначала указатель сдвигается на 2 позиции, затем операция разыменования предоставляет доступ к значению этого элемента.

Итак, для работы с динамическим массивом, необходимо:

1. Описать переменную-указатель (назовем p) определенного типа, который должен соответствовать типу элементов.

2. Начиная с адреса переменной p с помощью операции new выделить участок памяти необходимого размера (под n элементов заданного типа). Тогда к i-му элементу динамического массива p можно обратиться двумя способами: *(p+i) или p[i].

3. Когда участок памяти больше не нужен, его следует освободить с помощью операции delete.

Двумерный динамический массив удобно представлять с помощью одномерного, в который по порядку записываются все строки матрицы: сначала все элементы нулевой строки, потом все элементы первой строки и т.д. Для хранения двумерного массива из n строи и m столбцов необходимо выделить место памяти под одномерный массив размера nxm элементов.

Обратим внимание на то, что для обращения к элементу a[i,j] необходимо определить его положение в одномерном массиве. Чтобы в одномерном массиве дойти до элемента a[i,j], нужно, начиная с нулевого элемента, пропустить i строк по m элементов и j элементов в строке i, то есть указатель с нулевого элемента нужно сдвинуть на i*m+j ячеек. Поэтому обращение к элементу a[i,j] записывается как a[i*m+j] или *(a+i*m+j).

Объем занимаемой памяти параметрами функции имеет определяющее значение при работе с рекурсивными функциями, то есть такими функциями, которые вызывают сами себя. При этом в рекурсивных функциях обязательно должно быть предусмотрено условие выхода из рекурсии, чтобы не произошло бесконечного зацикливания. Как правило, заранее не известно, сколько шагов рекурсии будет произведено. И если при этом каждый раз происходит копирование массивов больших размерностей, то может произойти переполнение стека памяти, что приведет к фатальной ошибке (Stack Overflow). Переполнения памяти не произойдет, если вместо статического массива использовать динамический, передавая при этом в функции лишь указатель на первый элемент массива и его размерность.

Таким образом, при работе с рекурсивными алгоритмами использование динамических массивов позволяет избежать проблем с переполнением памяти компьютера и сбоем программы по этой причине.

Список литературы

1. Степович-Цветкова, Г. С. Языки и технологии программирования. Ч. I : Структурное программирование на языке C++ [Электронный ресурс] : учебное пособие для студентов бакалавриата направлений "Математика", "Математика и компьютерные науки", "Фундаментальная информатика и информационные технологии", "Информационная безопасность" / Г.С. Степович-Цветкова. – Электрон. дан. – Иваново : Иван. гос. ун-т, 2016. – 1 электрон. опт. диск (CD-ROM).

2. Шилдт Г. Искусство программирования на C++. – СПб. : БХВ-Петербург, 2005. – 496 с.