

Практическое применение связанных списков в программировании.

Автор: Четвериков Д.А.

В информатике, связный список - структура данных, состоящая из узлов, каждый из которых содержит как собственные данные, так и одну или две ссылки («связки») на следующий и или предыдущий узел списка. Принципиальным преимуществом перед массивом является структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями.

Односвязный список (Однонаправленный связный список)

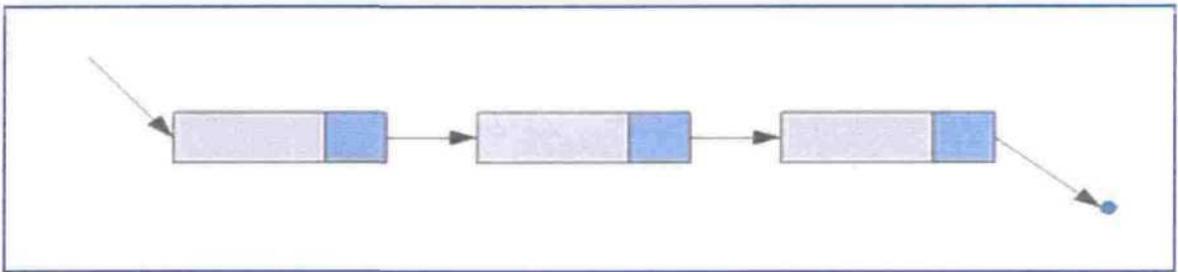


рис. 1 Односвязный список

В односвязном списке можно передвигаться только в сторону конца списка. По сходной схеме работает новогодняя гирлянда. 2 и последующие лампочки не загорятся, пока не загорится первая. Узнать адрес предыдущего элемента, опираясь на содержимое текущего элемента невозможно.

Двусвязный список (Двунаправленный связный список)

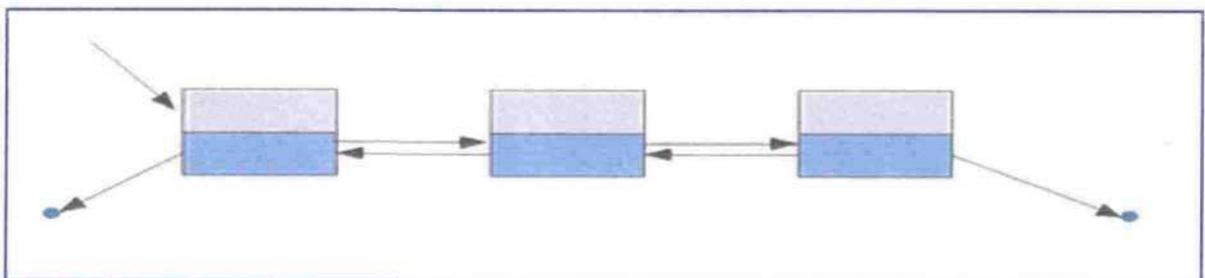


рис. 2 Двусвязный список

По двусвязному списку можно передвигаться в любом направлении - как к началу, так и к концу. В этом списке проще производить удаление и перестановку

элементов, так как всегда известны адреса тех элементов списка, указатели которых направлены на изменяемый элемент.

Также разновидностью связанных списков является кольцевой (циклический, замкнутый) список. Он тоже может быть односвязным или двусвязным. Последний элемент кольцевого списка содержит указатель на первый, а первый (в случае двусвязного списка) - на последний.

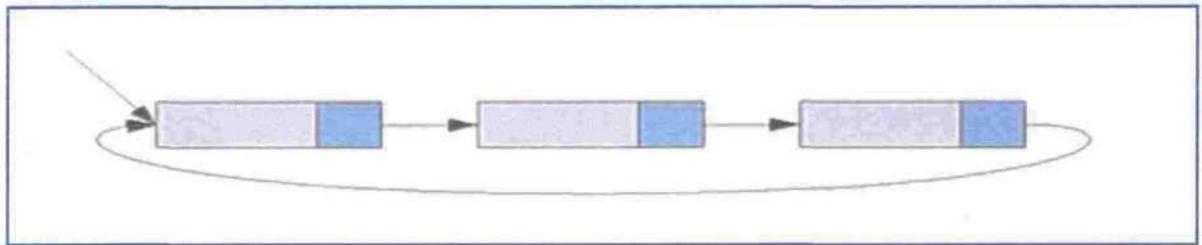


рис. 3 Кольцевой список.

Реализация такой структуры происходит на базе линейного списка. С каждым кольцевым списком есть указатель на первый элемент. В этом списке константы NULL не существует.

NTFS система - яркий пример реализации связанного списка. Она основана на структуре B+ дерева. Которая является примером реализации двусвязного списка.

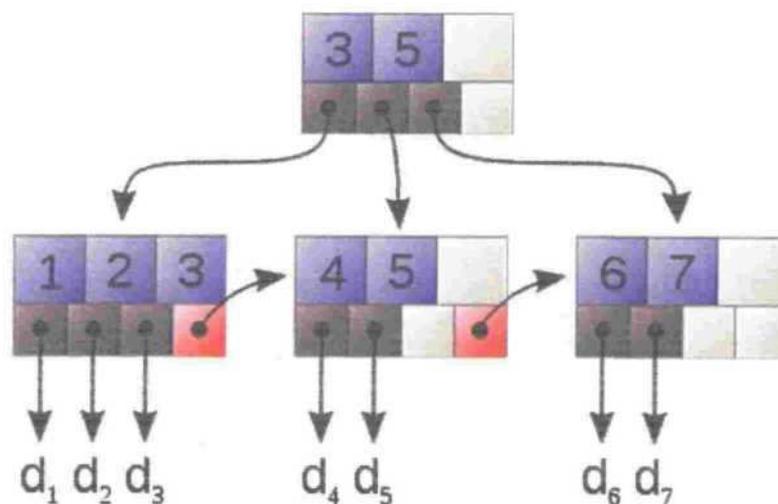


рис. 4 NTFS система

В FAT реализуется односвязный список.



рис. 5 FAT система

Связные списки наиболее часто используют при создании, редактировании и использовании СУБД.

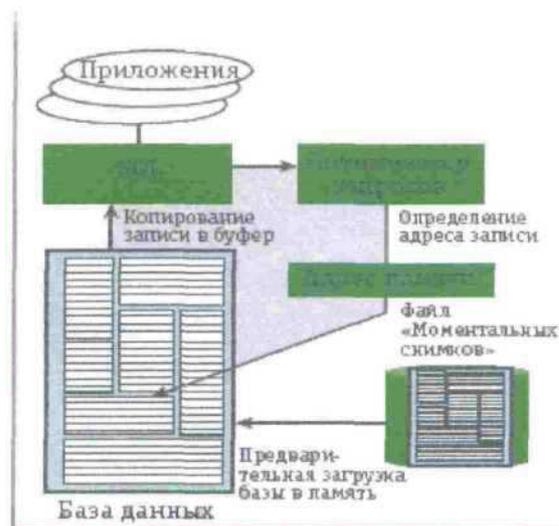


рис. 6 СУБД

Работа Интернет соединения тоже реализована на основании связанных списков.

Подводя итог, хочется отметить основные достоинства связанных списков. А именно:

- лёгкость добавления и удаления элементов
- размер ограничен только объёмом памяти компьютера и разрядностью указателей

• динамическое добавление и удаление элементов

К недостаткам данной технологии относится:

- сложность определения адреса элемента по его индексу (номеру) в списке

- на поля-указатели (указатели на следующий и предыдущий элемент) расходуется дополнительная память (в массивах, например, указатели не нужны)
- работа со списком медленнее, чем с массивами, так как к любому элементу списка можно обратиться, только пройдя все предшествующие ему элементы
- элементы списка могут быть расположены в памяти разреженно, что окажет негативный эффект на кэширование процессора
- над связными списками гораздо труднее (хотя и в принципе возможно)
- производить параллельные векторные операции, такие как вычисление суммы
- кэш-промахи при обходе списка.

Список используемой литературы

1. Связный список [Электронный ресурс]: (с изм. и доп.)- Режим доступа: http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%8F%D0%B7%D0%BD%D1%8B%D0%B9_%D1%81%D0%BF%D0%B8%D1%81%D0%BE%D0%BA;(дата обращения 15.10.2011)
2. Связный список [Электронный ресурс]: (с изм. и доп.)- Режим доступа: <http://www.codenet.ru/progr/cpp/dlist.php>: (дата обращения 15.10.2011)
3. Дэвис Стефан Р. С++ для «чайников» / М.: Издательский дом "Вильямс". 2003. С.256