

Связный список

Материал из Википедии — свободной энциклопедии

Свя́зный спи́сок — базовая динамическая структура данных в информатике, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну или две ссылки («связки») на следующий и/или предыдущий узел списка.^[1] Принципиальным преимуществом перед массивом является структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера^[2], а порядок обхода списка всегда явно задаётся его внутренними связями.

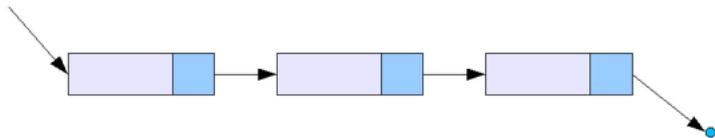
Содержание

Виды связных списков
Линейный связный список
Односвязный список (однонаправленный связный список)
Характеристики
Односвязный список в языках программирования
Двусвязный список (двунаправленный связный список)
XOR-связный список
Кольцевой связный список
Список с пропусками
Развёрнутый связный список
Достоинства
Недостатки
См. также
Примечания

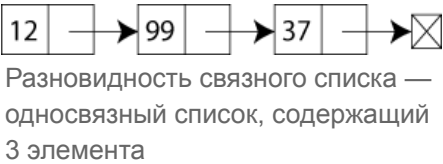
Виды связных списков

Линейный связный список

Односвязный список (однонаправленный связный список)



Линейный однонаправленный список — это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. Каждый элемент списка имеет указатель на следующий элемент. Последний элемент списка указывает на NULL. Элемент, на который нет указателя, является



первым (головным) элементом списка. Здесь ссылка в каждом узле указывает на следующий узел в списке. В односвязном списке можно передвигаться только в сторону конца списка. Узнать адрес предыдущего элемента, опираясь на содержимое текущего узла, невозможно.

В информатике **линейный список** обычно определяется как абстрактный тип данных (АТД), формализующий понятие упорядоченной коллекции данных. На практике линейные списки обычно реализуются при помощи массивов и связных списков. Иногда термин «список» неформально используется также как синоним понятия «связный список». К примеру, АТД нетипизированного изменяемого списка может быть определён как набор из конструктора и основных операций:

- Операция, проверяющая список на пустоту.
- Три операции добавления объекта в список (в начало, конец или внутрь после любого (n-го) элемента списка);
- Операция, вычисляющая первый (головной) элемент списка;
- Операция доступа к списку, состоящему из всех элементов исходного списка, кроме первого.

Характеристики

- **Длина списка.** Количество элементов в списке.
- Списки могут быть **типизированными** или **нетипизированными**. Если список типизирован, то тип его элементов задан, и все его элементы должны иметь типы, совместимые с заданным типом элементов списка. Чаще списки типизированы.
- Список может быть **сортированным** или **несортированным**.
- В зависимости от реализации может быть возможен **произвольный доступ** к элементам списка.

Односвязный список в языках программирования

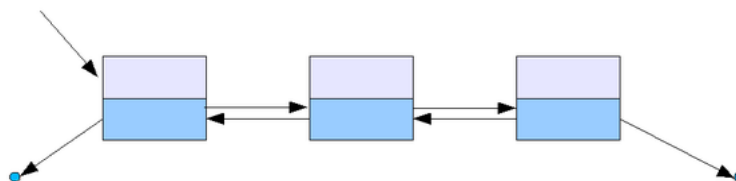
Си

```
typedef struct s_list
{
    int field; // поле данных
    struct s_list *next; // указатель на следующий элемент
} list;
```

применение односвязного списка:

```
1 list* l1 = (list*)malloc(sizeof(list));
2 l1->field = 1;
3 l1->next = (list*)malloc(sizeof(list));
4 l1->next->field = 2;
5 l1->next->next = (list*)malloc(sizeof(list));
6 /* и т.д. */
```

Двусвязный список (двунаправленный связный список)



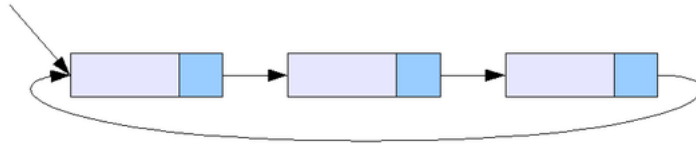
Здесь ссылки в каждом узле указывают на предыдущий и на последующий узел в списке. Как и односвязный список, двусвязный допускает только последовательный доступ к элементам, но при этом дает возможность перемещения в обе стороны. В этом списке проще производить удаление и перестановку элементов, так как легко доступны адреса тех элементов списка, указатели которых направлены на изменяемый элемент.

XOR-связный список

Используется редко.

Кольцевой связный список

Разновидностью связных 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.



Как правило, такая структура реализуется на базе линейного списка. С каждым кольцевым списком дополнительно хранится указатель на первый элемент. В этом списке ссылки на NULL не встречается.

Также существуют циклические списки с выделенным головным элементом, облегчающие полный проход через список.

Список с пропусками

Развёрнутый связный список

Достоинства

- эффективное (за константное время) добавление и удаление элементов
- размер ограничен только объёмом памяти компьютера и разрядностью указателей
- динамическое добавление и удаление элементов

Недостатки

Недостатки связных списков вытекают из их главного свойства — последовательного доступа к данным:

- сложность прямого доступа к элементу, а именно определения физического адреса по его индексу (порядковому номеру) в списке
- на поля-указатели (указатели на следующий и предыдущий элемент) расходуется дополнительная память (в массивах, например, указатели не нужны)
- некоторые операции со списками медленнее, чем с массивами, так как к произвольному элементу списка можно обратиться, только пройдя все предшествующие ему элементы
- соседние элементы списка могут быть распределены в памяти нелокально, что снизит эффективность кэширования данных в процессоре
- над связными списками, по сравнению с массивами, гораздо труднее (хоть и возможно) производить параллельные векторные операции, такие, как вычисление суммы: накладные расходы на перебор элементов снижают эффективность распараллеливания

См. также

- Сортировка связного списка

Примечания

1. Cormen, Leiserson, Rivest, and Stein. Introduction to Algorithms, 2nd edition. The MIT Press, 2001. ISBN 0-262-03293-7
 2. Выравнивание данных
-

Источник — https://ru.wikipedia.org/w/index.php?title=Связный_список&oldid=89775498

Эта страница последний раз была отредактирована 21 декабря 2017 в 04:44.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.

Свяжитесь с нами