

приема или передачи пакета. Функции IPX, принимающие или передающие пакет, не выполняют операции ожидания полного завершения приема или передачи. Они сразу возвращают управление вызвавшей их программе. После завершения операции приема/передачи в соответствующем поле блока ЕСВ устанавливается признак. Программа может периодически опрашивать ЕСВ для обнаружения признака завершения операции. Есть и другая возможность проверки выполнения операции приема/передачи. В блоке ЕСВ можно указать адрес процедуры, которая будет вызвана при завершении операции передачи данных. При таком способе нет необходимости тратить время на периодическую проверку блока ЕСВ. Такова приблизительная схема использования драйвера IPX для организации взаимодействия между рабочими станциями локальной вычислительной сети по схеме «клиент-сервер».

Статья не претендует на руководство по написанию подобных программ, а лишь указывает наиболее существенные моменты при разработке сетевых программ, не использующих системные вызовы файлового сервера. Что же касается законченных разработок, использующих протоколы обмена IPX/SPX, то авторами создан целый ряд подобных программ, которые с успехом применяются в локальных сетях различных учреждений нашего региона и Украины.

SUMMARY

The article is devoted to aspects of using data exchange IPX net protocol for organizing interaction between workstations without calling server. Practical recommendations for writing low-level functions to work with IPX is given.

СПИСОК ЛИТЕРАТУРЫ

1. А.В.Фролов, Г.В.Фролов. Локальные сети персональных компьютеров. Использование протоколов IPX, SPX, NETBIOS. М.: Диалог-МИФИ, 1993.
2. Барри Нанс. Программирование в локальных сетях. Пермь: 1992.
3. NOVELL Netware Version 3.11. Novell Inc. USA, 1992.
4. А.Ермолов. Что такое IPX и как его использовать в своих целях //Монитор, 1992, №7,8.

Поступила в редколлегия 18 января 1995г.

УДК 681.3.06

КОМБИНИРОВАНИЕ МОДЕЛЕЙ ДАННЫХ КАК РАДИКАЛЬНЫЙ ПУТЬ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРОГРАММ ОБРАБОТКИ БАЗ ДАННЫХ

Любчак В.А., доц., Лученко С.В., инж.

В этой статье на основе сравнительного анализа достоинств и недостатков реляционной и сетевой моделей баз данных (БД) указан один из возможных путей повышения эффективности программ, обрабатывающих базы данных.

Рассматривается распространенный случай представления базы данных в виде набора связанных файлов. При этом становится актуальной проблема выбора модели данных. Модель данных (или модель базы данных) определяет каким образом осуществляются взаимосвязи между записями связанных файлов внутри базы данных. От удачного выбора модели зависит эффективность работы создаваемой прикладной программы, затраты на ее разработку.

В реляционной модели база данных рассматривается как множество двумерных таблиц. Столбцы таблицы соответствуют полям данных, а строки таблицы - записям. При разделении информации на несколько таблиц объем избыточной информации уменьшается. Чтобы сохранить

соответствие двух таблиц, в реляционной модели между таблицами устанавливают связи через общее поле данных, которое должно быть определено в обеих таблицах, а индекс обязательно должен существовать. В результате установления таких связей БД оказывается представленной в виде виртуальной таблицы, составленной из нескольких таблиц, связанных общим столбцом (посредством индекса). С помощью такого табличного представления базы данных особенно просто манипулировать стандартными операциями реляционной модели [1, 2].

Принципиальная разница между реляционной и сетевой моделями заключается в том, что в сетевой модели базы данных связи между типами записей точно определяются и непосредственно обслуживаются с помощью множеств. Множество определяет отношение типа "один к нескольким" между двумя типами записей, непосредственно поддерживая связь между записями, используя указатели записей. Сетевая модель исключает избыточность данных, так как связывает два типа записей непосредственно, не нуждаясь в дублировании поля и в индексном файле. Можно сделать справедливый вывод о том, что сетевая модель является более реляционной, чем реляционная модель, поскольку в сетевой модели содержится больше информации о том как связаны записи. Структура базы данных, построенной на основе данной модели, напоминает сеть (поэтому модель получила название сетевой).

Каждая из моделей обладает своими достоинствами и недостатками.

Реляционная модель систем управления базами данных чрезвычайно популярна, в основном благодаря тому, что модель базы данных, лежащая в ее основе, проста и упрощает использование данных. Эта модель данных лежит в основе всех xBase-подобных СУБД (dBase, Clipper, FoxPro и т.д.), а также многих других [3].

Наоборот, основным преимуществом сетевой модели являются более высокая эффективность, не столь жесткие требования к памяти, а также лучшее обеспечение целостности данных. Еще одним достоинством сетевой модели является то, что доступ к связанной записи осуществляется непосредственно с помощью одного оператора чтения, в то время как в реляционной модели необходимо сначала прочесть индекс, а затем связанную запись.

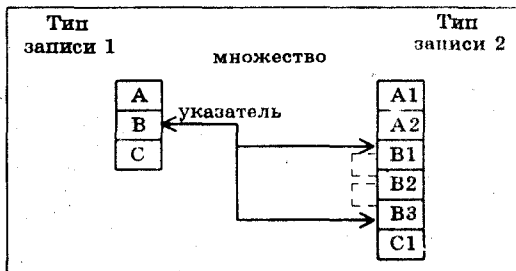
Предложенное в статье решение позволяет комбинировать обе модели, устраняя присущие им недостатки. Исходя из собственного опыта, авторы считают наиболее целесообразным для написания эффективных программ обработки баз данных использование языка СИ и собственных или уже готовых библиотек для взаимодействия с базами данных. Разные библиотеки имеют разные возможности, но тем не менее все они могут предоставить разработчикам программ некий минимальный набор функций, позволяющий осуществлять доступ к БД. И поэтому свой выбор следует основывать именно на различиях библиотек: наличие дополнительных функций, скорость обработки файлов данных, легкость освоения, удобство в работе, размеры программ. Так, библиотека CodeBase имеет набор функций, очень похожих на xBase-языки не только по выполняемым действиям, но и по названиям. Ее наиболее легко освоить, но она более громоздка и обладает худшими характеристиками по скорости обработки. Библиотека DBS позволяет осуществлять доступ к БД на нижнем уровне, манипулировать средней выполнением программ. Библиотека db_Vista обладает возможностью использовать различные модели данных (реляционную и сетевую) [4].

Именно комбинированное использование двух моделей данных позволяет добиться наилучших показателей эффективности программ - высокой скорости обработки данных, меньших размеров программ и файлов данных.

Исходя из собственного опыта разработки такого рода программ, авторы считают, что в случае больших прикладных программ, когда требуется осуществлять самые разные манипуляции с БД,

целесообразнее использовать уже готовые библиотеки, например, такие, как db_Vista фирмы Raima. В частности, авторы в одном из проектов при разработке информационно-поисковой системы успешно применяли библиотеку db_Vista v.2.41 совместно с компилятором Microsoft C 6.0. Но для написания небольших высокопроизводительных программ более эффективным является использование собственных библиотек функций обработки БД.

Создание подобных библиотек не является такой уж сложной задачей. Представление информации в реляционной базе чрезвычайно просто. Сложность представляет лишь создание индексного файла, для чего наиболее целесообразно использовать одну из модификаций системы В-деревьев [1, 2]. А для сетевой модели применяется следующая структура данных:



Тип записи в соотношении "один к нескольким" со стороны "один" называется "старшей" записью. Тип записи со стороны "несколько" называется "подчиненной" записью.

Каждый экземпляр "старшей" записи содержит:

- количество всех "подчиненных" записей во множестве;
- указатель на первую "подчиненную" запись;
- указатель на последнюю "подчиненную" запись.

Каждый экземпляр "подчиненной" записи содержит:

- указатель на "старшую" запись;
- указатель на следующую "подчиненную" запись;
- указатель на предыдущую "подчиненную" запись.

Результатом является сеть связанных записей.

Сочетание методов сетевой и реляционной моделей обеспечивает эффективную организацию и доступ к информации независимо от сложности данных. Объединение методов дает огромный выигрыш в скорости, а также позволяет свести к минимуму избыточность данных.

Индексный метод, используемый в реляционных БД, позволяет найти положение записи с помощью ключа. Ключ может точно совпадать с содержанием поля, тогда указатель установится прямо на запись, или совпадать приблизительно, и тогда указатель установится перед записью, значение ключа которой является более высоким в сортирующей последовательности. Можно также установить указатель на первой или последней записи в соответствии с индексом. Когда позиция выбрана, можно перемещаться к предыдущему или последующему ключу в сортирующей последовательности. Ключи обрабатываются и управляются в том порядке, который поддерживается в индексе, независимо от физического упорядочения тех записей, на которые они указывают.

Метод множеств, используемый в сетевых БД, дает возможность перемещаться по связям множества в различных направлениях. Можно переместиться из позиции, соответствующей главному элементу множества, к первому или последнему членам множества. От члена множества можно переместиться к записи предыдущего или последующего члена, или к записи главного элемента множества.

В одних случаях более эффективной оказывается реляционная модель и индексный метод доступа, а в других случаях - сетевая модель и

доступ к записям, используя метод множеств. Хороший проект базы данных, как правило, использует оба метода: реляционный там, где требуется независимость данных, и сетевой, где требуется оптимальное представление связей. Каждый метод обладает своими способами поиска и изменения позиции в базе данных. Использование одного не должно препятствовать использованию другого. Общее для них всех - текущая запись. При использовании комбинированной модели данных доступ к записям может быть осуществлен через навигацию в множествах, через ключевые поля (с помощью быстрого просмотра или индекса) или через объединение того и другого.

Таким образом, именно применение комбинированной модели данных, построенной на основе реляционной и сетевой моделей, обеспечивает написание наиболее эффективных программ, гораздо более оптимальное проектирование базы данных и наилучшим образом удовлетворяет требованиям многих прикладных задач.

SUMMARY

The article gives comparative analysis of relation and net data models. Several programming facilities are described. The performance enhancing problem for data base processing is considered. Recommendations for designers and programmers are given.

СПИСОК ЛИТЕРАТУРЫ

1. И.Флорес. Структуры и управление данными. - М.: Финансы и статистика, 1982.
2. К.Дейт. Руководство по реляционной СУБД DB2. - М.: Финансы и статистика, 1988.
3. А.А.Мячев. Персональные ЭВМ: энциклопедический справочник. - М.: Финансы и статистика, 1992.
4. Инструментальные средства языка С для разработки баз данных // Д-р Добба, 1991, №4.

Поступила в редакцию 18 января 1995г.

УДК 547.42:667.633

ВЛИЯНИЕ КОНЦЕНТРАЦИИ ОЛИГОМЕРОВ АЛЛИЛЬНОГО ТИПА В ПЛЕНКЕ НА КИНЕТИКУ ТРЕХМЕРНОЙ ПОЛИМЕРИЗАЦИИ

Салистый С.М., проф., Дыченко Т.В., преп.

Зависимость параметров трехмерной полимеризации соединений с активной л-связью (винильных) от концентрации субстрата хорошо изучена [1]. В то же время отсутствуют литературные данные, касающиеся аналогичной зависимости для соединений с малоактивной л-связью (аллильных). На практике, неполимеризационноспособные соединения (пластификаторы, высококипящие растворители, диспергаторы и др.) нередко входят в состав композиций на основе непредельных олигомеров аллильного типа и влияют на концентрацию полимеризующегося субстрата. Поэтому вопрос о влиянии концентрации субстрата на кинетику полимеризации аллильных олигомеров имеет определенное практическое значение.

Определение связи скорости и глубин радикальной полимеризации с концентрацией субстрата в случае аллильных соединений усложняется развитой передачей цепи на мономер и возможным ингибированием полимеризации малоактивными аллильными радикалами [2]. В связи с этим, для исследования был выбран пленочный вариант полимеризации в присутствии кислорода. Тем самым, малоактивные аллильные радикалы превращаются при взаимодействии с кислородом в активные пероксидные. Инициирование процесса осуществляется за счет вырожденного разветвления цепи на продуктах реакции (аллильных гидропероксидах) с участием солей металлов переменной валентности.