7953

ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

7953

Экз. чит. зала
10 - 7953

Д.Д.Арнаудов

АНАЛИЗ МЕТОДОВ ДОСТУПА ИНФОРМАЦИИ К ВНЕШНИМ ЗУ НА МАГНИТНЫХ ДИСКАХ ПРИ РАБОТЕ ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ (ИПС), РЕАЛИЗОВАННОЙ НА ЭВМ ТРЕТЬЕГО ПОКОЛЕНИЯ

1974

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И АВТОМАТИЗАЦИИ

10 - 7953

Д.Д.Арнаудов

АНАЛИЗ МЕТОДОВ ДОСТУПА ИНФОРМАЦИИ К ВНЕШНИМ ЗУ НА МАГНИТНЫХ ДИСКАХ ПРИ РАБОТЕ ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ (ИПС), РЕАЛИЗОВАННОЙ НА ЭВМ ТРЕТЬЕГО ПОКОЛЕНИЯ

Направлено в сборник "Цифровая вычислительная техника и программирование"

ОИЯИ
БИБЛИОТЕКА

При работе ИПС приходится иметь дело с большими информационными массивами, которые надо хранить в памяти машины и обрабатывать с большими скоростями. Но стоимость оперативной памяти существующих ЭВМ (например, на магнитных сердечниках) очень высока, и поэтому необходимо иметь дополнительную память (т.н. внешнюю память), которая является не столь быстродействующей и удобной, но которая гораздо дешевле. Для этой цели можно, конечно, использовать и магнитные ленты, но в таком случае требуется много времени для поиска информации.

Более удобные носители информации – это магнитные диски; они являются устройствами с произвольным доступом (это такие устройства, в которых время для доступа к любой части информации относительно не зависит от места расположения этой информации на МД).

Магнитные диски являются неразрывной частью внешней памяти ЭВМ третьего поколения. Более подробно устройство дисков и их принцип работы описаны в /6/.

Мы остановимся на вопросах, которые имеют прямое отношение к программистам и разработчикам информационных систем; вопросах, связанных с методикой использования магнитных дисков для сохранения и обработки информационных массивов в режиме произвольного доступа.

В литературе /I,5,6/ встречаются различные определения и классификация разных видов доступов к информации на магнитном диске. Но мы считаем, что можно выделить два основных вида доступа:

а) последовательный;

б) произвольный.

В случае последовательного доступа магнитные диски используются как быстрая магнитная лента, и здесь нет ничего принципиально нового.

Произвольный доступ может быть

I) индексно-последовательным;

2) прямым.

Все остальные виды доступа к информации на МД можно рассматривать как некоторую комбинацию, разновидность или частный случай уже упомянутых видов.

Индексно-последовательный метод

Наборы данных, организованных индексно-последовательным методом, должны записываться с ключами. Программы индексно-последовательного доступа обеспечивают обращение к записям из набора данных последовательно в соответствии с порядком ключей, несмотря на то, что расположение этих записей на диске может быть совсем другим. Кроме того, используя конкретный ключ, можно обращаться к записям в произвольном порядке.

Для того, чтобы данный метод доступа стал яснее, рассмотрим некоторую конкретную реализацию этого метода - индексно-последовательный метод системы СДС[7]. Подобный метод для системы IBM 360 показан в[6]. Несмотря на универсальность индексно-последовательного метода, имеются некоторые принципиально различные способы его реализации в виде стандартных программ на разных вычислительных машинах.

Данные, организованные индексно-последовательным способом в

системе СДС, расположены в 2-х видах блоков: блоки данных и индексные блоки. Величина каждого блока кратна одной логической единице информации. Одна логическая единица информации - это то количество, которым обмениваются оперативная память и диск за время, равное времени одного доступа. В данном случае это 64 шестидесятиразрядных слова. Таким образом, конструктору системы дается возможность самому рассчитывать относительное время обработки индексно-последовательного массива.

Каждая запись массива должна иметь, как уже отмечено, единственный ключ. Ключи используются для определения последовательного порядка записей массива.

Итак, индексно-последовательный массив состоит из двух основных частей:

I. Блоки данных, содержащие записи и информацию о ключах, которые идут последовательно как логически, так и физически. Эти блоки могут быть произвольно расположены на диске, но посредством адресов связи они логически связаны.

2. Индексные блоки содержат записи ключей, которые идентифицируют соответствующие блоки данных.

Пользователь определяет размеры блоков данных и индексных блоков, но не манипулирует информацией внутри блоков. Это делается при помощи стандартных программ матобеспечения реализации индексно-последовательного метода на ЭВМ.

Чтобы легче понять суть метода, рассмотрим структуру блоков данных и индексных блоков.

Блоки данных

Каждый блок данных состоит из заголовка, записи данных и ключей для записей. Все блоки данных для данного массива имеют одну и ту же размерность и являются отдельными логическими записями. Имеется возможность оставлять "пустое" место в блоке для добавления новых данных. Таким образом, если предполагается, что массив будет расти, можно зарезервировать место для новых данных в каждом блоке.

Общая структура блока данных показана на рис.I. Каждый блок имеет заголовок, в котором записывается адрес следующего по порядку блока данных. Кроме того, там указано и количество слов, зарезервированных для будущих записей, и начало этой зарезервированной области.

Каждая запись данных тоже имеет заголовок, который показывает число слов записи и число неиспользованных разрядов в последнем слове.

Информация о ключах в блоке содержит сам ключ и указатель записи, к которой относится этот ключ. Указатель представляет собой адрес заголовка записи данных относительно начала блока. Если ключ является частью записи, он остается как часть записи. Физически информация о ключах располагается таким образом, что самый маленький номер ключа находится в конце блока. Резервная часть, если такая имеется, находится в середине блока.

Адрес следующего блока	Количество зарезервированных слов	Адрес I-го слова	Заголовок блока
Число неиспользованных разрядов	Число слов		Заголовок записи
Запись с ключом ТЕСТ 37700			
Число неиспользованных разрядов	Число слов		Заголовок записи
Запись с ключом ТЕСТ 38000			
Ключ: выражение эквивалентное	ТЕСТ 38000	Указатель записи	
Ключ: выражение эквивалентное	ТЕСТ 37700	Указатель записи	

Рис. I

Индексные блоки

Каждый раз, когда вводится новая запись в блок данных, необходимо создать и индекс для данной записи. Самый маленький номер ключа в блоке данных и адрес этого блока на диске образуют некоторый указатель и располагаются в области, называемой 0-й индексный блок.

Количество уровней индексных блоков, необходимых для нахождения соответствующей записи данных, определяется величиной индексного блока в соответствии с числом блоков данных в массиве. Можно создавать и многоуровневые индексные блоки. Но во всех случаях самое маленькое значение ключа с каждого 0-го уровня ставится в индексный блок более высокого уровня (в данном случае I-го) и т.д., пока поя-

вится один индексный блок самого высокого уровня, который охватывает все ключи массива. Этот блок на самом высоком уровне называется "первичным индексным блоком".

Индексный блок состоит из группы ключевых записей: а именно, из заголовка, где записан номер уровня и адрес I-го слова зарезервированной области. Все ключевые записи для данного массива имеют одну и ту же величину. В последних 30 разрядах последнего слова ключевой записи записан указатель, который является адресом на диске индексного блока более низкого уровня или блока данных.

Каждая запись дублирует ключ, который является ключом с самым маленьким номером в блоке, на который указывает. Если имеется зарезервированная область, она находится в конце блока.

На рис. 2 показана структура блока.

Места записей в индексных блоках и блоках данных меняются, если это необходимо для соблюдения логической последовательности массива. Если запись указывает логически в наполненный блок данных, записи этого блока расщепляются на два блока данных и соответствующие перемены делаются в индексных блоках.

Таким образом, записи всегда могут обрабатываться посредством поиска в логической последовательности ключей.

Индексный блок тоже расщепляется, когда необходимо вместить запись в уже заполненный блок.

Новый блок будет иметь тот же самый номер уровня, как и прежний. Среднее время обработки записи не увеличивается, пока новый уровень индексов не будет сформирован.

Чтобы стало яснее взаимодействие этих двух видов блоков в системе индексно-последовательного метода, рассмотрим конкретный пример. Предположим, что имеем блок данных, величина которого определена

Номер уровня	I-ое зарезервированное слово	Заголовок
Самый маленький ключ в блоке данных: напр., ТЕСТ 3900	указатель блока инд.данных	Запись ключа
Самый маленький ключ в блоке данных: напр., ТЕСТ 4900	указатель блока инд.данных	Запись ключа
Зарезервированная область		

Рис. 2

так, что могут быть собраны три записи; причем место одной записи зарезервировано на будущее. Кроме того, предположим, что имеем индексный блок величиной в две записи без резервной области.

Заполнение этих блоков записями соответственно с ключами А, В, Е, Т (А, потом В и т.д. А, В, С, Д - по алфавиту) образует массив.

Блок данных I - заголовок
А - запись
В - запись
резерв
В - указатель
А - указатель

Блок данных 2 - заголовок
Е - запись
Т - запись
резерв
Т - указатель
Е - указатель

Индекс 0-го уровня
А - указатель
Е - указатель

Если необходимо ввести запись с ключом Д, то при исследовании информации индексного блока, видно, что ее надо включить в БД-I. Это не вызывает изменения индексного блока, т.к. значение ключа Д больше, чем значение ключа А. В БД-I получается следующая картина:

Блок данных I – заголовок
А – запись
В – запись
Д – запись
В – указатель
А – указатель

Но если сейчас необходимо ввести запись с ключом О, то это приведет к изменению порядка в блоке данных 2, изменения индексного блока не наблюдается.

Блок данных 2 – заголовок
Е – запись
О – запись
Т – запись
Г – указатель
О – указатель
Е – указатель

Пусть необходимо ввести запись с ключом С. Т.к. ее место в блоке I, но там нет пустых мест, и блок расщепляется. Получаются следующие два блока:

Блок данных I – заголовок
А – запись
В – запись
р е з е р в
В – указатель
А – указатель

Блок данных 3 – заголовок
С – запись
Д – запись
р е з е р в
Д – указатель
С – указатель

В заголовке "Блок данных I" будет указан адрес связи с блоком данных 3, который является следующим логическим блоком в последовательности.

Запись ключа для нового блока данных должна появиться на О–ом уровне индексного блока. В данном случае этот индексный блок должен расщепляться на два блока О–го уровня.

Индекс О–го уровня
А – указатель
С – указатель

Индекс О–го уровня
Е – указатель
р е з е р в

В связи с этим расщеплением необходимо создать первичный блок на уровне I. Записи ключей в этом блоке будут указывать на самые маленькие ключи в О–ом уровне.

Индекс I–го уровня
А – указатель
Е – указатель

Получается следующая структура:

Индекс I-го уровня
A - указатель
E - указатель

Индекс 0-го уровня
A - указатель
C - указатель

Индекс 0-го уровня
E - указатель
р е з е р в

Весь этот процесс характеризует основные действия при работе индексно-последовательного метода.

В системе СДС для этих операций существуют стандартные программы. Но этот метод очень трудно используется и является неэффективным при работе ассоциативно-адресной ИПС, т.к. трудно организовать узлы списков. Зато его хорошо можно применять при реализации и обработке последовательных массивов информации.

Гораздо лучше в ИПС со списочной структурой можно использовать метод прямого доступа.

Прямой доступ

В работе /4/ показано, что расположение и организация узловых структур на диске сводятся к эффективной организации массивов с прямым доступом. При этом надо иметь в виду, что это массивы с достаточно большим количеством элементов. В данном случае, когда массив становится настолько большим, что невозможно или по крайней мере неэкономично просматривать каждый элемент массива для проверки его

соответствия, появляется особенно важная проблема - проблема организации массива. Она порождает технические трудности при разработке системы информационного поиска, связанные с согласованием объема памяти, времени, необходимого для получения ответа, и т.д. Поэтому при разработке организационной структуры массива необходимо достичь приемлемого компромисса между:

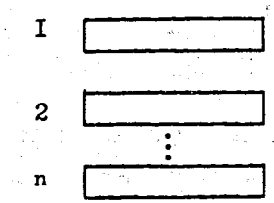
I) временем, затрачиваемым на поиск элемента массива;

2) объемом памяти на диске, необходимым для размещения данного массива.

Эти возможности могут быть реализованы при организации массива методом прямого доступа. Как индексно-последовательный метод, так и метод прямого доступа имеет ряд разновидностей при его реализации на разных вычислительных машинах. Так, например, на ЭВМ серии IBM его можно реализовать как при помощи ключей, так и без них. На ЭВМ системы СДС для реализации этого метода обязательно наличие ключей. В /4/ подробно описана методика реализации прямого доступа с помощью алгоритмического языка КОБОЛ для ЭВМ типа СДС.

Здесь отметим основные черты этой методики и выясним некоторые количественные характеристики, связанные с использованием прямого доступа.

Итак, пусть имеется массив, состоящий из n записей.



Для того, чтобы организовать этот массив как массив с прямым доступом, необходимо каждому элементу присвоить некоторое значение ключа. Так, например, в КОБОЛе СДС для этого используется ACTUAL KEY (фактический ключ), который определяет поле данных, не являющееся частью входной-выходной записи. Это поле содержит относительный номер дорожки (взятый относительно первой дорожки массива) диска. Так, например, если n =7, то, присваивая фактическому ключу (в некоторой произвольной последовательности) числа от I до 7 (например, 3,5,I,4,2,6,7) и используя соответствующие операторы "Писать" и " Читать", можно записать (читать) данный массив на диске. Это будет массив с прямым доступом. Происходит фактически следующий процесс: операционная система ЭВМ СДС составляет в момент формирования массива т.н. "ИНДЕКС" массива. В данном случае этот индекс имеет семь позиций и выглядит следующим образом:

ИНДЕКС

I.	адрес на диске
2.	адрес на диске
3.	адрес на диске
4.	адрес на диске
5.	адрес на диске
6.	адрес на диске
7.	адрес на диске

Это означает, что когда фактический ключ примет, например, значение "3", то сразу при помощи ИНДЕКСа находится соответствующее место записи на диске. Это будет третья запись массива с прямым доступом.

Содержимое этого индекса хранится вместе с данным массивом на диске. (После того как массив оформлен и записан на нем).

Но при достаточно большом массиве становится неэкономично располагать его таким образом и каждый раз читать только одну запись в ОП. Гораздо удобнее читать определенное количество записей с диска в оперативную память и иметь фактически массив, состоящий из "зон", содержащих некоторое количество записей. Эти зоны можно рассматривать как элементы массива с прямым доступом.

Сейчас проведем анализ подобной организации метода прямого доступа.

Итак, под "доступом" к МД понимается единичное " перебрасывание" (чтение или запись) некоторого количества информации; термин "логическая единица информации" употребляется как мера количества той информации, которая может быть сразу прочитана или записана на МД. Конечно, величина логической единицы зависит от данной ЭВМ. Мы уже отметили, что у ЭВМ СДС-6200, например, это фиксированная величина размерностью 64 слова, каждое слово - 60-разрядное. Для ЭВМ типа IBM это переменная величина, и ее значение зависит от конструктора поисковой системы. Для вычислительных машин типа ЕС это тоже переменная величина.

Примем, что время, необходимое для чтения (записи) одной логической величины с МД, - t_s имеет фиксированное значение.

Информация в информационном массиве представляется набором записей (скажем, запись поисковых образов документов (ПОД)), каждая из которых характеризуется некоторым количеством дескрипторов.

Пусть общее число различных дескрипторов, т.е. количество дескрипторов в дескрипторном языке, будет N . Код каждого дескриптора характеризуется некоторым числовым индексом - j , где $1 \leq j \leq N$.

Пусть $f(j)$ – это общее число случаев, когда j -ый дескриптор используется для индексирования в ПОД (для поиска всех ПОД, где он встречается).

Пусть S – это общее число случаев, в которых все дескрипторы встречаются в различных ПОДах данного массива.

Следовательно, $S = \sum_{j=1}^{N} f(j)$.

Пусть $P(j)$ – это вероятность того, что выбранный для индексирования дескриптор есть j -ый дескриптор. Функции $f(j)$ и $P(j)$ могут быть неизвестными или очень трудно определяемыми.

Самые различные связи раскрываются при определении $f(j)$ и $P(j)$. Вид этих функций зависит прежде всего от конкретной информации в массиве, от метода индексирования этой информации и так далее. Могут быть предложены различные распределения для этих функций, но самое удачное, на наш взгляд,– это распределение по закону Ципфа. /7/

Были сделаны эксперименты /I/, которые показывают, что закон Ципфа по отношению к английскому тексту дает $P(j)= 0{,}1/j$. Некоторые исследователи, работающие с дескрипторами в индексных массивах, нашли эмпирическое подтверждение закона Ципфа, но у этих авторов постоянное К не равняется 0,I, т.е. $P(j)=K/j$, вследствие вариации в числе дескрипторов (N). Три случая разного изменения функций $f(j)$ и $P(j)$ рассмотрены в /I/.

I) В этом случае предполагается, что $f(j)$ и $P(j)$ равномерно распределены и постоянны для всех j. Тогда $f(j)= \frac{S}{N}$; $P(j)= \frac{1}{N}$.

2) Если принять, что $\sum_{j=1}^{N} \frac{1}{j} \approx \ell n N + \gamma$; тогда $f(j)=\frac{S}{j(\ell nN+\gamma)}$, где $\gamma \approx 0{,}5772$, и употребляется в качестве константы Эйлера. В этом случае $f(j)$ представляется законом Ципфа, а $P(j)= \frac{1}{N}$ остается постоянной.

3) Третий случай – это когда и $f(j)$, и $P(j)$ имеют распределения по закону Ципфа:

$$f(j) = \frac{S}{j(\ell nN+\gamma)} \quad ; \quad P(j) = \frac{1}{j(\ell nN+\gamma)} .$$

Конечно, возможно и еще одно предположение, когда $f(j)$ является равномерно распределенной и постоянной для всех j, а $P(j)$ распределено по закону Ципфа. Но это маловероятный случай.

Чтобы не обременять читателя математическими выкладками, мы остановимся и будем иллюстрировать дальше наше изложение, принимая, что

$$f(j) = \frac{S}{N} \qquad \text{и} \qquad P(j) = \frac{1}{N} .$$

Сначала дадим математические формулы для определения объема памяти и среднего времени обработки инверсного массива, расположенного на диске.

Термин "зона" будем употреблять в смысле логической записи. В инверсном массиве каждая зона относится к определенному дескриптору, и в ней размещаются записи (ПОД), связанные с этим дескриптором.

Символ C характеризует число записей, которые могут быть размещены в одной логической записи. Зона, связанная с некоторым дескриптором, может содержать больше чем С записей; в этом случае эта зона охватывает больше чем одну логическую запись. (Мы уже отметили, что, в частности для СДС, одна логическая запись состоит из 64 машинных слов).

Пусть как отдельный элемент данной записи на диске рассматривается одно машинное слово. Тогда m таких слов определяет длину одной записи (одного ПОД). Тогда минимальная память, которая необходима для размещения массива на диске,– это $S.m.$

Пусть V_s определяет объем всей памяти на диске, которая исполь-

зуется для данного массива. Тогда P (коэффициент эффективного использования дисковой памяти) можно вычислить по следующей формуле:

$$P = \frac{V_s - S.m}{S.m} . \qquad (I)$$

P=0, когда память не расходуется впустую; P = I, когда используется в два раза больше памяти, чем это необходимо, и т.д.

В инверсной организации важные параметры- это N (число дескрипторов в дескрипторном словаре) и $\frac{S}{C}$ (число логических записей, которые используются при записи массива на диске).

Используя все данные зависимости, можем определить память на диске для инверсного массива, т.е.

$$V_s = m.C\left\{\sum_{j=1}^{N} \frac{f(j)}{C}\right\} , \qquad (2)$$

где m - число машинных слов, необходимых для формирования одной записи. Считаем, что все записи имеют одинаковую длину.

C - число записей, размещенных в одной логической записи.

$\sum_{j=1}^{N} \frac{f(j)}{C}$ - число всех используемых логических записей.

V_s - обозначает объем памяти в машинных словах.

Если поставим (2) в (I), получим P:

$$P = \frac{C}{S}\left\{\sum_{j=1}^{N} \frac{f(j)}{C}\right\} - 1. \qquad (3)$$

Так как выражение $\sum_{j=1}^{N} \frac{f(j)}{C}$ не может быть меньше N, то

$$P \geqslant N\frac{C}{S} - 1. \qquad (4)$$

Среднее время T_r для обработки инверсного массива может быть определено следующей формулой:

$$T_r = t_s\left\{\sum_{j=1}^{N} \left(\frac{f(j)}{C}\right)\right\} \cdot P(j)$$

число логических записей — вероятность

Если $f(j)=\frac{S}{N}$ и $P(j) = \frac{1}{N}$, то

$$\frac{T_r}{t_s} = \sum_{j=1}^{N} \left(\frac{S}{C}\ \frac{1}{N}\right)\frac{1}{N}\ , \qquad (6)$$

$$\frac{1}{N}\frac{S}{C} \leq \frac{T_r}{t_s} < \frac{1}{N}\cdot\frac{S}{C} + 1, \qquad (7)$$

$$P = \frac{C}{S}\left\{\sum_{j=1}^{N} \left(\frac{S}{N}\frac{1}{C}\right)\right\} - 1, \qquad (8)$$

$$0 \leq P < N\frac{C}{S} . \qquad (9)$$

Можно применить данные формулы к узловому методу организации информационного массива /2/. Примем, что каждая логическая запись достаточно велика, чтобы в ней разместилась одна или несколько записей. Каждый дескриптор имеет отдельный список, и каждая запись в этом списке представляется соответствующим узлом. Это означает, что если ПОД индексирован несколькими дескрипторами, то ПОД является узлом цепных списков этих дескрипторов.

В такой списочной структуре имеется f(j) узлов в списке, соответствующем j -му дескриптору; следовательно, j -ый список имеет длину f(j). Время, необходимое для обработки всех узлов (записей) в списке, расположенном на диске, есть $f(j)\cdot t_s$. Это есть и среднее, и максимум необходимое время для обработки всех узлов.

Ясно, что в данном случае имеется в виду, что каждая запись (ПОД) охватывает не больше чем одну логическую запись. В противном

случае больше чем один доступ будет необходим для тех узлов, которые содержатся в больше чем одной логической записи.

Итак, в данном случае:

$$T_r = t_s \sum_{j=1}^{N} f(j)P(j), \qquad (10)$$

$$\frac{T_r}{t_s} = \sum_{j=1}^{N} \frac{S}{N} \cdot \frac{1}{N}\ , \qquad (11)$$

$$\frac{T_r}{t_s} = \frac{S}{N}\ . \qquad (12)$$

Мы обобщили эти выкладки и сделали машинный эксперимент на ЭВМ СДС-6200 для большого информационного массива, состоящего из ПОД. Массив можно расположить больше чем на одном диске. (Основные результаты показаны в /3/ и /4/). Структура массива с прямым доступом следующая

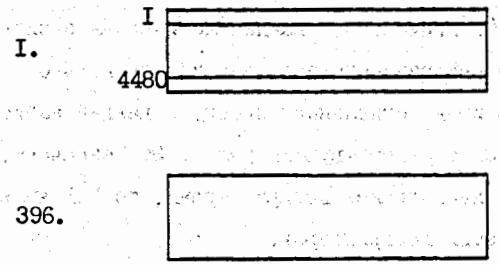


Рис. 3

На диске используются 198 цилиндров (0-ой и 199-ый остаются для служебной информации). Сам массив состоит из 396 зон, где каждая зона представляет величиной 1/2 цилиндра. Каждая зона фактически состоит из 140 логических записей, т.е. для чтения одной зоны нужны $140 t_s$.

В каждой зоне находятся дескрипторные списки. Эти списки состоят из узлов переменной длины. Каждый узел состоит из набора дескрипторов, а на каждый дескриптор уходят два машинных слова. Все списки сегментированы по зонам.

Предельное отношение средней длины списков к общему объему массива, при котором поиск по спискам на МД становится равноценен последовательному поиску на МЛ, можно получить, используя следующее выражение.

$$\text{Предельное отношение} = \frac{\text{время чтения с МЛ}}{\text{среднее время выбора с МД}}\ .$$

Для СДС 6200 $= \frac{10}{75} = 13\%$ (для записи в среднем из 600 символов). Следовательно, число членов в сегментированных списках по отношению к общему объему информационного массива не должно превышать 13%.

Это отношение показывает, что поиск по спискам не может быть оправдан, если время, необходимое на выполнение этого поиска, превышает время, необходимое для осуществления поиска во всем массиве последовательно.

Отсюда можно сделать вывод, что списочная структура выгодна при работе с МД при незначительных размерах сегментированных списков по сравнению с основным информационным массивом.

В принятой нами организации информационного массива на диске каждая зона, к которой обращаемся для ее выбора в ОП, состоит из "времени", $140\ t_s$. Кроме того, любые две зоны с номерами i, $i+1$ расположены на одном цилиндре диска. Но исследования показывают /5/, что для минимизации среднего времени доступа и записи на диске необходимо еще правильно разместить их на цилиндрах этого диска (какие зоны будут находиться в сердце диска, какие на торцах диска).

Чтобы была ясна наша мысль, проведем некоторый анализ расположения отдельных логических записей на диске с целью минимизации

среднего времени доступа к этим записям, зная вероятность обращения к отдельным записям.

В /5/ показано, что запись с самой высокой вероятностью должна быть расположена на "центральном" цилиндре. В нашем примере это цилиндр с № 99. На этом цилиндре будет расположена та зона, вероятность обращения к которой самая большая. Последовательно зоны с меньшей вероятностью будут располагаться по обеим сторонам этого "центрального" цилиндра.

Таким образом, номера зон на рис.3 (от I до 396) не будут соответствовать хронологическому расположению зон на диске. Если в начале работы алгоритма формирования (до заполнения данного диска) I-ая и 2-ая зоны будут расположены на 2-ом цилиндре и т.д., то после накопления статистики о вероятностях обращения к соответствующим зонам будет сделана сортировка так, чтобы зона с самой высокой вероятностью располагалась на "центральном" цилиндре диска.

Сравнительный анализ инверсного и ассоциативно-адресного способа организации массивов

Мы уже показали, что прямой доступ очень удобен для проведения поиска в ИПС. Но особое значение при работе алгоритмов информационно-поисковой системы имеет, как уже отмечалось, правильная организация массива. Основные способы организации поисковых информационных массивов - это последовательный, инверсный и ассоциативно-адресный /9/.

Чтобы дать их сравнительную оценку, описать общие блоки и характерные для каждого способа стороны, введем некоторую "общую структуру массива". Для описания общей структуры необходимо определить некоторые основные понятия.

Запись R является подмножеством $A \times V$, где каждый признак A имеет только одно значение V.

Индекс записи R - это множество тех признаков, которые характеризуют R. Их будем называть дескрипторами. Для описания дескрипторов будем употреблять символ К.

Список (L) записей данного дескриптора К, или проще, К - список,- это есть множество записей, содержащих К таким образом, что:

I. Каждый адрес связи в L дает адрес записи только в L.

2. Имеется единственная запись в L, на которую не указывает ни одна другая запись, содержащая К; она называется первой записью списка.

3. Есть единственная запись в L, содержащая 0-ой адрес связи,- это конец списка (т.е. последняя запись списка).

Множеством F называют массив, если каждый К-список, включающий одну или больше записей, содержится в F.

Рассмотрим массив F, записи которого содержат m разных дескрипторов: $K_1, K_2, \ldots\ldots K_m$.

Пусть n_i - это число записей, содержащих дескриптор K_i, и h_i - число различных списков (K_i - списки) в F.

Отметим через a_{ij} начальный адрес j-ого K_i-списка. Тогда управляющая часть массива F представляет собой множество следующих параметров (K_i, n_i, h_i; a_{i1}, a_{i2}, ... a_{ihi}) для $i = 1,2,\ldots\ldots m$.

Тогда общая структура любого информационного массива представляет собой совокупность его управляющей части и множества конкретных записей.

Инверсный и ассоциативно-адресный массивы являются частными случаями так формулируемой общей структуры массива /IO/. Под "инверсным" массивом мы будем понимать такой массив, где в каждом списке имеется только одна запись, т.е. $n_i = h_i$. Общая схема инверсного массива показана на рис. 4. Везде $h_i = n_i$ (т.е. $h_1 = n_1$; $h_2 = n_2$ и т.д.).

Этот массив можно рассматривать следующим образом: он содержит в своей управляющей части все дескрипторы, к которым присоединены адреса всех записей, содержащих данный дескриптор. Из этого следует, что управляющая часть инверсного массива обычно большая, т.к. для каждого дескриптора имеется связанная с ним последовательность адресов; кроме того, данный адрес записи может встречаться много раз в разных последовательностях. Это видно на рис.4, где показаны записи, содержащие дескриптор K_m. Первая запись, например, содержит дескрипторы K_2 и K_m, и оба адреса одинаковы.

В ассоциативно-адресном массиве последовательности в управляющей части соответствуют отдельным спискам. Для каждого дескриптора имеется только один список, т.е. $h_i = 1$. В управляющей части массива показаны только начальные адреса каждого списка (см. рис.5). Последовательные записи (их адреса) получаются посредством адресов связи, которые являются частью конкретных записей в массиве записей. Очень важно отметить, что в этом случае списки могут пересекаться и иметь одну или больше общих записей. Один и тот же массив, организованный инверсным способом, показан на рис.4, ассоциативно-адресным-на рис. 5.

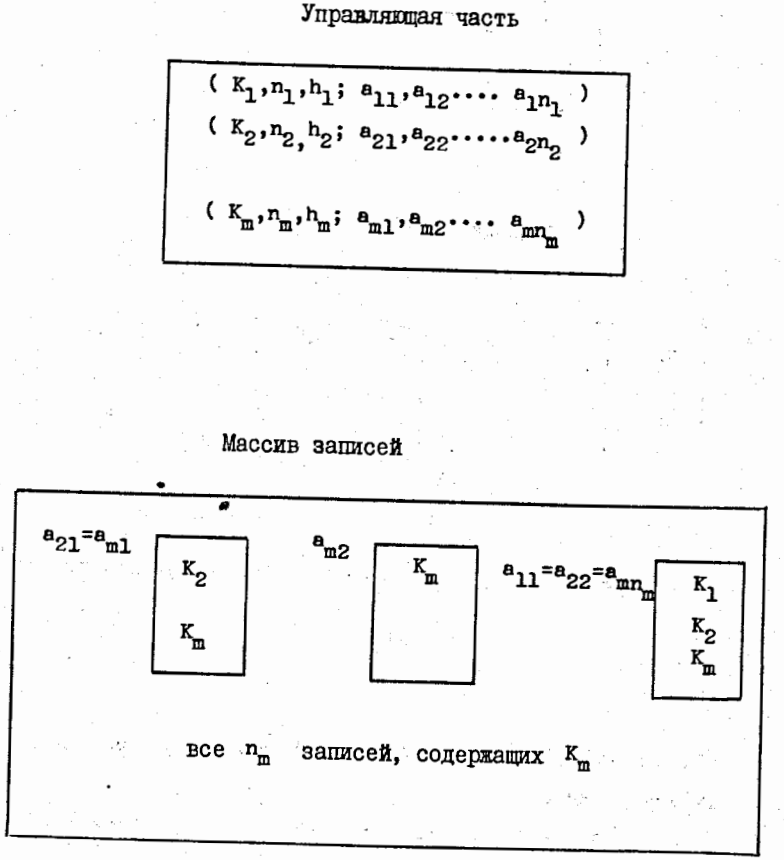
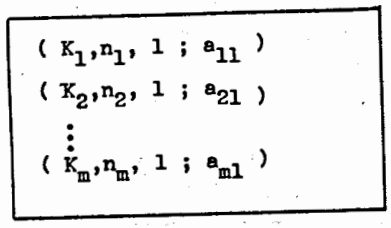


Рис. 4

Управляющая часть



Массив записей

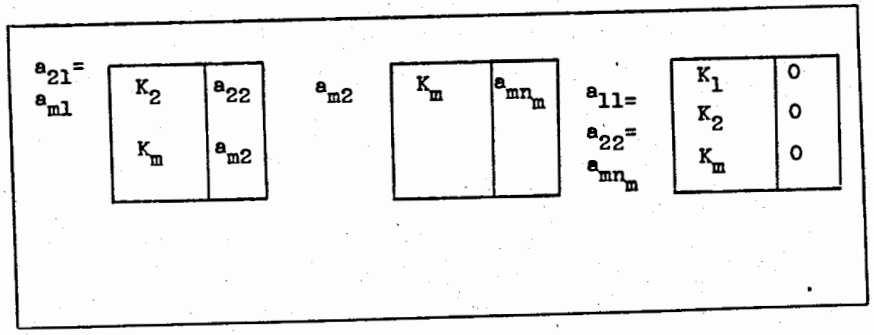


Рис.5

Для простоты будем считать, что запись считается обработанной, если найден ее адрес.

Тогда сама структура массивов определяет два вида поисковой обработки: обработка управляющей части и конкретного массива записей.

Для обработки управляющей части любого массива можно ввести некоторый функционал f, который позволяет "декодировать" начальные адреса различных списков, которые относятся к дескриптору K_i. Так, например, по отношению к инверсному массиву функционал f будет действовать следующим образом (не надо забывать, что в каждом списке инверсного массива с начальным адресом a_{i1}, a_{i2}, ... a_ih_i имеется только одна запись): для определения всех начальных адресов списков функционал f должен быть выполнен ($h_i + 1$) раза, т.е.

$f (K_i, 0) = a_{i1}$ находим первый адрес
$f (K_i, a_{i1}) = a_{i2}$ находим второй адрес
$f (K_i, a_ih_{i-1}) = a_ih_i$ находим h_i адрес
$f (K_i, a_ih_i) = 0$ нет больше адресов.

Для управляющей части ассоциативно-адресного массива необходимо иметь два применения функционала:

$f (K_i, 0) = a_{i1}$ находим начальный адрес
$f (K_i, a_{i1}) = 0$ нет больше списков.

Подобным образом можно ввести некоторый функционал G для обработки уже конкретных списков в массиве записей. Но, как видим, для инверсного массива этот функционал не нужен, так как этот массив фактически обработан после окончания обработки его управляющей части. Там сразу находятся все адреса записей. Необходимо обрабатывать дальше только ассоциативно-адресный массив записей, т.к. после обработки его управляющей части найдены только начальные члены списков.

Если рассмотреть внимательно рис.4,5, увидим, что, принимая некоторую стратегию поиска /2/ (например, для нахождения всех записей, содержащих дескрипторы K_2 и K_m), мы по существу просмотрим

одинаковое количество записей как в инверсном, так и в ассоциативно-адресном массиве. Но в первом случае обработка закончится в управляющей части, а во второй – в массиве записей.

При сравнительно небольших массивах оба способа действуют достаточно эффективно. Но при увеличении числа членов в массиве, становится трудно обрабатывать управляющую часть инверсного массива. Тогда уже видны преимущества ассоциативно-адресной организации, где из-за характера самой структуры поиск имеет минимум два уровня: поиск (обработка) в управляющей части массива и поиск в массиве записей. В некоторых случаях сама управляющая часть может быть разбита на n управляющих частей, а сами списки сегментированы по зонам /2/, и соответственно организован n -уровневый иерархический поиск /2/.

Мы показали, что ассоциативно-адресная организация массива является достаточно эффективной при обработке больших информационных массивов /2/.

ЛИТЕРАТУРА

I. Thomas C.Lowe. The Influence of Data Base Characteristics and Usage on Direct Access File Organization , PJ. of ACM, v15,4,1968 .

2. Д.Д.Арнаудов. Об одном способе организации информационного массива и дескрипторного словаря в библиографической ИПС . Сб. Цифровая вычислительная техника и программирование, вып.7, I970, Сов. радио.

3. Д.Д.Арнаудов. Обработка списочной информации при программировании основных алгоритмов ИПС на КОБОЛе , сообщение ОИЯИ, I0-7586, Дубна, I973.

4. Д.Д.Арнаудов. Организация мультисписочных узловых структур и их программная реализация на КОБОЛе , сообщение ОИЯИ, I0-7587, Дубна, I973.

5. David D. oth. Placement of Records on a Secondary Storage Device to Minimize Access Time ,J.of ACM, v20,3,1973.

6. К.Джермейн. «Программирование на IBM/360 , М, I97I.

7. I.Zipf,G.Ringsley. Human Behavior and the Principle of Least Effort , Addison and Wesley, Cambridge, Mass., 1949.

8. Scope Index Sequential, CDC,USA,1971.

9. P.R.Weinberg. A time sharing chemical Information refrieval system . Ph.D.Diss.,U.of Pennsylvania, Philadelphia,1968.

10. D.Hsiao and oth. A Formal system for Information Refrieval from Files , Comm. of the ACM , vol.13, 2,1970,USA.

Рукопись поступила в издательский отдел
I6 мая I974 года.