

**СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА**

P11-86-326

А.Бабаев*, О.Н.Ломидзе, Г.Л.Мазный

**АЛГОРИТМ СОРТИРОВКИ
С ИСПОЛЬЗОВАНИЕМ ДВУХ УРОВНЕЙ ПАМЯТИ**

* Научно-исследовательский институт прикладной физики
Ташкентского государственного университета

1986

Одним из условий успешного функционирования систем обработки информации является создание эффективного алгоритма сортировки данных. В настоящей работе описывается алгоритм сортировки с использованием нескольких уровней внешней памяти, который был реализован на ЭВМ KRS-4201^{/1/}, а в настоящее время реализован на ЭВМ СМ-4.

Прежде чем перейти к обсуждению предлагаемого алгоритма, приведем некоторые определения.

Единица данных, типично обрабатываемая информационной системой, называется записью^{/2/}. Записи составляют файл. Каждый элемент информации в записи - поле записи. Ключом называется поле, содержащее величину, используемую в правилах упорядочения /сортировки/ записей. Очевидные правила упорядочения - числовые, алфавитные. Ключ может состоять из нескольких полей. В таком составном ключе можно определить несколько уровней ключей, при этом ключ первого уровня называется старшим, все остальные - младшими.

Любая информационная система основывается на определенной модели данных. Большинство современных реализаций применяют иерархическую или сетевую модель^{/3/}. Однако на практике эти модели часто служат для создания простых файлов и реализации поиска по одному или нескольким ключам. При хранении данных записи приходится упорядочивать по значениям ключей. Иногда желательно упорядочение внутри упорядочения, т.е. файл может быть упорядочен по старшему ключу, а внутри этого упорядочения - по младшему. Упорядочение записей - функция программ сортировки.

Существует много различных алгоритмов сортировки и много подходов к их программированию^{/2,4/}. Первое необходимое условие выбора подходящего алгоритма сортировки зависит от характеристик сортируемых данных - количества записей в исходном файле, размера и постоянства длины записей, типа и длины ключей, распределения ключей /диапазона и сгруппированности значений/.

Представленный в данной работе алгоритм разрабатывался для сортировки записей больших размеров, расположенных на магнитной ленте, с использованием буфера на устройствах прямого доступа /диск, барабан/.

Каждая запись имеет фиксированную длину и два ключа - старший и младший. Причем упорядочение необходимо делать по старшему ключу, а внутри этого упорядочения - по младшему. Диапазон возможных ключей задан, однако не все значения из этого диапазона присутствуют в исходном файле. В известных авторам публикациях^{/2,4/} при описании различных алгоритмов сортировки пред-

полагается, что все значения из заданного диапазона ключей равновероятны. Возможности вычислительной машины также влияют на принятие решения о сортировке, поэтому второе условие, которое учитывалось разработчиками, - минимальный комплект внешних устройств /например, на СМ-4 в распоряжении авторов имелось всего 2 накопителя на магнитных лентах/. С учетом реальных условий эксплуатации при разработке алгоритма стояла задача свести число просмотров входной ленты к минимуму. Для решения этой задачи авторам потребовалось разработать оригинальный алгоритм, используя тот факт, что конкретная обрабатываемая информация представляет собой разреженный массив данных. На промежуточных этапах сортировки в качестве буфера используются файлы на внешних устройствах прямого доступа /диск, барабан/.

В рамках указанных выше условий разработанный алгоритм сортировки представляется достаточно универсальным. Однако заметим, что производительность программы сортировки всегда очень зависит от ее конкретной реализации. "Различие в производительности двух разных алгоритмов может быть в несколько раз меньше, чем между "хорошо" и "плохо", запрограммированными одним и тем же алгоритмом"/2/. Предлагаемый алгоритм был реализован авторами на ЭВМ СМ-4 и KRS-4201.

Специфику каждой из двух реализованных версий данного алгоритма, определяемую особенностями конкретной ЭВМ, в приведенном ниже общем описании процедуры сортировки детально рассматривать здесь не будем. Блок-схема разработанного алгоритма изображена на рис.1 и 2.

Цель программы - создать упорядоченный файл записей на выходной МЛ. Как уже упоминалось, сначала записи сортируются по старшему ключу, а затем внутри каждой группы записей с одинаковым значением старшего ключа они сортируются по младшему ключу. Причем младший ключ может неоднозначно идентифицировать запись, т.е. могут встретиться несколько записей с одинаковым значением младшего ключевого поля.

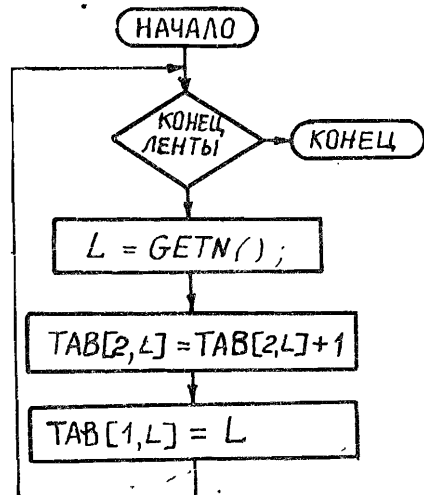


Рис.1. Нулевой просмотр.

На рис.1 представлена блок-схема нулевого просмотра, функционально отличного от всех последующих просмотров ленты. Функция $L = GETN()$ /см.рис.1/ читает со входной МЛ одну запись и равна значению старшего ключа.

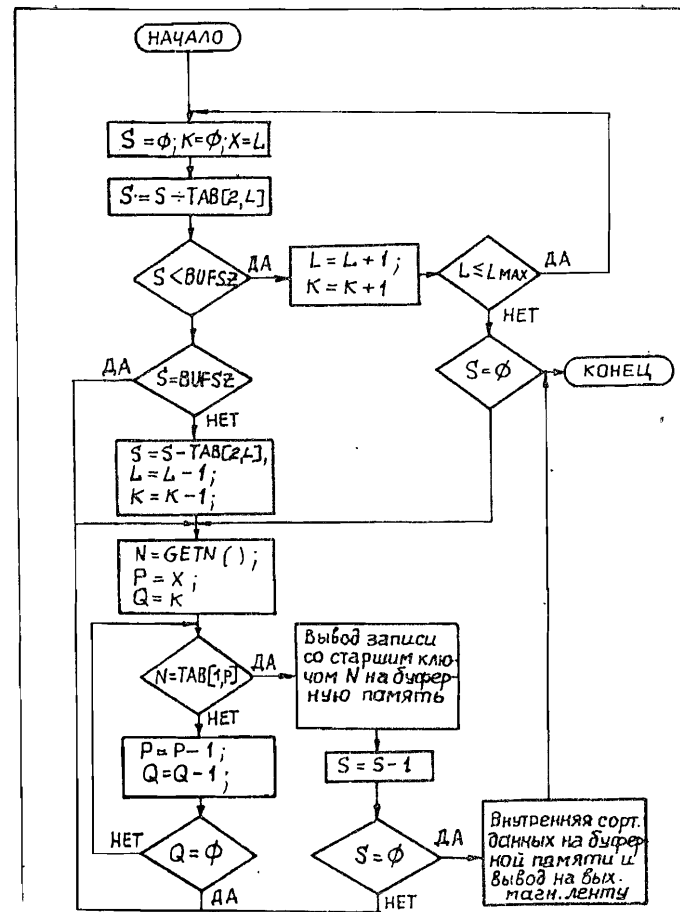


Рис.2. J-й просмотр.

С использованием прочитанной информации в оперативной памяти составляется таблица:

N	M
1	300
2	500
⋮	⋮
300	700
⋮	⋮
⋮	⋮

где N - старший ключ, M - ключ всех записей, определяемых значением старшего ключа N.

Составление таблицы происходит с одновременной сортировкой ее строк по значению старшего ключа. При этом алгоритм внутренней сортировки выбирается с учетом конкретных особенно-

стей ЭВМ, на которой реализуется программа. В процессе нулевого просмотра мы получаем информацию о минимальном значении старшего ключа, а также о количестве записей с одинаковым значением старших ключей.

Алгоритм всех последующих просмотров одинаков.

За время каждого j -го $/j = 1, 2, \dots, K/$ просмотра обрабатывается порция групп записей, помещающаяся во внешнем буфере на ВУ прямого доступа. Число групп записей с одинаковым старшим ключом в каждой j -й порции определяется соотношением: $S \leq BUFSZ$, где S - количество всех записей в выбранной порции обрабатываемой информации; $BUFSZ$ - размер буфера /кратный длине одной записи/ на ВУ прямого доступа. Очевидно, что чем большего размера дисковый буфер вы берете, тем быстрее пройдет сортировка. С другой стороны, минимальный размер буфера должен быть таков, чтобы в нем размещалась группа записей с одинаковым ключом, максимальная по количеству записей. Затем со входной МЛ читаются все записи, старшие ключи которых входят в j -ю порцию, и размещаются в буфер на ВУ в соответствии с функцией расстановки.

В результате в буфере ВУ мы имеем отсортированную по старшим ключам часть входной информации. Затем каждая группа сортируется в оперативной памяти и записывается на выходную МЛ; процесс повторяется.

Число этих просмотров не превышает K ,

$$K = \left[\frac{СК}{NB} \right] + L, \quad //$$

где K - число просмотров ленты; $СК$ - количество старших ключей; NB - минимальное количество старших ключей, по которым можно произвести сортировку за один просмотр ленты; при этом $L = 0$, если делится нацело на NB , а $L = 1$ в противном случае.

Авторы благодарят Н.Н.Говоруна за поддержку работы и И.Н.Силина за полезные рекомендации на этапе разработки алгоритма.

ЛИТЕРАТУРА

1. Мазный Г.Л. ОИЯИ, 11-80-134, Дубна, 1980.
2. Лорин Г. Сортировка и система сортировки. "Наука", М., 1985.
3. Атре Т. Структурный подход к организации баз данных. "Финансы и статистика", М., 1983.
4. Кнут Д. Искусство программирования, том 3. "Мир", М., 1978.

Рукопись поступила в издательский отдел
22 мая 1986 года.

Бабаев А., Ломидзе О.Н., Мазный Г.Л.
Алгоритм сортировки с использованием двух
уровней памяти

P11-86-326

Описан алгоритм сортировки больших массивов данных, реализованный с использованием двух уровней внешней памяти: магнитных лент и внешней памяти прямого доступа. Алгоритм реализован в мобильной операционной системе "ДЕМОС" на ЭВМ СМ-4, и на ЭВМ KRS-4201.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1986

Перевод авторов

Babaev A., Lomidze O.N., Mazny G.L.
Sort algorithm for Data with Using Two Levels
of Memory

P11-86-326

Sort program for data with using two levels of memory is described. Algorithm is realised in mobile operating system "DEMOS" for computers CM-4 and KRS-4201.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1986