IN-PLACE INDEX COMPRESSION

The present invention relates to methods of upkeep of hierarchical indices in and by data processing systems with particular reference to the macro operations of compression, insertion and deletion.

BACKGROUND

It is normal for a data processing system to support many data entities or files called for by the potential user by identity whereas the data processing system needs to know the address, usually in peripheral storage, at which the data entity or at least the start of the data entity is stored. Since the system allocates storage, the system also maintains an index in which it records the addresses at which it has stored data entities in association with the identities of those data items. In other words, an index is a machine maintained mapping of data identity onto data location. The main function performed in relation to an index is searching and to speed searching, hierarchical indices are normal in systems of any appreciable capacity. In a hierarchical index, there are plural sequential levels of mapping extending from an entry level or apex to an exit level or base. Each level maps onto the next sequential level so that the extent of the next level requiring processing during a search is limited.

However, alteration of a hierarchical index is much more time consuming than is alteration of the equivalent single level index since, again in the worst case, an alteration, always at the base level, may propagate "upwards" through all the levels up to and including the apex. Conventionally, the index is not available for its main function—searching—while it is being updated and its integrity is not maintained, if it is not maintained updated. Further, updating by insertion and deletion tends to create a most irregular index structure, and it is the function of the upkeep macro operation—compression—to restore, periodically, the regularity of the index structure. Compression is the most time consuming of the upkeep operations. For examples of conventional hierarchical index upkeep operations, please refer to:

U.S. Pat. No. 3,643,226
U.S. Pat. No. 3,611,316
U.S. Pat. No. 3,916,316
U.S. Pat. No. 3,651,483
UK Pat. No. 1,336,817

Ideally we would like to provide a hierarchical index which can be searched while it is being updated. Although we have not succeeded in achieving this end in absolute terms, we have devised an index which can

(a) be searched at virtually any point during compression of the index,

(b) be updated by insertion or deletion at regularly ocurring break points during compression of the index and

(c) accommodate compression restart at the point at which it was interrupted.

It is pointed out that an approach to these aims can be achieved by duplicating the index in such a way that one copy of the index is available for searching, insertion and deletion; while the other copy is being compressed, but this arrangement, apart from absorbing storage space, implies that neither copy is fully maintained.

In the case of the present invention, there is no duplication and the single index is, in terms, fully maintained.

SUMMARY OF THE INVENTION

From one aspect the present invention provides a method of maintaining said multi-level index by said system comprising the steps of executing a sequence of processing cycles wherein each cycle progresses through the index levels in a first and then a second direction, selectively performing at each level of the index, while performing a processing cycle in said first direction, a first subset of basic operation iterations, including basic operations for duplicating the presence ot an original parameter, and selectively performing at each level of the index, while performing a processing cycle in said second direction, a second subset of basic operation iterations including basic operations for deleting the original presence of said duplicated parameter.

From another aspect, the present monitor provides a method of compressing said index by said system, by the execution of a variable length ordered sequence of processing cycles, each said cycle comprising a variable length sequence of basic operation iterations, each said iteration comprising the selective performance of basic operations of a first subset and a non-overlapping second subset of said basic operations of an ordered fixed sequence of said basic operations, wherein each said full cycle progresses through all said index levels, in a first direction, performing said first subset of basic operation iterations and subsequently, in a second direction, performing a said second subset of said basic operation iterations, basic operations in said first subset duplicating the presence of an original parameter, basic operations in the said second subset deleting the original presence of the said duplicated parameter.

If in addition, in the compression method, the permitted iterations are defined by an action vector table, each vector having a vector element corresponding to each basic operation, arranged in sequence order, effective in relation to the contents of the moveable operating window in the index spanning three logically adjacent blocks at one index level together with the blocks containing the parent entries of the aforesaid three blocks, the current action vector selection being determined by the contents of the current window, it is possible to resume compression automatically from the point at which it was interrupted.

As far as interruption of compression for insertion or deletion is concerned, this has to be restricted to those points in the compression process at which the processing is at rest at the base level between cycles of compression.

A preferred embodiment of the present invention will now be described twice, firstly in general functional terms with reference to the accompanying drawings and, thereafter, insofar as the main compression operation is concerned, in more detailed and specific terms.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings,

FIG. 1 is a diagram of an index block;

FIG. 2 is a diagram of a very simple dislocated index;

FIGS. 3 to 6 are sequential diagrams illustrating the insertion of entries into an index;

FIGS. 7 and 8 are sequential diagrams illustrating the deletion of entries from an index;