BCL MOLECULAR -- ALPHA-MATCHING

Alpha Matching is a file indexing techinique for the fast access of records matching a given set of "keys". The index differs from conventional Indexed Sequential by having more than one key field.

If the file contains product records, the key fields will be derived from the description in a carefully defined manner. For example (Pharmaceutical Wholesaling) four keys may be chosen:

Key-A	First 4 characters of Description
Pack Size	Numeric Descriptor (eg number of tablets in one bottle)
Presentation	3 character descriptor (TABlet, LIQuid)
Key-B	4 character descriptor (eg tablet weight)

To access product records by Alpha-Matching only one key, namely Key-A, is compulsory. The VDU operator will be prompted (by the program) for all four keys, but may "pass over" the other three fields. The program will display all products matching the given keys (one line per product, pausing if the screen becomes full until the operator requests a continuation). Each line is numbered to allow selection by line number. In addition to the line number and product description, other information such as unit price and current free stock units may be displayed.

Provided response time is adequate (ie the first product appearing on the screen within one second) and provided certain additional support features are built-in to the system (eg alternative names for certain products; automatic suggestion of up to six alternative products if out-of-stock), alpha-matching is ideally suited to real-time telephone ordering: given a degree of familiarity with her employer's business the VDU operator may rappidly translate the customer's order for "one by twelve Amyl Nitrite 0.3ml capsules" into the key fields (Key-A=AMYL, Pack Size=12, Presentation=CAP, if uncertain of Key-B this would be passed over). The sequence of prompting by the program is as spoken by the above customer: Quantity, Pack Size, Key-A, Key-B, Presentation. A refinement is that if the customer quotes a non-existing pack size the program automatically drops this field and re-searches for other pack sizes, if there is still no match the program assumes a spelling mistake has been made and reprompts from Key-A.

Implementation aims to minimise the total number of disc/core transfers required to perform an alpha match.

Thus when a new data record is loaded, the load program extracts the key fields and uses them to create an "index" record which is placed at the end of an "index file" (the data record itself is loaded into a vacant slot in a Direct Access data-set; it may be subsequently retrieved by its slot number completely independently of the alpha-matching system). The new index record is then linked (by record pointers) into its correct position in the logical key sequence (eg Key-B within Presentation within Pack Size within Key-A). Each index record also contains its data record slot number.

Since index records are relatively small (typically 8 words long) it is possible to bring several index records into a core buffer with a single disc read. An index reorganisation utility is therefore provided, which ensures that the physical order of the index records corresponds to the logical key sequence. This reorgansiation is carried out at intervals, eg after 200 or so index insertions/deletions (to reorganise 10,000 records takes about 2 minutes).

The chain of, say, 10,000 index records in logical sequence is divided into 4,096 sub-chains. Each sub-chain contains all index records with the same first two characters in Key-A. Thus the index file is itself indexed by a "primary" index of 4,096 records, each primary index record pointing to the first main index record in its sub-chain (many of the sub-chains will be empty, one or two may be quite long, eg 200 entries).

Assuming the index to be properly reorganised, the number of disc transfers required will be determined largely by the size of the main index core buffer. A 1K word buffer (holding 128 records) is normally used. When given a key string, the access software computes the primary record required by means of a simple algorithm, obtains from it the first main index record in the sub-chain (=1 disc transfer), searches the chain (=1, sometimes 2 disc transfers) and retrieves the first data-record (=1 disc transfer). With an average random access time for D1600 drives equivalent to 20 transfers per second, it is obvious that the entire operation may be completed with apparent alacrity. It is worth noting that the demonstration software uses a smaller main index buffer (holding 32 records) and its longest chain (beginning "BE") contains about 150 entries.

In a multi-programming environment it is clearly undesirable to have more than one program (task) searching the main index at the same time (all tasks use the same transfer buffer). Requests for searches are therefore queued and handled one at a time. Of course, once a task has found its first alpha match, a second task may perform a search whilst the first is accessing and displaying its data record. The first task may then resume its search for further matches from the point in the main index that it had previously reached. This queuing mechanism allows index maintenance operations (loading, deleting) to be carried out without restrictions (only reorganisation requires a "dedicated" machine).

J. Templeman. (Leicester)