normally being possible by means of standard software. Control of the operation can be achieved by using a record of information loss output from updating runs to determine selection of records.

The total system is summarised in **Fig. 2,** which shows particularly the dual function of this 'unactioned movements' file. Data cycled in this way will, of course, not necessarily all be movements unactioned because of hardware failure, but those that are can be designated in a particular way and picked out by the reconstitution program. When records are blocked on the main file, one record being unreadable must also imply that those blocked with it are inaccessible, and account must be taken of the fact. Depending on the data organisation, it may in fact be easier to reconstitute a larger area still, possibly a cylinder, because data originally in a home area has been transferred to an overflow area or vice versa. Since all the up-date history has to be read in any event, this will make little difference to the time required for the operation.

General points

If the main file falls easily into logical sections that can be dumped separately on a cyclical basis, the work load can be spread more evenly, dumping, say, a quarter of the file every week. But it brings its attendant problems if used in conjunction with the above method. Not only will a complete updating record have to be maintained as far back as the oldest dump quarter that is current, but also updating information selected during reconstitution only if it is successive to the dump from which a particular record was replaced. Possibly the updating record could be purged of all information that has become redundant, whenever a section of the main file is dumped, but this seems unnecessarily involved for the small advantage it procures. It is preferable in terms of simplicity and machine usage to dump the complete file and start the updating record from scratch every time.

No mention has been made in the preceding descriptions of reporting procedures, which must be regarded as essential. Full details of data loss not only aid decisions about when to reconstitute, but also enable action to be taken outside the main system if delay is critical. With indexed files, it should be possible to ascertain the precise extent of data loss due to hardware failure, but if the manufacturer's software regards indexes as a private affair between itself and the hardware, a copy in more accessible form can always be maintained elsewhere. This is, in any case, advisable, since loss of part or all of the index can render large sections or the whole of a file immediately inaccessible.

Beyond these large-scale security arrangements, more controls will be needed within programs that update direct access files, to guard against operation errors. The consequence of operators starting a run with old input data, for example, can be to necessitate reconstitution of large sections of a file if not prevented by programmed controls. For instance, a dummy account could be held on the main file, and updated at the beginning of each run, containing details of input files previously processed and general information about the state of the system. By reference to this, attempts to input apparently out-of-date information could be queried and prevented before permanent damage was inflicted on the main file. The possibility of this happening, or of hardware or software proving fallible can be ignored if it does not seem to justify the extra systems and programming effort required, but it should be recognised as a dangerous gamble under the present state of development of manufacturers' operating systems.

Reference

FRASER, A. G. (1969). Integrity of a mass storage filing system, *The Computer Journal*, Vol. 12, No. 1, pp. 1–5.

Book review

Topics in Interval Analysis, by E. R. Hansen (editor), 1969; 130 pages. (*Oxford University Press*, £2.50)

This book is an account of lectures given by invited speakers at a symposium on interval analysis sponsored by the Oxford University Computing Laboratory in early 1968. The book is divided into two distinct parts.

Part 1 consists of a description of interval analysis used to obtain error bounds for computed solutions to standard algebraic problems such as the solution of linear and non-linear equations and the inversion of matrices. A description is also given of Triplex-Algol, a formalized language specially devised to cope with interval analysis algorithms. The contributors are R. E. Moore, K. Nickel, E. Hansen and J. Meinguet.

Part 2 deals with interval analysis applied to continuous problems. These include numerical integration, the numerical solution of two point boundary value problems, initial value problems for systems of ordinary differential equations, and partial differential equations. There is also a section devoted to statistical distributions of errors applied to linear programming. The authors here consist of R. E. Moore, E. Hansen, F. Krückeberg and M. Dempster.

Considering the complexity of the subject, the book is particularly easy to read. This has been achieved by the individual authors concentrating on simple examples to illustrate the methods for bounding errors in the various computed solutions. The conclusion to be drawn from this book is that interval analysis has met with considerable success in the analysis of errors for the numerical solution of algebraic prolems. The same cannot be said, however, with regard to continuous problems. The material is particularly thin with regard to differential equations and it is difficult to see how interval analysis can have much impact on the numerical solution of partial differential equations in the foreseeable future.

The book is highly recommended and the editor has done a good job in producing a review of recent progress in the fascinating subject of interval analysis.

A. R. MITCHELL (Dundee)

Downloaded from https://academic.oup.com/comjnl/article/13/2/126/349188 by guest on 19 April 2024