the time taken by the Roll-In user), indicates to the operator that the Roll-In is now complete, and then gives control back to MSOS which continues processing the interrupted program at the point at which it left off. Should the user program fail to return control to the RIOS, an emergency auto-load feature in the form of a special auto-load card, exists whereby the operator may reload memory and continue the interrupted program as if a normal return from Roll-In had occurred.

System usage

It was anticipated when the system was designed that system usage would vary considerably according to the requirements of the individual groups using it. This has been found true in practice and it is unusual to find two remote stations requesting the machine at the same time. When this does happen, however, the remote station operating on the interrupt line having higher priority is allowed to access the central computer first.

The actual time involved in a complete Roll-In operation depends, of course, on the user program. Usually the total time involved has been of the order of 10 seconds of which 6 seconds is overhead time: core swapping, loading, checking and accounting.

In the case of the group doing analysis of ionospheric soundings, Roll-In requests have usually been made over a period of an hour or two at intervals of about one minute. It has been found that this does not seriously disrupt normal batch processing operations.

Conclusions

The Roll-In System illustrates what can be done in the field of remote access with a machine not originally designed to provide this facility. No figures are available to indicate the cost of the hardware changes, as much of the equipment was built within the Establishment. The design and implementation of the software, however, took about three man months.

The system obviously has its limitations but, as a short term solution, is very effective.

Book review

Automatic Programming, Vol. 6, part 2, by P. J. Brown and G. F. Colouris, 1969; 67 pages. (*Pergamon Press Ltd.*, £1.75)

For Volume 6 of the Annual Review in Automatic Programming, the second to appear since its renaissance under a new Editorial Board, the publishers have adopted a new format, and the volume appears in four separate paperback parts (though a hard cover edition will be available later). This practice will commend itself to those who might balk at paying £7 for a complete volume when only one of the papers is of interest to them.

Most of this part-volume is devoted to 'A Survey of Macro Processors' by P. J. Brown. This is a masterly survey, comprehensive and clearly written. The author first defines a macro processor in a general way as 'a piece of software designed to allow the user to add new facilities of his own design to an existing piece of software', and discusses the main application areas of such systems. He then considers in detail the problems that face the designer of a macro processor, highlighting the difficulties and giving illustrations of the solutions that have been produced by various designers. He succeeds in fitting a number of diverse systems into a general framework that permits meaningful comparisons and gives a clear account of two problems which many designers of macro systems do not even consider, the question of 'call-by-name' versus 'call-by-value' and the question of multi-level calls, (that is, can a macro call be built up of separate pieces of text?). Lest the reader should think that this latter question is merely an academic fancy, the author shows how it has important applications in code optimisation. The problems of implementation are discussed, and the article concludes by returning to the application areas outlined at the start, and drawing conclusions as to the applicability of macro processors in each area. It is rounded off with a comprehensive and cross-indexed set of references.

The remainder of the volume is taken up with a short (15 page) article by G. F. Colouris, entitled 'A Machine Independent Assembly Language For Systems Programs'. The article gives a short account of SICTRAN, a language in which 'in order to achieve machine independence and yet retain the control of program and data organisation required, parts of the run-time structure of the program and its data are left undefined by the translator'. The description is brief, and illustrated by few examples. Possibly the examples are not typical, but they give the impression that one could achieve the same effect by writing a few macro definitions for one of Dr. Brown's macro processors.

The meat of this volume is undoubtably the survey of macro processors. This should be compulsory reading for anyone concerned with the design of such systems, and can be read with profit by anyone interested in computer software. I will certainly make sure that all my students master its contents.

D. W. BARRON (Southampton)

Vol. 6 part 3 appears on page 170