above. Job processing, in relation to the computer configuration being simulated, terminates as soon as one of the jobs completes execution. This constraint was imposed to preserve the basic assumption of a fixed number of jobs being active throughout the simulation.

The simulator was designed to permit the setting of values to three parameters. Firstly, the number of I/O channels available; secondly, a specified value for R; and thirdly, the number of jobs to be multiprogrammed. The latter two parameters were incorporated into the simulator to run over the range $0{\cdot}1 \leqslant R \leqslant 1{\cdot}8$ and the number of jobs multiprogrammed from 3 to 10. The model also assumed that a change in R, resulting directly from a change in buffer size, could be accommodated by the configuration, i.e. a transfer of a variable size block could be initiated.

4. Results

Three series of simulation runs were carried out, for one, two and three independent I/O channels. Within each series the number of jobs multiprogrammed was varied from 3 to 10. The percentage of CPU usage derived from these series is shown in Table **1.** Table **2** shows the percentage reduction in elapsed times for the same set of jobs.

Table 1 supports the view that CPU usage increases as R increases, i.e. as I/O decreases through increase in buffer size. Table 1 also indicates a relatively greater increase in CPU usage when R lies in the area of small values ($R < 1{\cdot}0$) and within this range, a larger increase in, for example, the interval [0·2, 0·4]. It is also interesting to note that in the case of one channel it is always better to double R rather than increase the number of jobs multiprogrammed. The same trend is observed in the case of 2 and 3 channels but here only for the low values of R.

A significant point is that for one I/O channel and $R < 0{\cdot}6$, CPU utilisation is approximately independent of the degree of multiprogramming. Further, for a higher degree of multiprogramming, i.e. 7 to 10 and a given number of I/O channels, CPU usage approximately independent of multiprogramming over the whole range of R obtained.

Table 2 indicates that there is an improvement in the elapsed time of jobs in all cases though again greater benefits are observed when R is less than 1·0 and with one I/O channel.

5. Conclusions

The results suggest that an increase in R (by reducing I/O time) results in a significant reduction in the elapsed time for the jobs executed. An important result, as Landy (1971) has pointed out, is the danger of overdoing multiprogramming to the point where such times are longer than the mean free time between machine failures.

In almost all cases, the results indicate that increasing R gives a larger gain in CPU utilisation than increasing the degree of multiprogramming.

R can easily be measured for every job, using the system accounts. Hence, in a given installation, R can be measured and its distribution constructed (empirically). It can then be ascertained whether a majority of the jobs have a low R value (< 1) and these figures used in conjunction with the average degree of multiprogramming to determine whether the buffer sizes used by the system and the user (on average) are the most suitable.

References

ARDEN, B. W., GALLER, B. A., O'BRIEN, T. C., and WESTERVELT, F. H. (1966). Program and addressing structure in a time-sharing environment, *JACM*, Vol. 13, no. 1, pp. 1-16.

CRITCHLOW, A. J. (1963). Generalised multi-processing and multi-programming systems. *Proc. AFIPS Full Joint Comput. Conf.*, Vol. 24, pp. 107-125.

LANDY, B. (1971). Development of Scheduling Strategies, *Software Practice and Experience*, Vol. 1, pp. 279-295.

MEALY, G. H., WITT, B. I., and CLARK, W. A. (1966). The functional structure of OS/360, *IBM Syst. J.*, Vol. 5, pp. 3-51.

Book reviews

Software Reliability, Vols 1 and 2, 1977; 288 and 410 pages. (*Infotech International*, £72)

This book takes the usual form for Infotech State of the Art Reports of one volume of invited papers and a companion volume on the analysis and bibliography. Whether or not this is the most convenient way of publishing the proceedings of such conferences is debatable, and certainly the reader needs to make cross references between the two volumes.

The volume on the analysis is divided into the general area of software development, software validation and software fault tolerance. The sources for this material go beyond the invited papers and include past publications, both books and papers. Although this volume touches on a wide range of problems and contains several valuable contributions to the whole subject of software reliability, it is difficult to avoid the feeling that a shorter survey, setting out clearly the many useful points elucidated in this book would perhaps be more useful for the educational world.

The volume of invited papers gives a good coverage of this field, but as with other Infotech publications the high price puts it beyond the range of many users and not a few university libraries.

M. H. ROGERS (Bristol)

Fundamentals of Data Communications, by Jerry FitzGerald and Tom S. Eason, 1978; 260 pages. (*John Wiley*, £9·40)

A well structured first text book on Data Communications practice, covering transmission techniques, hardware, network configurations, error handling, software concepts and system design. Ten basic steps of the systems approach are presented and form the basis for the final chapter on designing communications networks for which the previous chapters have laid the foundations. Each chapter is complemented by a set of questions enabling the reader to verify his understanding of technical terms and concepts—a good idea for a subject full of 'much abused' jargon.

My overall impression (despite the purely American emphasis in the chapter on common carriers and network design) is that any reader experienced in central computer hardware and software will gather enough background from the book to participate at all stages of the data communications system design process—with suitable devotion to cost-effectiveness, which is well treated in the text.

Recommended as an introduction to the subject for students, and their teachers, and for DP managers and systems analysts getting involved for the first time.

R. C. WITTS (London)