The EMAS 2900 Operating System*

P. D. Stephens, Edinburgh Regional Computing Centre

Summary

In the summer of 1975 Edinburgh University took delivery of its first ICL 2900 series computer. By the summer of 1980 Edinburgh was operating two large-scale 2900 machines providing a popular and heavily used multi-access service. This service does not use any of the manufacturer's software, but rather a high-performance multi-access operating system written by the university.

In this personal account, one of the implementors reflects on the triumphs and disappointments of an eventful quinquennium.

The Original EMAS Operating System

The Edinburgh Regional Computing Centre (ERCC) was set up as a result of the recommendations of the Flowers report of 1964.[1] The ERCC was intended to become 'a centre of excellence in multi-access computing'. Little guidance was given on how this desirable state of affairs was to be achieved, but since the Centre was to be equipped with an English Electric Leo Marconi 4-75 computer for which there was no multi-access (MAC) software, some sort of do-it-yourself scheme was necessary. We learnt the hard way from our mistakes. The first effort, involving a joint university/manufacturer implementation team, ended in the summer of 1970; this produced mistakes in plenty, but no software. The Centre picked itself off the floor, licked its wounds, offered an interim batch service on the manufacturer's J-level operating system, and tried again. This second attempt was purely a university effort, involving the Centre and the Department of Computer Science - the latter providing most of the effort and the expertise. This attempt was less ambitious in scale, and aimed to find simple solutions to the problems that slowed the first attempt almost to a stop. In one year a self-supporting system was operational, and in the autumn of 1972 EMAS became the main service vehicle on the 4-75 computer.

The service was very popular, and was capable of supporting 35-40 terminals plus some batch work. By 1974 the system was overloaded and a second 4-75 computer was acquired; it too was soon overloaded. EMAS has been fully described elsewhere,[2] and no further details will be given here. However it is relevant to list the main features which gives the system its distinctive flavour:

(a) *Interactive working.* The system is normally accessed from an interactive terminal, which it uses as its major source of control information. This may be one of a large number of devices attached via a geographically distributed network.

(b) *Multiple virtual memories.* Each user is allocated a 16 megabyte virtual address space.

(c) *Mapped files.* Files are not accessed via a procedural interface (read record, write record, etcetera), but rather by being associated with a range of virtual addresses and accessed as if they were an extension of a program's memory space.

(d) *Controlled sharing of information.* The system organizes sharing of programs and data without imposing any restrictions or red tape on the user.

(e) *Transparent memory hierarchy.* The system operates a three-level memory hierarchy, but the user is only aware of files and virtual addresses.

(f) *Minimal user constraints.* The system attempts to restrain the user as little as possible in his use of files, languages, virtual addresses, etcetera. There is a set of facilities provided by a standard subsystem, but the user may ignore it and provide his own subsystem if he so wishes.

(g) *Minimal information loss through crashes.* If the hardware fails an attempt is made to minimize the resultant information loss.

(h) *No system degradation under load.* The system cannot allocate more resources than it possesses, but it should allocate as much as possible to the users at all times. A novel approach to resource control precludes the possibility of 'thrashing' under load.

(i) *Repeatability and enforced fairness.* The resources used by a job should be dependent on its requirements, and not on other demands on the system; therefore, if a job is run again it should use the same resources. A user should get a fair share of the system, determined by his own requirements and by the current number of users of the system.

Performance Measurement

Although responsibility for EMAS development passed from the Department of Computer Science to ERCC with the opening of the service, the transfer did not mark the end of academic interest in the system. A number of research projects were started during which user behaviour, supervisor overheads, paging characteristics, and response times were subject to minute scrutiny. As part of one project, the Edinburgh Remote Terminal Emulator (ERTE) was constructed.[3] ERTE is a very sophisticated

* A more complete version of this paper has been submitted for publication in *Software Practice and Experience* by P. D. Stephens *et al.* (see Reference 4).

emulator that can simulate from 1 to 64 simultaneously operating terminals, each with a different script of commands as well as with different user characteristics (such as think times and typing speeds). ERTE was used in the construction of an interactive benchmark for the proposed Glasgow University 1906S Computer; this Benchmark was based on observed user behaviour from performance measurements on EMAS, and was to be re-used to benchmark the early Computer Board funded ICL 2900 computers. Combining the emulation and response measurements of ERTE with both the precisely repeatable workload of the Glasgow Benchmark, and the supervisor overhead measurements of EMAS, produced a development tool of immense potential. It became possible to subject rival strategies of scheduling to competitive measurement. For several years, the early hours of the morning would find EMAS and ERTE locked in near mortal combat, while the afternoons would find researchers and system designers puzzling over the results, and planning the next experiments. The outcome of this effort was both humbling and rewarding: humbling because it revealed how little system staff knew about what went on in a (relatively simple-minded) MAC System like EMAS; rewarding because it eventually became possible to isolate the few factors that were crucial to system performance from the many which might have been expected to be important but which were not.

The Advent of the 2900 Computer

While EMAS was providing interactive computing in Edinburgh, the batch processing computational facilities were provided on a rented IBM 370/155 computer. This machine served all three universities of the Regional Computing Organisation (that is, the Universities of Edinburgh, Glasgow and Strathclyde). To satisfy an increasing demand for computing, the RCO drew up a requirement for a machine to provide twice the batch throughput of an IBM 370/155 system together with twice the interactive power of EMAS on a 4-75 computer. ICL proposed a 2980 computer running under the VME/B operating system to meet this requirement, the machine to arrive in 1975 and to pass the combined batch and interactive Glasgow Benchmark by June 1977. The interactive component of the benchmark was based on ERTE emulating the terminals, and was demanding: not only were the MAC facilities required considerably in excess of those provided generally in 1975, but the user wait and delay times in the scripts were based on real measurements. The Benchmark was to be completed in 2 hours, but the interactive part was such that it could not be completed in less than 1 hour 20 minutes even on an infinitely powerful machine. To pass the combined batch and interactive Benchmark would require a system that could give a consistently good interactive response, while devoting half of its capacity to batch processing. Apart from demanding a very good MAC response, the benchmark was not very severe. Based on a simple ERCC test, the 370/155 system had a power of about 2.75 Atlas Units, and a 4-75 computer 1.5 Atlas Units. Allowing 10% system overhead on the batch machine, and 33% on the MAC machine, the benchmark required about 7 Atlas Units of power to be delivered to the user. The 2980 was intended to be 20 times as powerful as Atlas (although the ERCC test rates it between 12 and 13 Atlas Units). Thus the 2980 computer could spend nearly half its power in system overhead, and still have enough left to run the user work.

The 2980 system was eventually delivered, and for 2 years it provided a batch service while ICL project teams wrestled with the Benchmark. Basically, VME/B provided a reasonable batch service, but did not get very near the MAC response required. When the time for the benchmark ran out, ICL announced that it was in default and proposed some reparations. After complex negotiations, Edinburgh emerged with a 2970 computer in lieu of its share of the 64 MAC terminals which the 2980 was not providing. The only way to service more than 20 terminals on a one-megabyte 2970 computer would be to move the EMAS operating system from the 4-75 environment.

Moving an Operating System

The university had no intention of moving EMAS to the 2900 range until a considerable amount of experience of VME/B had indicated that its MAC facilities would not be satisfactory. In the meantime, the only relevant work carried out was to write a compiler for the EMAS implementation language, IMP. It was not until October 1976 that a small team of people (two from ERCC, two from the Department of Computer Science together with some part-time helpers) set out to produce a new operating system which was scheduled to provide a service from October 1978. By using the same implementation language, and by keeping many internal interfaces intact, it was hoped that much software - particularly superstructure - could be moved, and that the new system would retain all the desirable characteristics of EMAS.

In spite of the small size of the team, it immediately contrived to divide itself into two camps. The cautious considered that producing a mainframe operating system with a handful of staff was such a large undertaking that it was impossible to change the product in any way. The bold took the view that the team was so late in starting (five years behind the manufacturers) that producing a satisfactory system would not be enough. To wean users off VME/B would require a first class MAC performance together with a wider range of facilities than provided by the original. All the suggestions of the performance measurers had to be incorporated, and a few other long-discussed improvements as well. A compromise was reached, rather nearer the bold than the cautious approach.

The actual implementation was carried out entirely from EMAS terminals: code was entered, edited, compiled, linked, and written to magnetic tape. It was then taken to the 2970 computer where it was loaded by an initial program load procedure and tested. When the new system (called EMAS 2900) could support a terminal and an elementary file system, development work gradually moved from the 4-75 to the 2970 system. Progress speeded up as the work left the overloaded System 4-75, and the target date for service of October 1978 was met.

EMAS 2900 Performance

The 2970 system running EMAS did much more than provide the 20 or more terminals stipulated in the reparation agreement. In spite of its minimal main store, it could support over 40 terminals at the level of response demanded by the Glasgow Benchmark. Many features have survived the move from the System 4-75. The back-up and archive store, the student-proof management features, and the high mean time between software failures, are all reproduced in the 2900 version. The similar user interface and compatible compilers assist in transferring work. Among the relatively few new features is an on-line hardware error logging system; this permits an engineer at a terminal to have up-to-date summaries of hardware performance, together with the ability to extract details of specific failures within the previous six weeks.

A full report on the performance of EMAS 2900 has been submitted for publication elsewhere.[4] Tables 1 and 2 summarise ERTE tests described in that report.

Even with the optimistic total of 48 MAC users on the one megabyte 2970 computer, it still manages to service almost half the editing requests in less than two seconds. The system overhead is high, but the overhead in EMAS is directly related to paging, and with this number of users the paging rate is at the maximum which the disc and drum peripheral systems can support. If the main store were to be increased to, say, three megabytes, the system overhead would drop and the response would improve.

The two System 4-75s which have borne the brunt of the Edinburgh MAC service for eight years have reached the end of their useful lives. This summer both these machines and the 2970 will be replaced by an 8 Megabyte dual processor 2972 system. This will run EMAS 2900, and is expected to support between 120 and 150 simultaneous terminals.

2970 Benchmarking Using Glasgow Scripts 1 Megabyte 2970 with 2 FHD-6 paging devices					
MAC Users	Average Editing Response	% of Edits < 2 secs	CPU Utilisation		
			MAC	System	Idle
16	0.5 Secs	84%	37.7%	22.5%	39.8%
32	1.4 Secs	65%	56.5%	42.0%	1.5%
48	2.5 Secs	47%	51.7%	43.2%	5.1%

Table 1

2980 Benchmarking Using Glasgow Scripts 2.5 Megabyte 2980 with 3 FHD-6 paging devices							
MAC Users	Batch Streams	Average Editing Response	% of Edits < 2 Secs	CPU Utilisation			
				Batch	MAC	System	Idle
48	8	0.7 Secs	97%	29.3%	48.7%	22.0%	0.0%
64	8	0.9 Secs	92%	21.9%	52.7%	25.4%	0.0%

Table 2

Developments on the 2980 Computer

As soon as EMAS 2900 was in regular service, the 2980 computer was booked to run some tests. As it happened the New Year weekend was allocated, and in spite of Hogmanay hangovers, a set of figures of which Table 2 is a summary was acquired. This shows EMAS providing a superb MAC service for 64 users - much better, in fact, than required by the 2980 acceptance Benchmark. On the other hand, too little time was spent in batch processing mode to pass that part of the Benchmark: nearer 40% would have been needed, but there was plenty of scope for improving the batch mode CPU utilisation by degrading the MAC response, if required. The team failed to establish the maximum MAC load with no batch work that weekend - ERTE could not simulate enough terminals - but response was very acceptable at 120 users, the maximum number tried.

Armed with these figures reports were made to the other universities in the RCO to the effect that at last the 2980 system could provide the combined MAC/Batch load specified in the original Operational Requirement. This news was received in deafening silence: our colleagues in the other universities showed no inclination to take EMAS seriously. It had been produced too quickly by two few people - real operating systems needed hundreds of man years of effort. Joint management committees spawned working parties and a very fair report. No minds were changed by the study - two universities wanted to retain VME/B, so VME/B was to remain the operating regime for the foreseeable future.

A year later an economic blizzard had hit all universities and the 2980 service was causing concern. Although the 2980 computer ran for two shifts (16 hours per day) the billed time remained in the region of 2-3 hours per day, of which at least half was occupied by Edinburgh users squeezed off EMAS by overloading. This serious underuse was likely to prejudice attempts by any of the three universities to upgrade their local facilities; further, there was the prospect of a financial deficit to be faced. Edinburgh University, which would have to find the money, therefore reopened the question of operating systems on the 2980 computer with a little gentlemanly pressure: Edinburgh would of course run the VME/B system if the other universities wished it, but would expect each of them to pay for five hours of time per day as long as the service was in deficit; otherwise Edinburgh would run EMAS 2900 in the expectation that it would rapidly attract a sufficient number of users to become an asset to the Region. Belief in the superiority of VME/B, although vocal, did not stretch to the point of parting with money!

The 2980 service switched to EMAS from 1 January 1980; there was, indeed, no difficulty in attracting users - by mid-April the service was extended to three shifts. As might be expected, Edinburgh users led the rush, but it is pleasing to note that all three universities made more use of the 2980 computer under EMAS than they did under VME/B. The University of Kent also chose 1 January 1980 as the date on which to switch its 2960 computer from VME/K to EMAS. This wider acceptance was very gratifying to the designers; for years the manufacturers, and others, had dismissed EMAS as a one-off system for a freak community of users.

Reasons for EMAS Performance

The resource control, allocation and scheduling procedures used in EMAS differ radically from those used in other virtual memory systems (VME/B, TSS, etcetera). This section examines some of those procedures in a non-technical manner. Readers who require more technical information should consult Whitfield and Wight,[2] Shelness *et al.*,[5] or Stephens *et al.*[4] These ideas are not all original – indeed some of the most important were published in the mid 1960s;[6] however, all have been investigated over a period of time, and contribute to the performance, simplicity or robustness of EMAS.

(a) *Local control.* Most supervisory functions of EMAS are contained in a module called the Local Controller which has an incarnation for each user. It is concerned with the paging, interrupts and error conditions for that user alone; consequently, EMAS may be considered more accurately to be a collection of uni-programming systems that co-operate, rather than a multi-programming system. This results in simplicity: uni-programming systems are easier to write than multi-programming systems. It also contributes to reliability – any hardware failure is usually confined to one Local Controller and user. The Local Controller may crash, logging off the user, but other users are normally not affected.

(b) *Virtual file store.* EMAS has a virtual file store: files become part of the virtual memory and then are accessed as if they were memory. This contributes to simplicity in two ways: the paging software serves also as record access software, and any arrangement for program sharing will also apply to data sharing. There is also a substantial contribution to performance, since the intervention of system software is only needed when the file is entered into the virtual memory (or connected) and when it is no longer required. Connection involves no transfers – just one table entry for each 256 kbytes of file.

(c) *Drum loading.* In a time-sharing MAC system, the use made of the drum (or other specialised paging device) is likely to have a profound effect on response. The drum contents must vary as the demands on the system change – on a minute-by-minute basis. Certainly, any system that allocates information to the drum on the basis of criteria such as size or nomination by the Operations Supervisor, is not likely to perform well in MAC mode. The ideal situation is for the drum to contain, at any instant, all the pages that will be referenced in the next *n* seconds, *n* being chosen so as to keep the drum completely full. This solution requires the software to have second sight. EMAS uses a similar technique based on the *last n* seconds, with some restrictions for greedy users.

A further important consideration with the drum is that these fast devices are usually near the limits of technology, and thus have a higher failure rate than other rotating devices. To counter this, EMAS uses the drum as a cache – copies of the information are also held on slower devices, and both copies are updated when the page is written to. If a drum fails, no information is lost and the system continues to run, although with a degraded response to interactive users.

(d) *Programs and data sharing.* EMAS arranges to share programs and data whenever more than one user accesses the same entity. This sharing requires no special action by the Operations Supervisor; the owner's data will remain private unless and until he issues the necessary access permissions. It does impose some restrictions on the compiler writer: self modifying code is banned. The performance gains that can be obtained from sharing are very large.

During the benchmark work summarised in Table 2, approximately 45% of all possible page fetches were null operations, as the page requested was already in main store being used by someone else. The saving in drum store is even larger: if all the benchmark users had their own copy of shared material, the amount of drum space required on the system would treble! One further attraction is that it increases as the load on the system increases. Sharing gives EMAS it biggest gains when the system is struggling with what London Transport would call a 'crush loading situation'.

(e) *Memory management.* One of the many revelations to come from the prolonged ERTE measurements was the uselessness of a user defined 'store quota'. The fluctuations in demand for store on multi-access systems are extreme, and simple rigid quotas make poor use of store and result in sluggish response. In EMAS the Local Controller is continually computing the program's 'working set' – this is the minimum number of store pages which the program requires to run efficiently. Before a program is paged in, there will be sufficient pages free to accommodate the latest estimate of the working set size. After a short period of time – usually 1-2 seconds – its working set will be recomputed, and it will be paged out unless no other program is ready to run. In order to minimize pressure on the peripheral systems, programs with large working sets stay in store longer than programs with small working sets; to offset this, small programs are selected much more frequently. It is a debatable decision whether to pre-page the program's working set, or to allow each page to be loaded into main store once it has been referenced.

Whether pre-paging or demand-paging, user processes are all moved in to and out of store by the paging mechanism. EMAS has rejected the use of a supplementary roll-in and roll-out procedure as unlikely to assist, for the following reasons:

(1) It complicates the software.

(2) ERTE has revealed that the important performance parameter is the average elapsed time taken to produce a demand page. Roll-out, by blocking the channel with a large continuous transfer, delays the arrival of demand pages.

(3) A typical user at the start of paging (or rolling) out has 40% of his pages shared; these can be left. On average he will have updated one third of his unshared pages, say 20%; the other 40% are already on backing store. Paging out the user will transfer only 20% of his pages. To obtain a rapid subsequent roll-in, a roll-out system would have to transfer a further 40% of non-updated pages, and possibly some of the shared pages as well.

Conclusion

The principal differences between EMAS and other systems have been outlined. Many universities have bought MAC systems, and not a few have been disappointed with the results. Part of this disappointment can be traced to a system choice based too much on facilities provided, and too little on the fundamentals of system construction. If good MAC performance is required, the system fundamentals must be adequate; piling attractive facilities on the top of unsound foundations is likely to give a sluggish and unsatisfactory service.

References

1. 'A Report of a Joint Working Group on Computers for Research', HMSO Cmnd 2883 (January 1966).
2. H. Whitfield and A. S. Wight, 'EMAS - The Edinburgh Multi-Access System', *Computer J.*, **18** (1973), 331-46.
3. J. C. Adams, W. S. Currie and B. A. C. Gilmore, 'The Structure and Uses of the Edinburgh Remote Emulator', *Software Practice and Experience*', **8** (1978), 451-9.
4. P. D. Stephens, J. K. Yarwood, D. J. Rees, and N. H. Shelness, 'The Evolution of the Operating System EMAS 2900', *Software Practice and Experience*, (submitted for publication).
5. N. H. Shelness, P. D. Stephens, and H. Whitfield, 'The Edinburgh Multi-Access System, Scheduling and Allocation Procedures in the Resident Supervisor', *RAIRO (Informatic, Computer Science) B*, **3** (1975), 29-45.
6. B. W. Arden, T. C. O'Brien and F. H. Westervelt, 'Program and addressing structure in a time-sharing environment', *J. A.C.M.*, **13** (1966), 1-16.