

Communications Control in the Operating System EMAS 2900

William A. Laing

Edinburgh Regional Computing Centre and
Department of Computer Science
University of Edinburgh

James Clerk Maxwell Building
The King's Buildings
Mayfield Road
EDINBURGH
EH9 3JZ

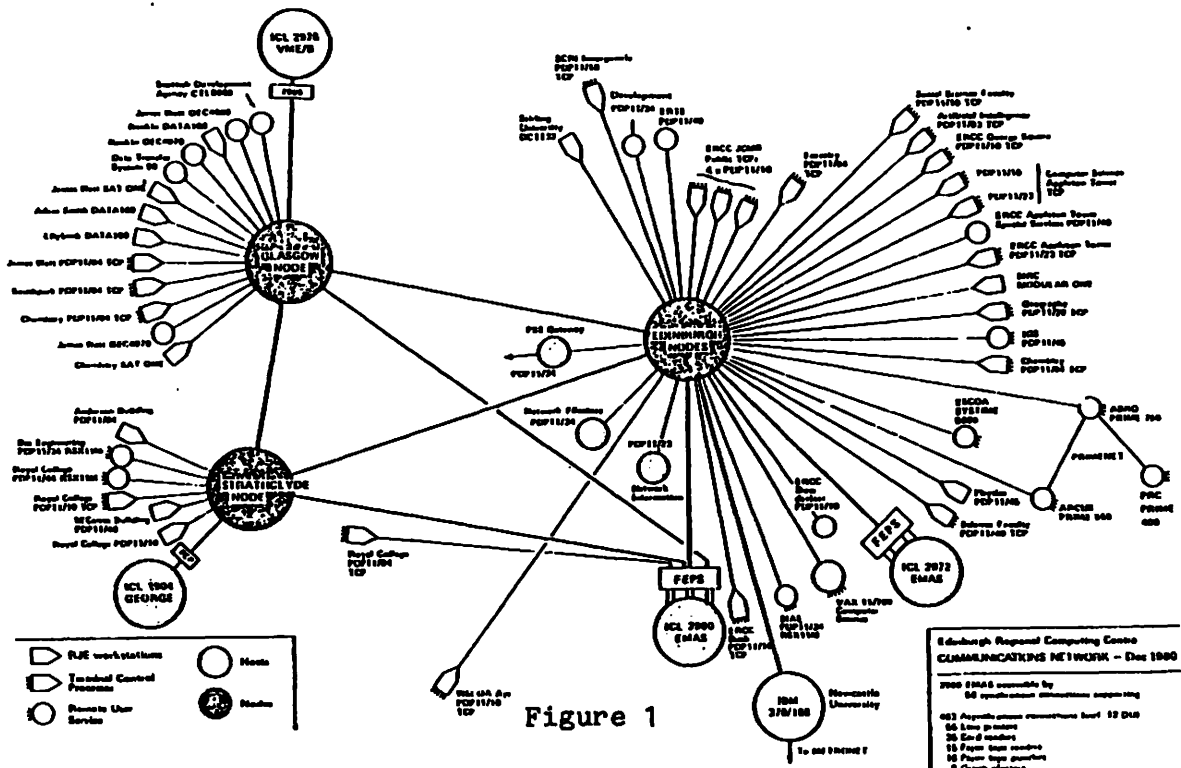
Abstract

A method for organising communications in a virtual memory operating system is described. Its novel features include the use of paging and virtual memory to avoid the need for fixed resident buffers, and the presentation of a single communications interface and data transport mechanism to other parts of the system. It efficiently supports large numbers of simultaneously active high speed interactive terminals, other hosts, and remote job entry terminals on a network.

Introduction

EMAS 2900 is a general-purpose time-sharing virtual memory operating system for the ICL 2900 series of computers. It is a re-implementation of the EMAS system [WH73] which ran on the ICL 4-75 computer, but with a number of aspects redesigned to take account of insights and experience [ST80]. The most extensive changes were made in the handling of communications within the system; these changes are the subject of this paper.

As the uses and range of communications changed substantially during the lifetime of the EMAS system, the effect these changes were having on the performance of the system became significant. The initial EMAS design had a hardware multiplexor which necessitated fixed resident buffers and a buffer management system. Later the multiplexor was replaced by a Front End Processor but the software design remained unchanged. It also became clear that communications protocols and standards were changing, thus a decision was made to make the EMAS 2900 software protocol independent by confining the network dependent parts of the communications system to the Front End Processor. The initial EMAS design worked well when all interactive terminals were operating at ten characters per second and there were only a few remote job entry terminals. What had not been foreseen was the great increase in terminal speed and hence I/O in a terminal session or the increase in remote job entry terminals with fast printers and local disks. A further surprise was the use of EMAS as a secure file store: programmers using machines in other locations kept their programs and data on EMAS and transmitted them daily. The amount of paging traffic and central processor time on EMAS devoted to trivial buffer filling and emptying rapidly became insufferable. At peak times up to 25% of the central processor time and 40% of the paging was going on communications. An indication of the present communications environment is shown in Figure 1. A re-think was necessary and the elegant solution adopted is described below.



Before going into the details of the communications system a little digression is required to describe three features which enabled the incorporation of the novel virtual buffering technique used. First, the address translation facility of the ICL 2900 Peripheral Controllers [BU78] requires segment and page tables when starting an I/O transfer. These controllers then perform address translation with full protection just as the central processor does, although they do not handle page faults which must be generated by software. The second feature concerns the software structure of the EMAS 2900 kernel [RE81]. An important principle of EMAS was that of local decision making; however, the implementation of local decision making was intermingled with the rest of the resident kernel. In EMAS 2900, on the other hand, a very clear separation has been made between the controllers of local functions and controllers of global functions. In principle, each process contains a local controller whose function is to control those and only those resources which have been allocated to that process by the global scheduling controller. This approach can be extended to handle communications: just as a local controller manages the paging of the working set of a user process, so a modified local controller - the Communications Controller - can manage the paging of all virtual buffers involved in transfers. The globally controlled data sharing of EMAS 2900 then operates on such buffer pages without any amendment. Finally, in EMAS and EMAS 2900 all file access is through the virtual memory. When a file is to be accessed, either a code file or a data file, it is first mapped into the virtual memory and then all references to that file are by virtual addresses. Note that the mapping operation does not involve copying the file, only the updating of some tables. Sharing of files is then automatic and transparent as long as access to the file has been permitted in a relevant mode.

EMAS 2900 Communications Design Aims

All communications traffic (i.e. interactive, inter-machine, remote job entry and transaction) was to be transported between a Front End Processor and the mainframe using a common logical interface, independent of the make, model or number of Front End Processors, or of the physical connections by which they were attached to the mainframe or of the network to which they themselves were attached.

Within EMAS 2900 there was to exist a single communications software interface. This requirement was made in order to avoid the use of different internal transport mechanisms (as had happened on EMAS) for each type of communications traffic, and so minimize the amount of software required. This does not imply a single user interface for all communications traffic.

Communications traffic was to be transferred to and from the user's virtual memory directly, not via resident buffer pools. This facilitated a single globally controlled mechanism for moving data within the system, i.e. paging.

The communications software was to be able to handle very high data rates without degrading the rest of the system significantly. To make the handling of communications efficient in a general-purpose time-shared virtual memory operating system there are two principal requirements:

- * The effects of the inherent data unit mismatch between variable length messages, which are the unit of transfer between Front End Processors and the mainframe, and pages, which are the unit of storage allocation and transfer within the mainframe, should be minimized.
- * A paged virtual process should be invoked to handle communications traffic only when enough information has been transferred to trigger computational activity in the virtual process; it should not be invoked for book-keeping operations. This means that an interactive process should be invoked only upon the reception of the complete input message for which it has been waiting, but not upon the reception of message fragments. Likewise, it implies that the Spooling Manager should be invoked upon the completion of a file transfer rather than upon the transfer of a block.

These were the aims, which were implemented within the context of the ICL 2900 machine environment, the local and global control architecture of the EMAS 2900 kernel, and the mapped file system. This was to give flexibility in the future and reduce the demands on the system, which had occurred in the previous implementation, due to communications.

Approach

The critical new feature in EMAS 2900 is that communications traffic is transferred directly between non-paged devices (Front End Processors, remote processors, card readers, line printers, etc.) and files, rather than between these devices and real memory buffers. Each active communications stream therefore has associated with it a file which is mapped into the Communication Controller's paged virtual memory.

In the case of interactive traffic, the files involved are also mapped into the appropriate user's paged virtual memory. Input is written by a Front End Processor into a file and is read from that file by programs running in the associated user's process. In a similar fashion, output is written within a user's process to a file and is read from this file by the associated Front End Processor. Synchronisation is achieved by passing control information between the reader and the writer.

In the case of remote job entry, inter-machine and local non-paged device traffic, a sequence of Spool files is associated with each communications stream by the System's Spooling Manager. It is not necessary for the Spooling Manager to have output files mapped into its virtual memory only input files which must be scanned for control information, e.g. job control language statements.

For I/O transfers to take place to or from a file, the relevant pages of that file must be resident in main memory. This does not imply that these pages should be permanently resident, only that they should be resident for the duration of the data transfer. As the time required

for a data transfer is small and the period between transfers is large compared to page transfer times, the pages containing or receiving the data are transferred to main memory from secondary (drum or disk) storage only as required.

Just as a Local Controller determines the subset of pages in the user's virtual address space which is resident, the Communications Controller determines the subset of pages which is resident from the entire set of communications files. The Communications Controller is given an allocation of pages by the Global Controller which it must not exceed. This allocation must be large enough to satisfy the maximum number of simultaneous peripheral transfers required and to minimize the delay between transfers. The Communications Controller - like the Local Controller - does not control the actual transfer of pages into and out of main memory; this is under the control of the Global Controller. As the Global Controller manages main memory in a way which minimizes transfers, in a large number of cases a request for a page by the Communications Controller will result in the page being found to be already in main memory, either through the page being shared (as for interactive buffer files) or through recapture, i.e. the page is requested before the physical page frame has been re-used.

In most existing computer systems, the inclusion of a Communications Controller for this purpose would be difficult, as such systems possess only a single paging manager to control the use of all memory resources. It is not a problem in EMAS 2900, in which multiple paging managers (the Local Controllers) are provided on a one-to-one basis for each virtual process in the system. The Communications Controller is merely a special form of Local Controller. A special form is required because ICL 2900 peripheral devices do not generate page fault interrupts on missing pages. The Communications Controller must generate these page faults by software and await the arrival of the pages before allowing the transfer to start.

In addition to data transfers to or from devices, there is a requirement for the ability to pass control information between Device Servers and the user, the user being either the Spooling Manager process (for non-interactive work) or the Director, a procedural interface which provides facilities to manipulate files and virtual memory, associated with a particular paged user process (for interactive work). A Device Server is the module that provides the communications interface to the user. The identity and format of this control information need only be agreed between a user and a Device Server. Control information is therefore transparent to the Communications Controller, and the introduction of a new interface only requires the introduction of a new Device Server - no other changes to the communications system are required.

A consequence of this approach is that one of the tasks which the Front End Processor must carry out is to transform the EMAS 2900 communications interface into the attached communications network's interface. Thus if the system is to be attached to a new network the only re-programming required is in the Front End Processor. Similarly, the mainframe may be attached to more than one network simultaneously, each using different protocols, without any special action being taken in the mainframe.

The communications software structure resulting from this approach is shown diagrammatically in Figure 2. A summary of the user interface and the overall structure of EMAS 2900 can be found in the EMAS 2900 Concepts manual [ER80].

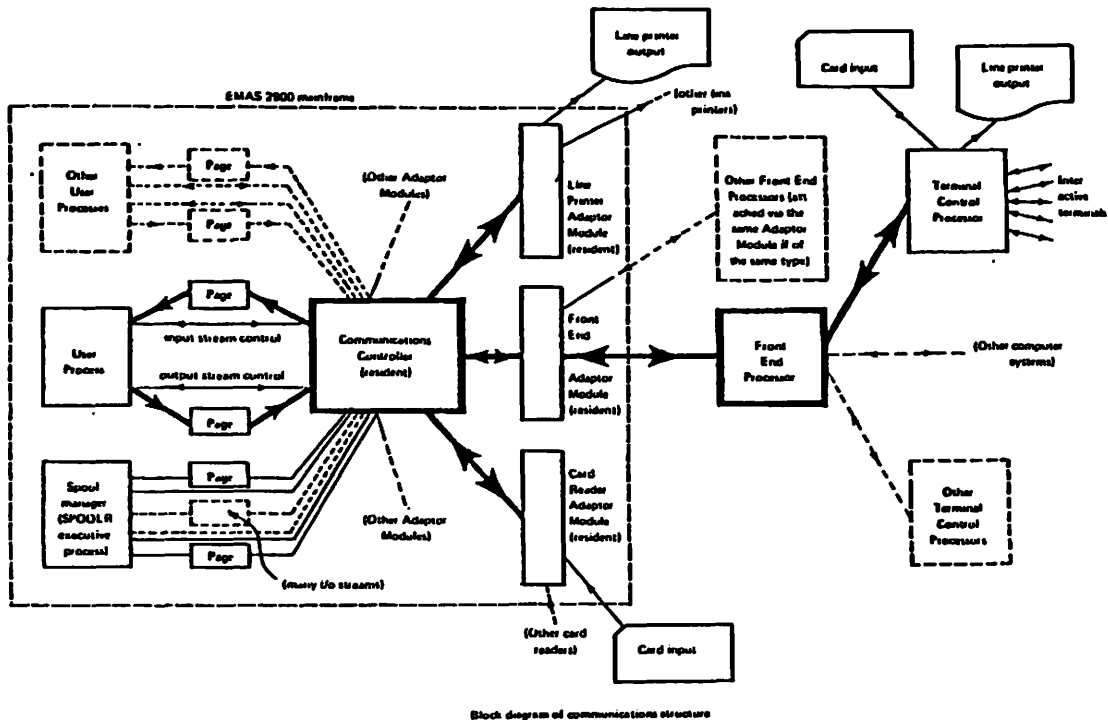


Figure 2

Software Architecture

The preceding section described the approach adopted for the handling of communications traffic within EMAS 2900. In this section the software modules required to implement this design within the mainframe are examined.

- Adaptor Modules

These modules handle control and data transfers between Front End Processors or local slow peripherals (operator consoles, card readers, line printers, etc.) and communications streams. The type of Adaptor Module depends upon the identity of the physical connection employed to attach each Front End Processor or local slow peripheral to the mainframe.

Device Servers

These modules provide the communications interface to users. For local peripherals, such as operator consoles, line printers and card readers, the Device Servers in each case are embedded in the Adaptor Module for that device. In the case of devices connected through a Front End Processor, such as interactive or remote job entry terminals, the Device Servers reside in the Front End Processor itself. Currently two types of interface are supported: interactive and file transfer.

The Communications Controller

This module handles communications streams and deals with page faults on behalf of Adaptor Modules. There is only one Communications Controller in the system.

Each communications stream is identified by a sixteen-bit stream address, with even addresses identifying input streams and odd addresses output streams. A Front End Processor will usually have a number of streams assigned to it, while a line printer, for example, has only one. All streams operate in simplex mode with data transfers taking place in only one direction, though control information may pass in both directions. If a duplex stream is required, as for example with interactive traffic, it is constructed from two streams. Stream clusters required to support remote job entry terminals may be similarly configured from multiple streams.

Interactive Interface Modules

These modules are part of the Director of each virtual process. They support the system's interactive communications interface by fielding users' interactive I/O commands, and generating the appropriate calls on the Communications Controller.

In order to initiate interactive terminal I/O, each process must map into its virtual memory one file for input and one file for output. The Communications Controller autonomously places input data in the input file and updates a register, visible to the user program, each time a complete message has been transferred. When the user program has read all such data and requires more, a Director procedure is called which causes a 'prompt' message to be dispatched for display at the terminal, if no data has been input in the meantime. The process is then suspended until an input message is available. Data to be sent to the terminal is written into the output file and a Director procedure is called to specify the extent of the data. Unless the output buffer is then full, processing may continue immediately. The interactive terminal input and output streams therefore form a full duplex pair. In addition to them, a control stream is available to the user by which interrupt-type messages may be sent independently from the terminal to the process.

The Spooling Manager

This is an executive virtual process that accepts output files from user processes, and generates the appropriate calls on the Communications Controller to transfer the contents to a local or remote line printer, card punch, paper tape punch or other output device. The Spooling Manager also generates new files into which jobs or files from local or remote devices (card readers, paper tape readers, etc.) are read under the control of the Communications Controller, before being transferred to the user process in which they are to be stored or executed.

Performance

On a dual processor configuration with 8 megabytes of main memory and 3 Front End Processors supporting over 100 interactive users, and over 20 remote hosts and remote job entry terminals it is only necessary to allocate 8 pages of main memory to the Communications Controller. In the EMAS system about 4% of the central processor time was spent in the kernel handling communications. In EMAS 2900 this figure has grown to about 7%. This was expected as more of the communications load is being handled by the resident kernel. More significantly the previous figures of 25% of the central processor time of the paged processes and 40% of the paging has been reduced to about 10% of the processor time and 7% of the paging. This has been a significant improvement especially as the communications work load has increased significantly.

Conclusions

The virtual buffering technique used in EMAS 2900 for communicating with interactive and remote job entry terminals has been very successful, and terminals of all speeds have been attached. A buffer of 4Kb is sufficient for slow and medium speed terminals although the software will accept any size up to 128Kb. Using the maximum virtual buffer size it is possible for the Spooling Manager - a paged process - to read or print 128Kb without any intermediate transactions. This is a great improvement on the EMAS system which required the paged Spooling Manager to be activated every 512 bytes read or printed.

Another design aim of the communications software - that of protocol independence - seems to have been achieved, judging by the experience of the University of Kent, who run EMAS 2900. They operate a very different local network, based on the Cambridge Ring, from that in use in Edinburgh but they were able to run the EMAS 2900 software unchanged. All necessary modifications were confined to the Front End Processor software and these were accomplished and tested in a few man months.

Acknowledgements

Many people contributed to the development and evolution of the EMAS and EMAS 2900 operating systems but in relation to the subject matter of this paper I would particularly like to mention D.J. Rees for rigorously implementing the local and global controllers, P.D. Stephens for having faith that such a very different communications system would work, B.A.C. Gilmore for designing and implementing the Front End Processor software, and N.H. Shelness for having had the original idea for the Communications Controller.

References

- BU78 J.K. Buckle,
The ICL 2900 Series, Macmillan, 1978.
- ER80 P.D. Stephens, W.A. Laing, R.R. McLeod, J.M. Murison,
D.J. Rees and J.K. Yarwood,
EMAS 2900: Concepts,
Edinburgh Regional Computing Centre, 3rd Edition 1980.
- RE81 D.J. Rees,
The Structure of The EMAS 2900 Kernel, to be published.
- ST80 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, Vol. 10, 933-1008(1980).
- WH73 H. Whitfield and A.S. Wight,
EMAS - the Edinburgh Multi-Access System,
Computer Journal, 18, 331-346(1973).