file - Multics
design proposals

TO:
F. J. Corbató
C. T. Clingen
R. C. Daley
J. W. Gintell
N. I. Morris
R. A. Roach
J. H. Saltzer ✓
V. L. Voydock
S. H. Webber

FROM: R. J. Feiertag

DATE: June 26, 1972

SUBJECT: Directions Regarding a Design for Interprocess Communication

How about exploring multiple processes
in a single address space as a third area?

This memo addresses the problem of how one should go about designing a new interprocess communication facility. It is assumed that there is general agreement that some new form of interprocess communication is necessary for the future development and growth of Multics. Therefore, the purpose of this memo will not be to justify a new IPC but rather to present some rough ideas of what directions a new design effort should take. Many of the shortcomings of the present IPC will be presented in order to illustrate where it is necessary to correct deficiencies.

Perhaps the most glaring deficiency in the Multics interprocess communication facilities is their lack of scope. Many problems readily identifiable as part of interprocess communication have barely been touched. The most obvious of these is mutual exclusion. At the user level the facilities available for achieving mutual exclusion of processes are barely adequate for straightforward applications. Even within the controlled environment of the supervisor many modules use varied methods of avoiding interference. Also, relatively little work has been done on finding better means of avoiding deadlocks. It is important that better solutions be worked out both for the convenience of users and to make the supervisor easier to program and understand and more reliable.

Another area that has remained largely untouched since its original design is that of interrupting of processes during their execution. This IPS facility was introduced primarily to implement quits. All the problems associated with interrupting a process during execution need examination. This includes inhibiting and masking. It is probably also useful to examine the alternative: not having such an interrupt capability at all, even though it is unlikely that Multics could do without such process interrupts.

There are several other areas within interprocess communication that need a great deal of thought. Protection for processes referencing other processes could be greatly improved, perhaps including access control lists for processes. Another area that requires a great deal of work is the interaction of interprocess communication and the ring mechanism. None of the current interprocess communication mechanisms interface properly with the ring mechanism and once the ring mechanism becomes fully utilized some problems will probably arise. The interface with the ring mechanism is probably not too conceptually difficult, but simply needs someone to workout the ideas.

One of the most important parts of a design of a new interprocess communication would be a coordinated approach. Currently the various interprocess communication facilities, under some degenerate situations, can destructively interfer with one another usually resulting in process terminations. A coordinated design is essential. Such a design should include a few basic primitives upon which all the different varieties of interprocess communication can be built. An understanding of all the desired types of interprocess communication is necessary in order to design the proper primitives and proper modularization. It is, therefore, desireable to study the various types of interprocess communication that are desired, to determine the scope of the problem and then to design a suitable set of primitives to implement the desired facilities. This, of course, is no trivial undertaking, but at least a partial study is possible.

Another area that will require some study is the user interface. The current interface is rather clumsy. The interface should be simple and the procedures inexpensive for simple applications. The primtives should allow simple and inexpensive operation. If such simple and inexpensive interfaces can be developed, then use of interprocess communication could become commonplace and user multiprocessing become a reality.