TO: Distribution

FROM: D. Clark

DATE: February 13, 1976

SUBJECT: MIT Laboratory for Computer Science,
Computer Systems Research Group Task Report.

The attached report covers progress of the Computer Systems Research Division of the Laboratory for Computer Science in the period 1 October to 31 December 1975.

During the last three months of 1975, the following events of note occured:

Allan Luniewski, as part of his research on system initialization, has successfully reconfigured the Active Segment Table.

Phil Janson has completed a PhD thesis proposal describing a strategy for structuring the bottom levels of an operating system.

Raj Kanodia and Dave Reed have completed the draft of a paper describing event counts, the new process co-ordination mechanism that they have developed.

Ken Pogran has introduced a new, more useful program for reading one's mailbox.

Several members of the group attended the Symposium of Operating Systems Principles in Austin, Texas.

These events are discussed in detail in the report.

Kernel Design Tasks

I. Restructuring of Page Control.

Only two steps remain to complete this task. Andy Huber is currently writing up his research. This writeup, which will represent his Master's thesis, will also be published as a Project MAC Technical Report. Bob Mabee is investigating the performance aspects of the new implementation. Initial comparisons between the standard page control and Andy Huber's experimental page control suggest that the experimental version requires about one and one half the time to process a page fault. It seems that this increase in time results from the coding of the experimental version in PL/1, and particularly from the use of a large number of external subroutine calls. We plan to determine the true magnitude of these two influences, in order to discover the intrinsic costs of the two algorithms.

II. Restructuring of Traffic Control.

All reportable progress in this task has been restricted to one particular aspect, interprocess communication. This is reported in task XVI.

III. Restructuring of the Answering Service.

Warren Montgomery has now written all the code required for this experimental redesign of the Answering Service. The testing and evaluation of this code is proceeding in parallel with the documentation of the design. If everything goes well, this project will be completed in the next quarter.

IV. Fast Processes in Ring 0.

We are still attempting to get fast processes installed into the standard system. The current position of Honeywell is that fast processes should not be installed except as part of a system that actually makes some use of them. The concern is that an unused mechanism in the system will not remain operational for very long. It appears that the ARPANET Network Control Program is the most likely candidate for modification so as to use fast processes. We will evaluate this possibility more fully in the next quarter.

V. A Methodology for Designing a Certifiably Correct Computer System.

Rich Feiertag, who is now working for SRI on the west coast, is continuing the writing of his thesis.

VI. Study of Multics System Initialization.

During the fall, Allan Luniewski has written a thesis proposal describing his project. This proposal will be available as an RFC shortly after the first of the year. Allan has also done considerable design and coding work during the fall; in particular, he has completed one portion of his project, the dynamic reconfiguration of the Active Segment Table. The ability to perform this and other reconfigurations of the software and hardware is an important cornerstone of his proposed strategy for a clean and understandable initialization procedure.

VII. Support of User Defined Object Types.

Over the past several months, Doug Hunt has been considering the question of how to implement a system that contains the feature of user defined object types. His particular interest is in discovering techniques by which the lower levels of such a system may be structured as hierarchically ordered layers. The lower levels of a system are those levels that implement the concept of "object" itself. See also task XIV.

VIII. Study of System Reliability and Recovery from Errors.

Harry Forsdick is studying the effect of errors on the operation of a computer system. One topic which he intends to consider is the relationship between the security of a system and the mechanisms in that system for dealing with errors. This project was discussed in earlier task reports. He is also proposing to do a systematic study of the known error recovery mechanisms, with the intention of describing some orderly mechanism for using these in combination so as to maximize the reliability of the system. For this purpose, he is proposing a model of a computer system in which errors can be represented; the model consists of a set of axioms, and errors consist of specified violations of these axioms. Harry is currently writing an RFC describing this research.

IX. Multics Performance Benchmark.

During the fall, we have organized a new undergraduate research project to carry on the development of our performance benchmark. This benchmark, which uses the ARPANET to provide the load on the system under test, was initially developed over the last year. It is largely designed and coded at this point, but still requires a considerable effort to integrate the various parts of the system, and to develop the particular script or scripts which we wish to use for the

benchmark itself. This project is being supervised and managed by Liba Svobodova and Nancy Federman, with some help from Dave Clark.

X. Cryptographic Synchronization for Secure Terminal I/O.

Steve Kent is continuing to develop a set of protocols that use an encryption scheme to provide bi-directional authentication between the user and the computer, as well as to provide security for the user's transmitted data itself. He has completed the coding of a test environment, and has used this environment for the successful evaluation of his proposed protocols. He is now writing a description of this research, which will become his Master's thesis.

XI. Group Study of Directory Control and the Active Segment Table.

During the fall, several members of the AST study group continued their individual investigations, as discussed below. In addition, Jerry Saltzer, Dave Redell, Mike Schroeder, and Dave Clark met several times in an attempt to assess and consolidate the insight gathered by the AST study group. The intention of these meetings is to develop a global plan for the research of the group in this area. Further meetings will be held after the Group Review in January.

XII. Study of the File System "Quota" Mechanism.

Nancy Federman has continued to examine all the code in the system related to the quota mechanism. It seems that the existance of quota in its present form is a major source of complexity. There is an interesting research problem, as yet unstudied here, which is to consider what forms of resources should be the basis for charging and resource control. If, for example, the user were charged for the "size" of the segment as he sees it (perhaps the current length), rather than the number of non-zero pages in the segment, the accounting problems related to disk usage would be reduced.

XIII. Separating Page Control and Segment Control Functions within the Active Segment Table.

Drew Mason continues his study of the mechanisms which cause Page Control and Segment Control to interact badly. His goal is to decouple them in such a way that page control need not know which segment a page belongs to in order to bring it into core at page fault time. This requires a solution to "the quota problem", and the elimination of various other per-segment attributes

which page control maintains.

XIV. Structure of Software to Implement Objects within the Computer System

Phil Janson has written a PhD thesis proposal in which he describes a technique for the orderly structuring of that portion of the computer system whose function is the implementation of the base level objects (e.g. segments) out of which the system storage is implemented. The code which implements these objects is perhaps the most difficult of all system code to structure since it cannot make use of the storage abstractions that it is realizing (or if it does, without causing difficulties of a different sort). His thesis proposal, which has been signed by Mike Schroeder, will be available as an RFC shortly. Doug Hunt is also exploring this general area; he is interested in the low levels of a system which is able to provide objects of extensible type to the user. He also has an RFC in preparation.

XV. ITT Elimination.

Bob Mabee has, as time permits, been working on a scheme to eliminate the ITT from the kernel of the system. He has been prevented from working on this project due to other, more pressing tasks. We continue to hope that a design document will appear.

XVI. Event Counts, A New Process Control Mechanism.

Raj Kanodia and Dave Reed are, as mentioned last time, preparing a paper describing a new mechanism that they have developed for the co-ordination of process. The paper has now been through a complete revision, and should be available soon. This new co-ordination mechanism has several nice features not found in other known schemes, for example, it does not require that those waiting for an event have write access on the co-ordination element, which simplifies correctness proofs, and it seems to be especially appropriate for synchronization of distributed processes. It is a simple mechanism that seems to model many actual co-ordination situations very naturally.

XVII. Provision of "Breakproof" Environment for User Programming.

As various parts of the operating environment are removed from the kernel, the question arises as to where they should be put. If they are placed in the same ring as the executing programs of the user, than they can be destroyed by a programming error of the

user. It would be very nice if the removal of programs from the kernel did not lead to a reduced robustness of the programming environment. Jeff Goldberg is attempting to define a consistent collection of user support programs that can be protected from destruction by casual error of the user. This might include, for example, the linker and its tables, the linkage sections and the LOT. These might be protected by placing them in an inner ring, which contained data private to this process. An RFC describing his particular proposal is now in preparation.

XVIII. Census of Ring 0.

About two years ago, Vic Voydock performed a census of ring 0, in which he summarized the total size of the programs in the kernel, and divided them into catagories according to the subsystem to which they belonged and what language they were coded in. Skip Ellis is currently redoing this census, in an attempt to find out what has happened to ring 0 in the last two years, and also to assess the impact of the new storage system, which will be installed in a few weeks.

XIX. Performance of the Multics at University of Southwestern Louisiana.

Jerry Saltzer and Dave Clark recently visited the Multics installation at USL. They were particularly interested in the performance of this machine, since it is configured with one million words of main memory and no bulk store. The various performance measures which were taken on the machine will be reported in an RFC to come out shortly. The machine does perform very well, as expected. With one CPU, it runs out of processor power at about 45 to 50 users, and then slows down in a smooth way proportional to the total demand. One surprising discovery is that page fault processing time is very high to the disk, as high as 6 or 7 ms. per fault, as compared with 2-2.5 ms. on our machine. Preliminary investigation by Bob Mabee suggest that this is due to write queues building up on the disks, caused by the find core algorithm in page fault. He is exploring the problem further.

XX. Input/Output in a Security Kernel.

Input/Output represents an area of the security kernel which the current research of the group has not investigated yet. In particular, the question of how to handle multiplexed I/O connections requires some consideration, since the operation of multiplexing and demultiplexing information over the connection must be protected as a kernel function. Gene Ciccarelli has

been exploring the current implementation of tape software on Multics now, with the intention of determining how much its throughput would be degraded if tapes were operated through a network interface of the sort which now controls the ARPANET. During the next months, Dave Clark intends to consider how multiplexed devices such as a network should be handled inside the kernel, and to what extent this is a viable interface for all user I/O.

Technology Transfer and Network Related Tasks

I. Improvements to the Network Control Program

The development of the new version of the NCP, which was described in the last task report as nearing completion, has been somewhat delayed by the discovery of various persistent bugs in the code. In examining the code to fix these bugs, Doug Wells has also discovered a number of flaws in the code, which may have contributed to various sorts of unreliable operation that we have occasionally experienced in the past. Thus, while the installation of this code has slipped somewhat, we have as compensation its improved overall quality. We really do believe that at this point the code is nearly ready for installation, since it has passed all of its tests. Installation may, however, be somewhat delayed, since this will inevitably go in after the new storage system.

II. Production of MPM Network User's Supplement.

During this quarter we have made limited progress in preparing module documentation for various subroutines and commands which will be part of the network environment. In particular, Doug Wells has started to write specifications for the various network IOSIMs and user interface commands.

III. Improvements to Network Mail Facilities.

Ken Pogran has implemented a new program for reading mail from a mailbox. This program, called read_mail, is considerably more powerful than the installed mail command, allowing for the selective reading, deleting, forwarding, and filing of individual messages. This program is in response to the more powerful requirements placed on our mail system by its network users, and is part of our attempt to provide a more sophisticated mail handling environment. The program is currently available for experimental use; contact Ken Pogran for details.

Steve Swernofsky continues his rewriting of the net_mail command and the mail queuing facility. He has not been able to devote a large amount of his time to this project during the fall, due to other commitments, but hopes to be able to work on it over IAP.

Ken Pogran is serving as a member of the ARPA committee on network message service, a committee intended to standardize and improve the facilities used to send mail between the various hosts in the ARPANET. His

participation provides us an excellent opportunity to stay in touch with the current attitudes toward network mail, and to make sure that our system remains compatible with any new mail system which may be installed in the future.

IV. Specialized Inter-Host Tools for Development Machine.

The existance of the network connection between the development machine and the service machine has led to the creation of commands for moving files between one machine and the other. We are currently proposing to provide several new features of this sort. For example, we would like to make the printer on the development machine accessable through the dprint command on the service machine, and we would like to be able to queue requests for file transfers, so that they occur whenever both systems happen to be available for the operation. In addition to the obvious functional advantage of these tools, they will provide us an opportunity to experiment further with user ring multitasking in order that we may gain confidence that the multitasking software is, and will remain, operational. Doug Wells and Nancy Federman are working on the development of this software.

V. New Login Protocols for Answering Service.

Arthur Gottlieb is currently involved in a rewriting of a piece of the Answering Service, in order to eliminate a most unfortunate interaction between the Answering Service and the File Transfer Protocal for the ARPANET. The problem arises whenever an FTP program on one machine attempts to login a server on another machine in order to affect a transfer, because the FTP specifications define the syntax for the login sequence, and, unfortunately, the syntax is not that which our Answering Service expects. For example, instead of the work "login", the FTP uses "USER". This problem is currently solved by the insertion of a program, called the transmogrifier, in the input stream of the Answering Service for network connections attempting to perform an FTP login. The chief difficulty with this approach is not the translation of an FTP word into its equivalant Answering Service word, but the translation in the reverse direction, when an error message from the Answering Service must be analyzed and translated into the equivalant FTP error message. Whenever a new message is added to the Answering Service, the transmogrifier almost always fails. To eliminate this problem, we are modifying the Answering Service so that the normal login proceedure, dialup, can be replaced by an alternative which uses a different syntax for logging in and generates a

different set of error messages. Once this is done, not only will we be able to perform FTP logins more reliably, but the mechanism will be available to other subsystems that wish to present a specialized appearance to the user from the moment of his very first interaction.

VI. Exploration of Limits of Bandwidth of ARPANET.

In the past, we have reported changes which were made to our Network Control software for the purpose of increasing the speed of data transmission. Raj Kanodia is currently performing an experiment to determine if the currently observed limit on the transfer rate between two Multics machines is due to the IMP, the host, the 50 kb. phone lines, or the IMP-host cable.

VII. Development of National Software Works on Multics.

We continue to report little activity in this area. Doug Wells has attended a number of meetings, and it appears that in the next quarter protocols may be specified that we will be interested in implementing.

VIII. Distributed Data Base Management.

Art Benjamin is continuing his investigation of distributed file systems and the use of a data network for improving the reliability of on line data storage.

IX. Implementation of RSEXEC on Multics.

During this quarter, Tony Smith has started his design of a user interface to RSEXEC on Multics. His first project is the creation of a version of hcs_$initiate that will detect that the path name of the object to be initiated describes a segment on some other system, and will make a local copy of that and initiate the local copy. The code for the server RSEXEC is apparently complete, but before it can be placed in operation, a new version of the NCP must be installed, and certain small changes to the Tenex user RSEXEC will be required for compatibility.

X. Study of a Local Network for Project MAC.

During this quarter, Ken Pogran has continued his designing of a data communication network for Project MAC. As part of this design, Project MAC invited Bob Metcalfe to come and discuss his ETHERNET, and we have invited Dave Farber to come and discuss his ring net architecture. Ken is also considering other communications issues for Project MAC, such as the