PROJECT MAC

July 21, 1975

Computer Systems Research Division

Request for Comments No. 86

ANNUAL REPORT

by J.H. Saltzer, M.D. Schroeder, and D.D. Clark

Attached is the first draft of our division's annual report for the year July, 1974 - June, 1975. Please report any suggestions or corrections to one of the authors. Make a particular effort to check the list of publications, talks, and theses for errors and omissions.

PROJECT MAC

Computer Systems Research Division

July 15, 1975

DRAFT

ANNUAL REPORT: JULY, 1974 - JUNE, 1975

by J. H. Saltzer, M. D. Schroeder, and D. D. Clark

OVERVIEW

The Project MAC Computer Systems Research Division engages in pragmatic engineering studies of computer operating and information storage systems, with the goal of developing conceptual insight from practical experience. Currently, CSR activities are focused in three areas, one of which is just in the planning stage: 1) protection of information and the correctness of system facilities affecting security, 2) network interconnection of computer systems, and 3) physically distributed implementations of information storage systems.

Two thirds of the effort of our division is in the computer security activity, for which some background is in order. The United States Air Force has contracted with Honeywell Information Systems Inc. (HISI) to create a secure general purpose information processing system based on Multics. HISI describes this activity as "Project Guardian". As part of Project Guardian, HISI has contracted with Project MAC to do research to discover the essential mechanisms needed to support controlled information sharing. The current reporting year is the second year of a four-year project, so final results are not

available. However, considerable progress has been achieved, as discussed in detail in the section of this report devoted to computer system security.

Most of the remaining attention of the division is engaged in computer network activities, especially the ARPANET. This Spring we began a new activity, participation in the ARPA National Software Works (NSW) project, by helping design NSW protocols and by implementing Multics versions of NSW protocols. We have also initiated a technology transfer dialog between HISI and ARPA, with the goal of making the HISI Phoenix site the fourth Multics site on the ARPANET, and making the Multics ARPANET attachment software an optional feature of the HISI Multics product. Another activity just started has the goal of linking together Project MAC terminals, private computers, PDP-10's, Multics, and the ARPANET. These projects are described in more detail in the section of this report devoted to network related activities.

The third area is a new one, and represents a probable major area for activity for the next few years. In order to create information-storing systems that are robust, expandable, and that take advantage of modern technology, it seems desirable that they should be implemented out of physically separable components, perhaps distributed geographically. On the other hand, there is only superficial understanding of how to organize a shared information storage system as a centralized facility, much less in a distributed form. We have thus been exploring

this problem, looking for a suitable experimental vehicle. We anticipate that in the coming year, this activity will result in a detailed proposal for a new experimental project.

COMPUTER SYSTEM SECURITY

Introduction

The overall plan for this project to engineer a security kernel for Multics was described in last year's annual report. To review, the specific goals are to identify the minimum mechanism that must be correct to guarantee computer enforcement of desired constraints on information access, to simplify the structure of that minimal mechanism to make certification by auditing possible, and to demonstrate by test implementation that the security kernel so developed is capable of supporting the complete functionality of Multics. Because Multics permits direct sharing of information among computations, this research can lead to a better understanding of the structures necessary to support the primary functional advantage of shared computer systems, high-bandwidth communication among users, as well as lead to a system whose security inspires a high degree of confidence.

At year's end, the computer security project has run for about half of its intended four year span, and most of the specific tasks in the initial set are nearing completion. So

far, the reductions in size and the simplifications in structure of the security sensitive software in Multics that were expected to result from these initial tasks seem to be occurring. As the results of these first round activities are becoming available and being assimilated, a second round of activities is being planned. In what follows, the current state of the project is described by discussing four catagories of activities: those described in the last annual report that were completed this year, those described last year on which work continues, new activities started this year, and finally new tasks planned for the next year.

Completed Tasks

1. Removal of the Linker from Ring 0.

   Most of the design and implementation work on a linker that does not need to run in the protection domain of the supervisor was reported last year. During this year the performance of the user ring linker was more thoroughly evaluated and found not to be degraded from the old design, and a plan for installing this new linking facility in the standard system was developed. It now appears very likely that HISI will adopt the new design as part of the standard system in the near future. (See MAC TR-132 by P. Janson for more details on this task.)

2. Removal of Name Space Management from Ring 0.

   This project was to remove from the supervisor the facilities for managing the association between reference names and segments in the address space of a process. During the year implementation and testing were completed, and a plan developed for installing the new facility in the standard system. We expect it to be adopted by HISI at the same time the new linker is adopted.

   Removing the reference name management mechanism from the supervisor required that a data base central to the management of the address space of a process -- the known

segment table -- be split into a private and a common part, and that the supervisor learn to lie convincingly on occasion about the existence of certain file system directories. The result of the removal is a reduction by a factor of five in the size of the protected code needed to manage the address space of a process. Another result is a new simpler interface to the file system portion of the supervisor. Instead of identifying a directory with a character string tree name, a segment number is now used. The algorithm for following a tree name through the directory hierarchy to locate the named element is thus removed from the supervisor. Performance tests indicate that the new design outperforms the old as well! R. Bratt's master's thesis describing this work will be ready early in July, and the MAC Technical Report in early fall.

3. Processes in the Kernel.

Several of our proposed changes to the kernel will require the use of separate processes to execute portions of kernel algorithms. To meet this requirement R. Mabee has implemented a facility that provides inexpensive, efficient processes inside ring 0. These processes lack some of the capabilities of a full Multics process, for example they lack the ability to initiate a new segment in an address space, but these restrictions make it possible to use these processes to implement low-level kernel algorithms such as core management. The implementation of this process facility is completed, and we are awaiting installation of the mechanism in the standard system so that we can begin exploiting processes to restructure the kernel.

Continuing Tasks

1. Restructuring Page Control.

A. Huber, as his master's thesis research, has taken on the task of restructuring page control as described in last year's annual report. The goal of this research is to utilize several asynchronous parallel processes to perform the various functions of page control. In particular, separate processes can be used to remove pages from memory and from the paging device so that a free storage pool will always exist to be used for the servicing of page faults. It is felt that the use of parallel processes in this context provides an intrinsic simplification to the algorithm, since it eliminates certain artificial interactions that occur if the functions are performed as part of the same process, which

constrains them to run in a particular synchronized order. We also feel that separation of the algorithm into separate processes will allow the page fault processing to scale up more effectively to a larger system, by eliminating contention on the global page table lock. Design of this multiprocess page control is essentially complete at this point, and coding has been started. This thesis should be completed during the fall term.

2. Restructuring Traffic Control.

D. Reed, as his master's thesis research, has proposed a restructuring of the traffic controller that partitions its functions into two levels. The lower level multiplexes the real processors of the system among a fixed number of so called virtual processors. By fixing in advance the number such virtual processors, this low level processor multiplexer need make no use of the system's virtual memory facilities. Thus there is a strict isolation and ordering between the multiplexer and the virtual memory. A higher level scheduler multiplexes some of the virtual processors among all of the currently operating real Multics processes. This higher level scheduler can use all of the facilities of the Multics virtual memory, since they are implemented at a lower level. It is expected that this restructuring, in addition to clarifying the relationship between traffic control and page control, will have various other benefits such as separating the idea of interprocess signaling from the idea of traffic control. In this proposal, no ring 0 data base (such as the current message table) will be needed for messages between processes. Rather, messages between processes will be sent using segments that are protected using the standard system access control mechanisms. This appears to be a great simplification over the current mechanism. This thesis should also be completed during the fall term.

3. Restructuring the Answering Service.

The answering service is a large interconnected set of functions, all of which are security sensitive given the current modularization. W. Montgomery, as his master's thesis research, is considering a rearrangement of the answering service to achieve an isolation of those particular components that are in fact crucial to assure secure operation of the system. To this end, he has applied a similarity between the creation of a new process and the entering of a new protection domain to allow access control lists to be used to regulate the creation of processes on any particular user's behalf. In general, his scheme avoids the need for certified software by

providing the means to assure any user that a process created with that user's identification will start executing only in certain specified programs that the user provides. These programs, using various tools provided, can determine that the process has been brought into execution under appropriate circumstances. This project has proceeded to a fairly detailed design stage, and the thesis should be completed during the next year.

4. Study of Multics System Initialization.

If one is to certify that a system works correctly, one must begin by verifying the "initial state" of that system. For this reason, it is very important to understand how the Multics system initializes itself. The current initialization procedure is relatively unstructured and confusing, and is apparently not amenable to verification. A. Luniewski has proposed a restructuring of initialization that reduces the amount of code in the initialization phase and that simplifies the task of verifying it. His approach has been that much of what is now considered initialization ought rather to be considered as reconfiguration. What follows from this approach is a decomposition of initialization into two phases. The first phase involves getting a minimal Multics up and running on any configuration, the second phase involves a series of reconfigurations based on input describing the actual configuration. The advantage of this strategy is that all of the reconfigurations run in a complete operational Multics environment, which is much easier to understand than a partial and ever changing environment presented in various stages of initialization. Further, since the minimal Multics is independent of configuration, it can be largely generated as the system tape is created. Algorithms that run at the time the system tape is created are easier to verify since they run in a fully operational Multics environment. Detailed design of this restructuring is currently being performed, and the thesis should be completed within the next year.

5. Multitasking in the User Ring.

The purpose of this task is to provide an environment that, in the context of a single ring, allows the user to write various programs as if they are executing in separate cooperating processes. In fact, the execution of these processes is supported by multiplexing the one single process of the user. Thus, we describe this environment as multitasking rather than multiprocessing. Clearly, the creation of this programming environment in the user ring does not directly contribute to simplification of the kernel. However, this environment

allows us to experiment with different techniques for dealing with the handling of the quit signal propagated to the user ring by the kernel. In particular, the quit can, inside the supervisor, be translated into a process wakeup. This change to the processing of the quit signal inside the supervisor seems to imply a great simplification to this particular mechanism. The multitasking environment has been completed by D. Wells and is usable, but is distinctly experimental at this point. It has yet to be applied to handling quit signals.

6. Computer System Design Methodology.

The work this year has been concerned with developing a general methodology for designing certifiably secure systems. A method has been formulated and is being tested by attempting to design a security kernel for a system with functional capabilities similar to Multics. The basic design is complete and some of the abstract programs of the system have been written in the CLU language. Work is under way on making the verification of the resulting system easier, and improving the efficiency of the resulting system without making verification harder. Documentation of the methodology, the design of the Multics-like system, and other results are in progress as a Ph.D. thesis by R. Feiertag.

7. Study of Multics Security Holes.

As part of our continued interest in attempting to discover ways to circumvent the security of Multics, we are maintaining a list of all known security flaws in the current system. H. Goldberg has been working on producing the annual revision of this list. An attempt is made to analyze each flaw to determine ways of preventing reoccurrence in future versions of the system.

8. Restructuring of Network Control Program.

This task has been active in the past, but it is currently not progressing due to lack of manpower. It is hoped that it will be possible to reactivate it in the near future. The central idea behind this task is that the Network Control Program represents an ideal candidate to use in an experiment involving a multiple process implementation of a control algorithm on Multics. The Network Control Program is concerned with the flow of data to and from the ARPANET. The principal job of the Network Control Program is the management of a multiplexed communication path, which implies the management of multiplexed buffers. It is proposed that the flow of data among these various buffers be implemented by processes, and that the virtual

memory be used to simulate buffers of infinite capacity. Several possible restructurings have been considered during the last few months, but no further design work can be expected until more pressing tasks have been eliminated, chief among which are the NSW demonstration described later in this report.

New Tasks

1. Cryptographic Synchronization for Securing Terminal I/O.

   This year S. Kent has started a master's thesis to explore the protocols required to utilize cryptographic synchronization to secure user to computer and computer to user communications. The application of encryption to terminal I/O has become feasible with the introduction of effective encryption algorithms that can be implemented on a single chip, e.g. IBM's LUCIFER algorithm, and thus can be put in most terminals. Properly employed, encryption can provide highly secure user to computer and computer to user authentication, as well as protect against unauthorized release or undetectable modification of transmitted data. In order to realize these advantages, protocols must be engineered for key generation and distribution, authentication, message sequencing, and resynchronization following line disruption. This activity will include developing a set of protocols and evaluating their impact on the system's human interface by simulating a test implementation in Multics using the ARPANET.

2. Independent Domains and Breakproof Services.

   This new task, being developed by H. Goldberg as a potential master's thesis, is to explore some of the implications of removing certain traditional supervisor functions from the Multics security kernel and to explore extending the functionality of the Multics protection mechanisms to allow multiple, independent domains to be part of one process. A traditional supervisor includes many mechanisms that are not security sensitive simply to protect these mechanisms from accidential damage from user errors. To produce a security kernel for Multics, many such mechanisms are being moved out of the supervisor. By moving them to the user environment, mechanisms such as the linker, the reference name manager, and the search rules become breakable, which could make the system harder to use. Fortunately, the Multics protection rings provide a place to protect non-kernel mechanisms that should be breakproof. They can execute in a ring, say ring 3, above

the kernel but below the normal user ring. Because all data bases managed by these service mechanisms are private to a process, they are not part of the security kernel and need not be certified. Yet they cannot be broken inadvertently by user errors. Part of this project is figuring out how to provide such breakproof services for Multics.

The second aspect of this project concerns protected subsystems. Multics has always supported user-defined protected subsystems, although the protection rings can provide only one way protection. It is not possible, however, to use the rings to protect both subsystems and breakproof services at the same time in the same process without making the breakproof services common to all subsystems in a process and therefore part of the security kernel for those subsystems. The essential difficulty is the total ordering of privilege implied by the protection rings. Thus, to provide breakproof services some other way must be found to protect subsystems, if the functionality of protected subsystems is to be maintained. The method being explored is simulating multiple independent domains (containing rings) in a process using multiple descriptor segments for a process. This project is just beginning.

3. Support of Extended Type Objects.

A directory can be considered as an object defined in terms of a lower level object, the segment. It is possible that the mechanisms that define the directory could be generalized to allow the definition of new user-defined object types. This sort of ability has been provided in systems that are based on capabilities but not in systems that are based on access control lists. D. Hunt has been considering the question of whether user-defined object types can be supported in a system such as Multics, hoping that the project will lead to a Ph.D. thesis.

4. Study of Relationships between Security and Reliability.

H. Forsdick has been studying the relationship between the security of a system and the reliability of that system. In the Multics system it is presumed that a system failure may have an unknown effect on the security status of the system; this is the reason that the system is shut down, salvaged, and restarted after every unexplained system failure. It is not obvious, however, that security is directly dependent on reliability. If it were possible to determine that certain classes of computer failure could not influence the security state of the system, then the

two functions would have nothing to do with each other. More strongly, it is possible that a highly reliable system contains mechanisms that are not desirable from the viewpoint of security. For example, one way to increase the reliable storage of information on a system is to make several copies of that information; many copies, however, increase the probability that the information may be compromised. In an attempt to understand better the relationship between security and reliability, Forsdick has undertaken a study of a variety of systems that maintain high reliability as one of their goals, to investigate mechanisms in the system in the light of their implications for the security of the information stored on that system.

5. Multics Performance Benchmark.

As modifications to the system are proposed by our group, it is important to be able to to determine the performance effects they have. For this reason, it is very desirable that we have a stable and reliable performance meter. During the year, we have spent a certain amount of effort attempting to reduce the cost and increase the stability of the currently available benchmark. We have to some extent been successful, although further improvements are still possible. We are also developing a version of the performance benchmark that can be run using the ARPANET to drive the test processes. This seems to be a potentially interesting variant on the normal benchmark, since it provides a test of the I/O system.

Plans

As the initial set of tasks is being completed, two areas present themselves as the focus for the next set: directory and segment control, and input/output.

The removals of the linker and of name space management from ring 0 are the first two steps in the restructuring of the file system. Now that these two layers have been pushed out of the security kernel, the rest of the file system can be dealt with. What remains is directory control, which manages the

directories that are the nodes of the file system hierarchy, and segment control, which manages the activation and deactivation of segments in the virtual memory. Together, they constitute one of the largest, most complex, and least well understood components of the system. Success of the certification project is conditional upon achieving a significant simplification in these two areas. As a first step, D. Redell is leading an intensive study of the existing mechanisms of directory and segment control. The two general questions being addressed by this study are 1) can any mechanisms be removed from the security kernel? and 2) how can the structure of the part that must remain in the kernel be simplified? The tentative conclusion on the first question is that no substantial piece remains that can be removed without substantially altering the existing functionality of the system. With respect to the second question, the current ideas for simplifying the structure of the file system involve decomposing it into two layers. The bottom layer would implement a single catalog that maintains a physical description of each segment and is indexed by a segment unique identifier. This layer would also implement the controls on sharing of information among various security compartments. The naming hierarchy would be implemented on top of this first layer. The directories of the naming hierarchy would be much simplified by virtue of the removal from directory entries of all physical attributes of segments. The structure of the active segment tables would be simplified by the ability to locate the physical attributes of a

segment directly from its unique identifier and without referencing the directory hierarchy.

Two specific tasks so far have been identified as useful first steps in simplifying directory and segment control. One is the removal of the storage space quota and accounting mechanism from the directory hierarchy and the other is reimplementing the segment backup mechanism to be independent of the date/time modified information in directory entries. These two tasks appear to be necessary first steps that have large potential payoffs, and will be started as soon as possible. Other specific tasks will be identified as Redell's intensive study is completed this summer.

In the input/output area, our plan is to perform another intensive study lead by D. Clark starting later this summer. The study will investigate the possibility that common elements can be identified in the software that is required to interface to different multiplexed communication streams. Thus, for example, it is possible that there exists a similar function in the software that interfaces the ARPANET and the software for the DATANET 6000 communications processor. A buffer manager is an obvious example of a possible common routine. It is very interesting to try to identify these modules and, if possible, isolate them, since this would markedly reduce the amount of code inside the kernel, and since multiplexed streams, in particular networks, represent one way of bringing all I/O to the computer system. If all multiplexed communication streams could be

handled by one set of kernel modules through a network oriented interface it would provide a great simplification of the kernel. The feasibility of forcing all I/O through a network oriented interface will depend the bandwidth that can be obtained as well as the functionality of such an interface.

NETWORK ACTIVITIES

Introduction

The activities of the Network Group during the year can be divided into two categories: a continued effort to export our developments and our expertise in the network area, and an increased pursuit of new research topics.

During the year, two new Multics sites were connected to the ARPANET using the hardware and the software developed by our group. The first of these was the Multics system that is used for the development of new versions of the system, and that we use for the development and checkout of new network software. The other new system is the Multics site at the Rome Air Development Center at Rome, New York. One of our group has visited the Rome site to train network support staff and introduce Multics users to the net. These two sites represent the first installations of a Multics connection to the ARPANET outside the M.I.T. site; we feel that it is a good indication of the stability of our implementation that these sites were

brought on the net with essentially no complications.

Our group initially designed and fabricated the hardware interface between Multics and the ARPANET. We have, however, managed to hand off responsibility for providing new copies of this hardware to the Electronic Systems Laboratory of M.I.T. This laboratory has developed a production version of the hardware interface, and is prepared to supply small quantities of these to new Multics sites wishing to be connected to the ARPANET. The first of their interfaces was used at the Rome Multics site. Their interface will also be installed at the M.I.T. site, at which point our group should be free from any further maintenance responsibility for the network interface hardware.

We are still actively pursuing the goal of persuading HISI Information Systems Inc. to take over official support for the ARPANET software on Multics, and we feel that significant progress is being made. We are currently attempting to generate interest within HISI in connecting the HISI Multics site in Phoenix to the ARPANET, as a means of demonstrating to HISI the utility and viability of the ARPANET. While it is still too early to tell how this idea will develop, initial suggestions from our group have met with enthusiastic responses at several levels inside HISI.

Performance Improvements

During the year, the data transfer rate between Multics

and the ARPANET was considerably improved. The data rate from Multics to the network has always been lower than the rate from the network to Multics. During the year, it was determined that this low rate was due to a bottleneck caused by a small buffer size. A simple change that provided optimal sized buffers for large messages while still using small buffers for short messages increased the overall transmission rate from ten Kbps to forty Kbps for machines on adjacent IMP's; and to fifty Kpbs for two machines on the same IMP.

Improvements to Network User Interface

During the year we have improved many of the programs that provide user access to the ARPANET. We have installed an improved version of the user interface to TELNET and File Transfer Protocols. These new versions support the new protocols, are more flexible and powerful, and are easier to use. We have also built software that allows the network to be used via the Multics standard I/O interface. While such software has existed before, it has been in a somewhat incomplete form. The software now available is powerful enough to allow experimental programs to access the network in sophisticated ways using a high level interface.

We continue to devote effort to improving the mail service on Multics. We have installed a mechanism that allows mail to be sent to Multics users over the network without the sender of the mail having to know the project name of the recipient. This was

a persistent annoyance to users on machines other than Multics, who were unfamiliar with our concept of project identifier. We have implemented a mechanism for forwarding of incoming mail, so that a user no longer on this machine may have his mail sent to the network host on which his mailbox is currently located. We have just completed a new version of the network mail sending program that provides several new features, such as the ability to send mail to a predefined list of recipients.

Perhaps the most important improvement in the user support area is the publication of a preliminary edition of the Network User's Supplement to the Multics Programmers' Manual. This document provides a description of all the commands that may be used to access the network from Multics and also a description of the various facilities available to network users of Multics. While this document is still incomplete, and will be formally issued sometime in the next year, the preliminary edition is sufficiently complete to distribute to other Multics sites, such as the site at Rome, New York, just coming onto the ARPANET. A significant amount of effort went into the preparation of this document.

Research Activities

Since January, our group has had considerable involvement in the project to build the National Software Works using the ARPANET. Our role in this project has been first, to serve as a implementation site that is not a Tenex machine, so that any

potential Tenex dependencies in the NSW protocols can be identified and eliminated, and second, to provide a more general critical review of the various protocol definitions. To this end, we first familiarized ourselves with the NSW documentation, a task that required considerable time due to the preliminary state of the protocols and their descriptions. Next, we attempted a trial implementation, which was quickly carried to the point where we could proceed no further because of uncertainties and instabilities in the protocols themselves. These activities led to a continued dialogue with the designers of the NSW protocols that we feel was beneficial to the development of the protocols themselves and has also aided us greatly in achieving a deeper understanding of some of the issues and pitfalls in the design of a strategy for sharing resources among various computers. Our current NSW activity is to identify various possible solutions to the problems that the NSW protocols are attempting to solve.

Another project has been the implementation of the server for the RSEXEC protocol. We feel that this server, which is now essentially complete, will serve a very practical function in allowing us to transfer files between various Multics sites as well as between Multics and Tenex systems; its implementation has also given us additional insight into the issues of resource sharing across the network. We propose to implement a user interface to RSEXEC on Multics as time permits.

Although more pressing tasks such as NSW have taken up

most of our time, we have continued our project of designing a new Network Control Program, as described earlier. A master's thesis underway in the group is exploring the redesign of the Multics information backup system. A. Benjamin proposes to replace the current backup storage medium, tape, with a network connection to another file storage system, such as the Datacomputer. Although the bandwidth would be somewhat restricted, the random access available from the Datacomputer would provide a greatly increased functionality for recovery of lost information.

Project MAC has been considering the question of whether a local network is a useful tool for its future research. D. Clark and K. Pogran participated in a MAC committee to consider the relevance of such a net. As a result of this committee's recommendations, Pogran is starting an intensive study of the technology and functionality of a possible Project MAC network.

PHYSICALLY DISTRIBUTED INFORMATION SYSTEMS

The division has been considering a new major research effort in the area of distributed storage and management of information. The effort this year has consisted of preliminary planning, with the intent to discover the most interesting and fruitful areas for research effort. Additional groundwork for this project has been provided by several of our projects

described earlier that have served to provide us with experience with distributed information management. In particular:

> The National Software Works provides access to data files by the various NSW tools running on various hosts. Thus the NSW must support a distributed file system.
>
> RSEXEC, the resource sharing protocol discussed above, provides the user of one system access to the file system of another. Thus, it directly faced the issue of providing distributed information storage that appears to be centralized to the user. While the capabilities of RSEXEC are somewhat restricted, it is a valuable protocol with which to have experimented.
>
> Benjamin's work providing an information backup system using the Datacomputer provides a chance to deal with the issues of maintaining multiple consistent copies of segments and of accessing these in a manner invisible to the user. A backup system with random access capabilities may in fact be a very general and powerful example of a distributed information management system.

During the next year we intend an exploration of systems that, although constructed of physically distributed components, appear to the users as uniform storage systems. We hope to identify a system organization that, by virtue of its distributed nature, is very robust and scales up to large numbers of users and large data stores, and that hides the actual differences between local and distant objects.

PUBLICATIONS, TALKS, AND THESES: JULY 1974 TO JUNE 1975

Publications

Saltzer, J.H., "Ongoing Research and Development in Information Protection," ACM Operating Systems Review 8, 3, (July, 1974), pp. 8-24.

Saltzer, J.H., "Protection and the Control of Information Sharing in Multics," Comm. ACM 17, 7 (July, 1974), pp. 388-412.

Pogran, K.T., Network Users' Supplement to the Multics Programmers' Manual, (Initial Release), January, 1975.

Sproull, R.F., and Thomas, E.L., "A Network Graphics Protocol," ACM Computer Graphics 8, 3 (Fall, 1974).

Technical Reports

Redell, D.D., "Naming and Protection in Extendible Operating Systems," Ph.D. thesis, University of California at Berkeley, MAC-TR-140, November, 1974.

Richards, M., Evans, A., and Mabee, R., "The BCPL Reference Manual," MAC-TR-141, December, 1974.

Theses Completed

Rudisin, G.J., "Multics Implementation of a Server for the ARPANET RSEXEC Protocol," S.B. thesis, M.I.T. Department of Electrical Engineering and Computer Science, May, 1975.

White, P., "Development of a Benchmark for a Burroughs 7700 Computer," S.B. thesis, M.I.T. Department of Electrical Engineering and Computer Science, May, 1975.

Sandman, J., "Insertion Supervisor for a Direct Numerical Control Component Insertion System," S.B. thesis, M.I.T. Department of Electrical Engineering and Computer Science, May, 1975.

Theses in Progress

Bratt, R.G., "Minimizing the Naming Facilities Requiring Protection in a Computing Utility," S.M. thesis, M.I.T. Department of Electrical Engineering and Computer Science, expected date of completion, July, 1975.

Reed, D.P., "A Simple Implementation of Processes as a Basis for a Certifiable Computer Utility," E.E. and S.M. thesis, M.I.T. Department of Electrical Engineering and Computer Science, expected date of completion, January, 1976.

Benjamin, A.J., "Improving Information Storage Reliability Using a Data Network," E.E. and S.M. thesis, M.I.T. Department of Electrical Engineering and Computer Science, expected date of completion, February, 1976.

Huber, A.R., "A Multiprocess Design of a Paging System," S.M. thesis, M.I.T. Department of Electrical Engineering and Computer Science, expected date of completion, September, 1975.

Montgomery, W.A., "A Secure and Flexible Model for Process Initiation for a Computer Utility," E.E. and S.M. thesis, M.I.T. Department of Electrical Engineering and Computer Science, expected date of completion, February, 1976.

Wilens, M., "High Level Language Memory Management in Modular Memory Computers," S.B. and S.M. thesis, M.I.T. Department of Electrical Engineering and Computer Science, expected date of completion, September, 1975.

Talks and Presentations

Feiertag, R.J., "A Methodology for Designing Certifiably Secure Computer Systems," given at:
University of Illinois, 3/3/75
Purdue University, 3/5/75
University of California at Berkeley, 3/10/75
Stanford Research Institute, Menlo Park, California, 3/12/75
IBM San Jose Research Laboratory, 3/14/75
University of Utah, 3/17/75
Cornell University, 3/20/75
City College of New York, 4/1/75

Saltzer, J.H., "Current Research on Information Protection," given at:
IBM Yorktown Research Laboratory, 7/19/74
I.L.P. Symposium on Information Privacy and Computer Security, 9/24/74
Honeywell Symposium on Privacy and Security, Phoenix, Arizona, 4/30/75

Saltzer, J.H., "Some Unsolved Protection Problems," given at:
ACM 1974 National Computer Conference, 11/12/74

Saltzer, J.H., Session Chairman, "The Impact of Technology on System Organization," AFIPS National Computer Conference, 5/21/75

Thomas, E.L., "Possible Standards in the Area of Graphical Inputs," given at:
Conference on Computer Graphics and Interactive Techniques, Boulder, Colorado, July 15-17, 1974.

Pogran, K.T., "The Use of Multics," given at:
ARPA Information Processing Techniques Office, Washington, D.C., 10/30/74

Pogran, K.T., "Multics and the ARPA Network," given at:
Rome Air Development Center, New York, 5/8/75

Hunt, D.H., "Implementing Extended Type Objects Using an Access Control List Protection Mechanism," given at:
Institute for Advanced Computation, Sunnyvale, California, 6/11/75
Stanford Research Institute, Menlo Park, California, 6/13/75

Schroeder, M.D., "System Certification Research at Project MAC," given at:
I.L.P. Symposium on Information Privacy and Computer Security, 9/24/74
Stanford Research Institute, Menlo Park, California, 11/11/74
University of Utah, 11/15/74

Committee Memberships

Kanodia, R.K., National Software Works Working Group

Wells, D.M., National Software Works Working Group

Saltzer, J.H., ARPA Secure Systems Working Group

Thomas, E.L., ACM SIGGRAPH Graphics Standard Planning Committee

Network RFC's

Kanodia, R.K., "Performance Improvement in ARPANET File Transfers from Multics," Network RFC 662, November 26, 1974.

Kanodia, R.K., "A Lost Message Detection and Recovery Protocol," Network RFC 663, November 29, 1974.

CSR PERSONNEL LIST, JULY, 1974 - JUNE, 1975

Faculty and Research Associate

Professor J. H. Saltzer, Division Head
Professor M. D. Schroeder
Professor D. D. Redell
Professor F. J. Corbato
Professor E. Wada (visiting)
Dr. D. D. Clark

Programming Staff

N. C. Federman
R. K. Kanodia
R. F. Mabee
K. T. Pogran
E. L. Thomas
D. M. Wells

Graduate Students

A. J. Benjamin
R. H. Bratt
R. J. Feiertag
H. C. Forsdick
H. J. Goldberg
A. R. Huber
D. H. Hunt
P. A. Janson
S. T. Kent
A. W. Luniewski
A. H. Mason
W. A. Montgomery
D. P. Reed
V. L. Voydock

Support Staff

S. M. Clark
P. G. Heinmiller
C. Sarner
M. F. Webber

Undergraduate Students

D. Anderson
J. K. Avery
F. Dowla
J. L. Fenton
D. L. Gifford
A. G. Gottlieb
B. M. Grant
E. H. Michelman
D. A. Moon
G. J. Rudisin
S. A. Swernofsky
J. B. Williams

Guests

Professor R. S. Fabry, University of California at Berkeley
Professor A. K. Jones, Carnegie-Mellon University
W. Maczko, Honeywell Information Systems Inc.
Dr. R. M. Needham, Cambridge University, England
Dr. R. H. D. Walker, Cambridge University, England