MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Laboratory for Computer Science
(Formerly Project MAC)

Computer Systems Research Division

Proposal for

Continuation of Research on

Engineering of a Computer System for which Security can be Certified by Auditing

Submitted to

Honeywell Information Systems Inc.

United States Air Force Electronic Systems Division
and
Defense Advanced Research Projects Agency

July, 1976

Jerome H. Saltzer
Principal Investigator

Michael L. Dertouzos
Director, Laboratory of
Computer Science

George H. Dummer
Administrative Director
Division of Sponsored Research

SUMMARY OF PROPOSAL

The Computer Systems Research Division of the M.I.T. Laboratory for Computer Science (formerly Project MAC) proposes the continuation of its research program to make possible the certification, by auditing, of the centrally protected core of a general-purpose, remote-accessed, multi-user computer system. The current contract between HISI and M.I.T. (actually a directed subcontract of a cost-sharing contract between AF/ESD and HISI) provides funding through June, 1976. Progress during the last 12 months in file system simplification has been largely as expected, so it is proposed here that the original work planned for the next 15 months be continued. A specific budget for July, 1976 through September, 1977 is provided.

Overview

The Computer Systems Research Division of the M.I.T. Laboratory for Computer Science (formerly Project MAC) proposes to continue its work on the engineering of a computer system for which security can be certified. The overall project has made considerable progress in understanding how to construct simpler, better-structured file management systems during the current contract period, with several important experimental projects and conceptual tasks either completed or nearing completion. During the next contract period, the Division proposes to complete the experimental work on the file system area, and to continue explanation and technology transfer of what has been learned there. The remainder of this proposal briefly reviews the recently completed activities, and provides a proposed budget. This document is not self-contained; it presumes that the reader is familiar with the original proposal (October, 1973) and the first continuation proposal (April, 1975). The technical content of these two proposals is also summarized in a paper by M. D. Schroeder, which for reference is attached as Appendix I.

Accomplishments so far: System Simplification

Several activities have been completed or are almost ready for a final writeup. Together, these activities represent significant progress on the problem of designing simpler, more easily certifiable, computer systems. These activities are:

1. Accomplishing dynamic inter-program binding without privilege. This activity, reported in MAC-TR-132 by Janson, demonstrated that efficient interprogram binding could be accomplished without special supervisor privileges. By this single change the supervisor interface was simplified and reduced in scope by more than 10%.

2. Accomplishing user name-space management without privilege. This activity, reported in MAC-TR-156 by Bratt, demonstrated that user name

space management can be efficiently handled without special supervisor privilege. In addition to reducing the size of the protected supervisor, this change also demonstrated a way of eliminating the need for the protected file system to manage variable-length character string names (path names) for objects.

3. Demonstration of a two-layer process implementation. This activity, to be reported in an S.M. thesis by Reed (and later in a technical report,) is establishing the feasibility of the following idea: by doing process implementation twice, first producing a fixed number of virtual processes, then later the dynamically varying number needed by users, it appears possible to disentangle the process implementation from the virtual memory implementation. The primary question remaining is whether this separation can be accomplished without excessive overhead; the demonstration will help answer that question.

4. Demonstration of a multiprocess virtual memory implementation. This activity, to be reported in an S. M. thesis by Huber (and later in a technical report,) has shown that separation of the management of virtual memory into a process responsible for removing primary memory pages and another responsible for removing backup memory pages does considerably simplify the structure of that area of the system. Further work is underway to verify the hypothesis that the initial, fairly clumsy, PL/I implementation can be tuned up to have performance equivalent to the older, machine language version.

5. Demonstration of an alternative system initialization strategy. This activity, to be reported in an S. M. thesis by Luniewski (and later in a technical report,) intends to demonstrate that pre-initializing the supervisor before generating a system tape rather than initializing it on the fly at each system start is a simpler, easier-to-comprehend strategy.

6. Rearrangement of process initialization/domain crossing strategy. This activity, to be reported in an S. M. thesis by Montgomery (and later in a technical report,) intends to demonstrate that a significant reduction in the quantity of protected, privileged code can be accomplished by making

process creation unprivileged. The standard mechanism to control entry to a domain can be used to force new processes to start at a controlled location. This strategy, in turn, allows login of users to be accomplished by an unprivileged creation of a process in the potential user's domain, followed by authentication of the user by an unprivileged initial program in that domain.

7. A new model of process coordination. This activity, to be reported in a journal paper by Kanodia and Reed, represents a breakthrougn in thinking about coordination of cooperating sequential processes. By use of eventcounts (rather than semaphores) problems that require only that things be done in a predictable sequence can be handled without using more powerful mutual exclusion machinery. One result of this work is that it is possible for a reader of a database to synchronize its use with writers, while being completely unable to signal any writer; this ability has significant implications for multilevel security situations that require confinement. Another result of interest is that the eventcount model allows correct synchronization of processes separated geographically; this ability is important for network environments.

8. Partially ordered systems. This activity, to be reported in a Ph.D. thesis by Feiertag (and by a technical report,) is a study of the advantages of allowing partial ordering of modules when proving correctness of large systems. (The approaches suggested by other workers, such as those of SRI, allow partial ordering but do not take advantage of it.) The basic strategy required to take advantage of partial ordering is to provide a way to specify, for each module, exactly what environment the compiler should assume while compiling programs of that module. The goal of this thesis is to reduce the otherwise combinatorial build-up of effort required to prove correctness of multi-module systems.

9. Type abstraction in the design of the virtual memory manager. This activity, to be reported in a Ph.D. thesis by Janson (and a technical report,) demonstrates that the concept of type abstraction can be

successfully applied to the internal structure of the virtual memory manager that underlies the usual mechanisms for providing type abstraction. Most systems employing type abstraction as a simplifying concept have not attempted to utilize the concept in their own implementation, because the correct form of separation is not obvious. This thesis demonstrates that at least one self-consistent abstract layering of a virtual memory implementation exists. The implication is profound, since the virtual memory subsystem is the single most complex subsystem in Multics and it can now be broken apart into independently verifiable modules.

Proposed Continuation

The project to engineer a simpler kernel for Multics is close to completion, and we propose to finish the remaining research tasks in the next twelve months. Many of the tasks will be completed well before the end of that period, and the proposed budget reflects an anticipated tapering down of this activity. Two new tasks have been added, one that covers the writing of a final report on the project and the second to provide for continued "technology transfer" support of HISI and others who may wish to make use of our findings. All of the proposed tasks are described in detail below:

1. Continuing Research Tasks

1.1) Restructuring of Traffic Control.

This continuing project was described earlier, in the progress report. D. Reed, as his master's thesis research, has designed a restructured 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 of 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.

In addition, R. Kanodia and D. Reed have prepared a paper describing a new mechanism that they have developed for the coordination of processes. This new coordination mechanism, which uses monotonically increasing counters to record event occurences, 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 coordination 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 coordination situations very naturally.

The eventcount process synchronization mechanism seems to hold the potential for significant simplification in the area of interprocess coordination. The paper being prepared, a draft of which was published as RFC 102, discusses in some detail the theoretical benefits to be derived from eventcounts. The practical advantage of eventcounts to the Multics system can best be seen by an examination of the way they are used in the proposed implementation of the two level traffic controller, where it will be seen that the use of eventcounts has encouraged the exact sort of structure for which we are striving.

Two activities remain to be done, the completion of the detailed design and implementation study currently under way, and the completion of a paper describing eventcounts.

The design and implementation study, which was started in the beginning of 1976, is required in order to demonstrate that the two-level traffic controller has no adverse performance effects on the system. This performance issue is not addressed in the thesis, which discusses the theoretical justification for this structure and demonstrates the simplification which will result from its use. However, it is very important that the efficiency issue be addressed, since multi-level schedulers have been proposed before (although not with the goal of simplification in mind) and these two level schedulers have not operated with reasonable delay properties. We feel that the problems causing these delay properties have been side-stepped in the current design, but it is appropriate and important to confirm this.

By the start of this contract, the technical report describing the theoretical advantages of the structure should be finished, as should the

draft of the detailed design specification. The next step will be to determine exactly what sort of implementation of this design we propose to do. An implementation schedule will be constructed after this decision is made. Following the implementation, a document will be prepared which discusses the conclusions we draw from this implementation.

The publishable version of the paper on eventcounts will be prepared.

1.2) Separating of Page Control and Segment Control Functions within the Active Segment Table.

Currently, the Active Segment Table is not properly organized for functional modularity, in that it contains variables used both by page control and segment control, causing an unstructured interaction between these two systems. This task is to propose a particular interface between these two systems that will result in a system that is simpler, more structured, and easier to understand. Currently, we are completing a detailed investigation of all the current sorts of interactions between the two systems, particularly as manifested by the variables in the Active Segment Table. The next step is to try to determine what the correct sorts of interactions should be. In other words, we must specify some appropriate set of dependencies that can exist between segment control and page control. It is likely, as we propose and justify such a specification, that the work now being completed by Phil Janson on a theoretical organization of the virtual memory mechanism will be applicable. An example of the sort of problem that must be solved is the question of whether the functions of quota management should be assigned to segment control or page control.

The current project is to identify all the modes of interaction currently existing between Segment Control and Page Control. Using the understanding and insight derived from this study, the next step is to propose what the correct sorts of interaction should be. The result of this should be a specification for the interface between Segment Control and Page Control. Having this specification, it will be possible to determine what sort of implementation is appropriate in order to demonstrate the validity of the specification. It may be necessary to iterate on the interface specification, if it is not possible to discover all modes of interaction except by

experiment. The final part of the task is a preparation of the technical report (thesis).

1.3) Study of Multics System Initialization.

A. Luniewski is continuing, as his master's thesis, the definition of a methodology for the initialization of the Multics system which is simpler, and more structured than the current strategy. The development of this methodology is considered very important, not only to help systematize a traditionally adhore part of most systems, but also because all techniques for validating system security depend on being able to validate a secure initial state.

The work in this task so far has resulted in the description of an alternative mechanism for system initialization. We have performed certain selected implementations, to demonstrate that the ideas proposed in this thesis are indeed viable. The current project is to determine, by analysis and further implementation, to what extent this alternative structure is easier to understand than the one currently being used. It is not yet clear whether the alternative structure being proposed in this research will be of sufficient benefit to Project Guardian that the reimplementation of the system ought to be performed, but we feel that the research is valuable in any case, for the following reason. The programs that now perform initialization are not structured around any particular model of initialization, but are instead just put together in a way that works. As might be expected, current programs do display, a posteriori, some underlying structure, and this structure must be understood by anyone attempting to certify these programs. This study, in comparing the current structure with an alternative, may well yield insight to the certifier that will assist him in understanding initialization, whichever strategy he is called upon to certify.

One or two trial implementations of a hardware reconfiguration will be performed in order to demonstrate the validity of certain proposals made in this study. Certain other implementations, or at least preliminary coding, may be proposed in order to demonstrate the superiority of this strategy over the one currently used. The final part of this project is the preparation of the technical report.

1.4) Provision of "Breakproof" Environment for UserProgramming.

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. H. 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 of the LOT. His thesis will provide a practical solution to the problem, will demonstrate the utility of rings for user self-protection, and will contribute to our understanding of the correct functionality of the user programming environment.

The implementation of the proposed environment now under way must be completed. The completed user environment must be evaluated by use, in order to ascertain that the proposals in this study are indeed correct. Following this, the final report must be prepared.

1.5) Restructuring of Page Control.

As described in last year's annual report, the goal of this research was to utilize several asynchronous parallel processes to perform the various functions of page control. In particular, separate processes were used to remove pages from memory and from the paging device so that a free storage pool would always exist to be used for the servicing of page faults. This research has demonstrated that the use of parallel processes in this context provides an intrinsic simplification to the algorithm. The Master's thesis by A. Huber describing this implementation has been completed and will be available soon as a technical report.

The only part of this task remaining is the preparation of a report discussing the performance implications of the multi-process Page Control. It is expected that all experiments required in order to write this report will be completed early in the contracting period.

1.6) Support of User Defined Object Types.

D. Hunt is continuing work on his EE thesis entitled "Building Blocks for

an Object-Oriented Virtual Memory System". The thesis deals with modular decomposition of large systems as a technique to make verification possible. In particular, the thesis focuses on the inter-module relationships which may exist. It is claimed that just a few relationships, each with simple semantics, are sufficient to express inter-module dependencies in a large number of cases. To support this claim, a virtual memory subsystem--similar to a Multics virtual memory enhanced to support type extension--is considered as a case study. A modularization for the virtual memory is suggested and justified, and the inter-module relationships are specified. There will be a report describing the work done on this project to date.

1.7) Study of System Reliability and Recovery from Errors.

This past quarter H. Forsdick has developed a model of the occurance and handling of errors in a computer system. In terms of this model, he has started to look at the way errors are handled in the current implementation of Multics. He has traced down the missing segment fault path in an attempt to develop a characterization of the structure of Multics with respect to errors. Several insights on the placement of error checks have come out of this study. A report will be written describing the work done on this project to date, but further work is not anticipated.

1.8) Organization of the Virtual Memory Mechanism of a Computer System.

P. Janson has completed the writing of the first draft of his Ph.D. thesis on a method for producing modular, structured software to support the virtual memory mechanism of a computer system. The first part of the thesis describes the type extension concept that is recommended as the basis for organizing a virtual memory mechanism. A virtual memory mechanism should be regarded as implementing abstract information containers (e.g. segments) out of physical information containers (e.g. core blocks and disk records). The second part of the thesis explains how that type extension concept can be exploited to organize a virtual memory mechanism. In particular, it shows how one can implement the programs and the address space of the mechanism itself without violating modularity and structure. The last part of the thesis illustrates the use of the method by applying it to the redesign of the

virtual memory mechanism of Multics. The modularity and the structure of the resulting system are evaluated.

The work remaining on this task consists of publishing the final draft of the thesis and publishing the corresponding technical report.

2. New Tasks.

One new task area, not previously explored, is proposed. This task area involves analysis of input and output operations, especially to multiplexed channels such as a channel to a local or long haul network or a channel to a terminal-concentrating front-end computer system. The purpose of this analysis is to understand the impact of multiplexed channels on security requirements of a host, and on the complexity of the security kernel. In particular, those parts of the host software that perform connection setup, channel buffering, and multiplexing fan-in of output and fan-out of input apparently are security relevant. These functions are complicated by error handling, use of encryption, and host-to-host or concentrator to host messages. The ARPANET and the Multics to front-end processor interfaces are expected to be the vehicle for this study. Two specific tasks are currently anticipated.

2.1 Experimentally investigate alternative software protocols for the connections between the Multics host and multiplexed input/output channels (for example to the proposed secure front-end processor and to the ARPA network) to identify techniques that minimize the complexity of the security kernel of the host.

2.2 Experimentally investigate the division of function between the Multics host and a front-end processor for the case where the front-end processor acts not only as an I/O stream concentrator but also as an interface to a local or long-haul data communications network. The division of function is to explicitly take into account the impact on the complexity of security kernels not only in the host but also in the front-end processor. It is intended that this task develop techniques that can be transferred into the environment of the hardware and kernel of the proposed secure front-end processor.

3. Technology Transfer Tasks.

3.1) Continuing Support Activity.

This task covers the activities needed to communicate the insights and understanding gained over the course of the project to Honeywell and to other organizations creating certifiably secure systems. It involves writing of internal memoranda and publishable papers discussing results from different points of view. It also includes attendence at meetings and discussion and review of ways in which the results of the research projects can be applied to the Multics kernel design.

3.2) Final Report.

A retrospective final report will review all of the engineering proposals made or tested in the course of the kernel design project, to assess their cumulative impact on the size, performance, structure, and simplicity of the kernel required to support security requirements. It is intended that a version of this report will be a publishable paper.

Coordination with HISI Project Guardian

Over the past twelve months the style of interaction and coordination with Honeywell has evolved, under a variety of pressures. While early projects in the file system area were carried by M.I.T. to the stage that they could be installed in a production system, more recently the Honeywell teams have had more time and personnel available to review the conceptual issues involved in securing a file management system. In addition, Honeywell has contracted with a team at SRI, which has begun to advise Honeywell on the reimplementation work required to achieve verifiability. For these reasons, it seems neither appropriate nor desirable that current engineering experiments within the file system area be carried to the same state of production readiness attained by earlier experiments. The proposed work force levels and budget reflect this change in style.

Budget

During the last twelve months, one new faculty member (Liba Svobodova) has joined the Computer Systems Research Division and two current faculty members (Michael Schroeder and David Redell) have announced their plans to leave the Division. Since the rate of work achievable depends on the amount of faculty supervision available, the proposed budget level for salaries and other expenses for the coming twelve months is somewhat lower than that of the previous eighteen months.

The proposed budget includes a plan to taper down to a level that supports cleanup work and technology transfer.

Because computer time may be directly furnished by the government or by Honeywell, it is separately identified.

Indirect costs are calculated at the following rates:

Faculty, Research, and Support Staff:

benefits = 24.5% of salary

overhead = 68% of salary and benefits

Students

overhead = 68% of 50% of salary

Budget Summary, by quarters

(K = $1000)

	Salaries and Expenses	Interactive Multics Service	CISL Multics time
7/76 - 9/76	$ 72 K	$ 31 K	$ 13 K
10/76 - 12/76	76 K	34 K	14 K
1/77 - 3/77	41 K	19 K	0
4/77 - 6/77	32 K	13 K	5 K
7/77 - 9/77	32 K	13 K	5 K
Totals	$253 K	$110 K	$37 K
		Total request:	$400 K

Technical Personnel

Faculty and Research Associates

Jerome H. Saltzer, Professor (Principal Investigator)
Liba Svobodova, Assistant Professor
David D. Clark, Research Associate

Research Staff

Nancy Federman
Rajendra Kanodia
Robert Mabee
Kenneth Pogran
Douglas Wells

Graduate Students

Richard Feiertag
Eugene Ciccarelli
Harry Forsdick
Robert Frankston
Harold Goldberg
Douglas Hunt
Philippe Janson
Steven Kent
Allen Luniewski
Andrew Mason
Warren Montgomery
David Reed
+ 2 new graduate students

Undergraduate Students

to be assigned

DRAFT 4/29/76

Budget details, 7/1/76 - 6/30/77

Salaries:

	Head Count	Equivalent man-months	Direct Cost	Direct plus Indirect Cost	
Faculty and Research Associate	3	14	$24,794	$ 51,653	
Research Staff	5	30	34,967	72,844	
Graduate Students	11	69	68,360	91,602	
Undergraduate Students	3	12	3,000	4,020	
Support Staff	2	10	7,200	14,999	
Total salaries				$235,118	$235,118

Other Operational Expenses:

10 terminals and communication equipment at $162/mo.	$ 19,440	
Travel (10 trips at $500)	5,000	
Reproduction (120,000 sides at $.042/side)	5,040	
Telephone ($260/mo. basic service and $150/mo. tolls and message units)	4,920	
Miscellaneous (alterations, maintenance and repair of equipment, office and lab supplies, books and reports, page charges, postage and shipping)	4,300	
	$ 38,700	$ 38,700
Total salaries and operations, 7/1/76 - 6/30/77		$273,818

Computer Time:

Multics service time		
(storage: 72000 page/months at .60/pm)	43,200	
(dial up time: 10000 hrs at $7.50/hr)	75,000	$118,200
Multics development time		
(110 system hrs at $615/hr)	67,650	67,650

Capital expenditures for terminal/network/host interconnection experiments:

HISI Level 6/xx with 128K memory and 16 terminal ports	60,000*	
Special interface hardware for local network	10,000*	
High performance microprogrammable terminal	5,000*	
Total capital expenditures	$75,000*	$ 75,000
Total Requested		$534,668

* estimated; exact prices not available.