ATTACHMENT 2

Statement of Work -- Scope of Effort

for Period Extended from 01 July 1976 through 30 June 1977

Reduction of Hardcore to Permit Certification

The subcontractor shall complete this effort. Technical reports from efforts under these tasks shall be prepared as appropriate. A final report shall be prepared by the principle investigator (MIT) integrating and interpreting the results of their complete effort.

The following tasks define the efforts which are in process and are planned for completion during this period of performance.

Separation of Page Control and Segment Control Functions within the Active Segment Table (AST)

Currently, the Active Segment Table, a central data base in Multics, is not specifically organized for functional modularity, in that it contains two distinct types of information. The AST contains page table words, needed by Page Control, and other variables which describe per segment attributes rather than attributes of individual pages. These latter variables are used by both segment control and page control and causes somewhat needless interaction between the two subsystems.

This subtask will shown that it is possible to partition these variables in such a way that, in every case, they are used by either page control or segment control, but not both. This separation will lead to a significant simplification in the algorithm supported by page control. In particular, it is possible that a thorough separation of page control and segment control in this way would lead to a further simplification in the locking strategies required during the handling of a page fault and a segment fault.

Nearly all the modes of interaction currently existing between Segment Control and Page Control have been identified. Using the understanding and insight derived from this study, the correct type of interaction will be identified. The result of this study will be a definition of the interface between segment control and page control. This definition will provide the specific implementation that is appropriate to demonstrate the validity of the definition. This definition will be revised if experimentation shows that additional modes of the page control/segment control interaction still exist. This task will then be completed with the preparation of a final technical note which will include results of the investigation and the performance aspects of the new implementation.

Page Control Restructure

This research task is utilizing several asynchronous parallel processes to perform the various functions of page control. In particular, separate processes were used to remove pages from Main Memory and from the paging device so that a minimum free storage pool would always exist to be available for servicing page faults. This task has demonstrated that the use of parallel processes in this context provides an intrinsic simplification to the algorithm.

The theoretical design and trial implementation has been documented in a technical report. The remainder of this task will be the preparation of a technical note which discusses the performance implications of this multi-process page control. This technical note will be prepared when the corresponding experimentation, currently in progress, is completed.

Restructuring of Traffic Control

This subtask would develop the design for a restructured traffic control that partitions its functions into two levels. The intention is to split the traffic controller by separating the actual act of switching from one process to another from the more complex act of deciding which process is eligible to run. The lower level multiplexes the real processors of the system among a fixed number of so called virtual processors. By establishing in advance the number of such virtual processors this lower level processor multiplexer need make no use of the system's virtual memory facilities. A higher level scheduler multiplexes some of the virtual processors among all of the currently operating real Multics processors. This higher level scheduler can use all of the facilities of the Multics virtual memory since they are implemented at a lower level. This should clarify the relationship between Traffic Control and Page Control and also aid in separating the idea of interprocess signalling from the idea of Traffic Control. This task would complete the detailed design and implementation study and discuss the conclusions that were drawn from this implementation including performance evaluation.

Study of System Reliability and Recovery from System Errors

This study is developing a model of the occurrence and handling of system errors in Multics. In terms of this model, the missing-segment fault path is being traced to develop a characterization of the structure of Multics with respect to errors.

A technical note which identifies the issues studied to date will be written.

Study of Multics System Initialization

This task is defining 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 complex area of Multics, but also to identify a technique which validated the secure initial state of Multics.

With the design of this task now complete, trial implementations of a hardware reconfiguration will be performed to demonstrate the validity of some of the proposed designs made in this study. Preliminary coding of other implementations will be undertaken to demonstrate the gains made by this new strategy over the one currently used. This task will conclude with a technical note which identifies both the design issues and the experimentation which was undertaken.

Support of User Defined Object Types

This task deals with the modular decomposition of Multics as a technique to simplify certification. In particular, this study focuses on the intermodule relationships which may exist. It is presently believed that just a few relationships, each with a simple explanation, are sufficient to express intermodule 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. The intermodule relationships will be specified and a modularization of the Multics virtual memory will be identified.

The technical note describing this study is now being prepared. The completion of this note will complete this task.

Provisions of Breakproof Environment for User Programming

As various parts of the operating environment are removed from the Multics hardcore supervisor, the question arises as to where they should be relocated. If they are placed in the same ring as the user's executing programs, then they are subject to inadvertent destruction by a user's programming error. It is very desirable that removal of software from the Multics supervisor does not lead to a reduced robustness of the programming environment. A consistent collection of user support programs that can be protected from destruction by casual error of the user is now being defined. This collection might include, for example, the linker and the linker's data bases: the linkage sections and the Linkage Offset Table (LOT). This task will provide a practical solution to the problem and will further demonstrate the utility of the Multics ring structure for self protection of user software.

The implementation of the selected environment is now underway and will be completed. When completed and evaluated by experimentation, the techniques selected by this study can be verified. A final technical note describing the study, the design, implementation and experimentation will be prepared.