Reed

To: Distribution

From: T. H. Van Vleck

Date: 10/22/74

Subject: New Storage System Implementation Plans

This memorandum gives a brief overview of the implementation plans for the new Multics storage system.

PHASES OF IMPLEMENTATION

One possible approach to implementing the new storage system would be to make all the programs or program changes described in MTB-110, put them all together into a system, and try to bootload it in order to see what had been left out. Since other development work and bug fixing will be going on in the storage system area during the time that the new system is being developed, a serious coordination problem might arise. In addition, such a strategy not only postpones the detection of possible design and performance problems, it also defers the anticipated reliability and performance benefits of the new system until all features have been checked out. It seems clear that a strategy which leads to implementation of the new storage system in several phases could be constructed, and that such a strategy might help us to implement and install the new system with a minimum of surprises.

The major benchmarks which have been identified so far are:

1. Getting the system to command level.

   Until we reach this stage, we cannot perform any convincing performance tests, nor can we be certain that our data structures are adequate. We can reach this benchmark, however, without having to do the whole job; in particular, we can use a dummy version of the directory control locking program, since there will be only one process, and we may be able to bypass implementation of those directory control primitives which are not needed by initialization. In addition, the 64-word I/O facility for the VTOC manager can be simulated by segment IO, at the cost of some large AST entries; and the volume maps can remain wired down, instead of being subject to removal from core.

2. Runnable system.

   This stage implies that all currently-defined interfaces to Multics work. We can reach this stage without being able to support more than one volume group and without being able to mount and dismount volumes. However, benchmarking activities will be possible here, and we will be able to test the new storage system exhaustively to make sure that there are no surprise changes to the user interface.

3. Installable system.

This stage is attained when testing and measurement of the previous stage show that system performance, reliability, and function are superior to the then-current Multics. The current system's backup and retrieval mechanisms will be adequate if replacements have not been made available; but a new salvager must be available for the new system, although an interim salvager may be fairly easy to construct. Mounting and dismounting of volume groups, paging of the free storage maps for each volume, and even handling more than one volume group still need not be available.

There are significant advantages to installing the system, at least at MIT, when it reaches this phase. First, it should provide an improvement in system reliability, system performance, and operational convenience. Second, installing the system at this point "unblocks" other development and bug fixing efforts, which can again begin to share a common supervisor base with the new storage system development group.

In order to convince MIT that a change of this magnitude has been sufficiently checked out, we may have to consider running a "mini-service" on the CISL development machine, in order to gain practical experience.

4. More than one volume group.

This stage allows more than one volume group. The quota mechanism must be changed to take account of the new rules for quota manipulation; system commands must be modified, and new commands provided, to manipulate the new location attribute for segments and directories; and administrative and billing programs must be changed to handle more than one volume group. It would be possible, if the device registration package is not ready, to postpone the dynamic mounting and dismounting of volumes, and have this function performed only when the system is shut down. The 64-word I/O facility will be needed by this time, if it has not been already completed in order to accomplish our performance goals.

5. Full system.

At this level we must have all of the functions described in MTB-110. Paging of device maps will no doubt be the function which is the trickiest; but smoothing out all the rough edges of the volume mounting path is the one which will require the most iterations.

WORK TO BE DONE

The first steps toward the new storage system are already underway. We are starting to generate a "specification" for various parts of the implementation which seem to require advance documentation.

One thing we are doing is to list all the system include files which will be changed or superseded. The include-file cross reference can then be mined for a list of programs which are affected, and this list can be separated into groups of programs arranged by difficulty of modification. For example, many programs include the declaration of some ring-zero data bases simply because they extract statistics from the hardcore. Many of these programs need only a recompilation for the new storage system.

The modifications necessary to page control arise mostly from the new format of the AST and PTW. This project interacts strongly with the new core removal algorithm project now being pursued.

Segment control is the most heavily changed. The obtaining of AST entries and the initialization of them is almost completely redone. The new locking scheme, new AST format, and the introduction of the VTOC entry are the most salient features of this part of the job. This is the area which has had the most attention during the system design. The vtoc_manager_ program is central to the segment control changes. This program must be written completely from scratch.

The changes to directory control involve the greatest number of programs. Since the format of the directory branch changes and since some data items formerly in the branch must now be obtained from the VTOC entry or the ASTE, every module of directory control will have to be examined. It is hoped that most of the changes will be routine editing.

System initialization will probably still be our most significant test. Because the system uses itself for initialization, the completion of initialization verifies that most of the central mechanisms of the supervisor are working. The program initialize_dims is the most interesting part of this task; in the current multics, initialize_dims is the program which takes the first page fault.

The disk DIMs should be reorganized so that the device-dependent information is carefully contained. Error recovery should be improved, to the point that the system can attempt to continue if a disk drive goes bad; it would be nice if the operator could request the system to switch a pack to an alternate drive under certain circumstances.

The salvager will be almost compeletely rewritten. As mentioned in MTB-110, we envision three different salvagers, each invoked to handle a different kind of problem. Much of the code from the current salvager can be used in the construction of one or the other of these salvagers.