Peach

To: Distribution

From: T. H. Van Vleck

Date: October 1, 1975

Subject: New Storage System Long Range Plans (revised)

INTRODUCTION

This document supersedes MTR-081. We are well ahead of schedule on the main line of the project; further thought about backup and error recovery has led to important changes in these schedules.

OVERVIEW

The following table shows the major phases of the implementation of the new storage system.

Phase		Date
I	Command Level One user at command level	May 75
II	Prototype Running Several users	June 75
III	Design Review Error recovery, backup, mount/demount	Oct 75
IV	Installable System Run mini-service at CISL	Nov 75
V	Initial Installation at MIT One logical vol, no mount/demount	Feb 76
VI	Follow-up Installation at MIT Operational enhancements	March 76
VII	MR 4.0 Installed at MIT With mount/demount	April 76

VIII Release MR 4.0 June 76

IX Further Enhancements
Very large configurations
Administrative improvements

CHANGES SINCE LAST REPORT

Since MTR-081 was published, we have made significant progress. Phases I and II were finished ahead of schedule. The date for Phase III has been slipped slightly, to accomodate new ideas about error recovery; we still hope to accomplish the other phases on or ahead of schedule.

The significant events since March are summarized below.

1. Operational system. As mentioned above, we were able to get the system to command level, reload the libraries, execute the normal accounting setup steps, and log in users and daemons.

2. Preliminary performance tests on the new system suggest that its performance is about as good as that of the current system. This is an extremely pleasant surprise. Further analysis of the performance data is being carried out so that we can find out why, and how to improve performance still further.

3. Our backup designs are now becoming more concrete, and it appears that we will be able to adapt the current system's backup subsystem for use with the new storage system. This strategy defers some of the benefits of the new storage system changes, but decreases the uncertainty about schedules.

4. We underwent a similar learning process with the salvager design, and as a result have chosen to implement a straighforward adaptation of the current system's salvager as a first step in a campaign which will eventually lead to vastly better error recovery software.

5. The final plan for salvager-like functions is to make them an integral part of the system, so that all data is checked for validity (and repaired if necessary) at the time that it is used.

6. Finally, our designs for the dynamic mounting of volumes have solidified into a set of steps leading to an extremely clean implementation.

WORK COMPLETED

1. Statement of Problem.

   MTB-017, November 1973.

2. Preliminary Design.

   MTB-055, April 1974
   MTB-060, May 1974
   MTB-065, April 1974
   MTB-095, June 1974
   MTB-110, August 1974
   MTB-167, February 1975
   MTB-203, June 1975
   MTB-206, June 1975
   MTB-213, July 1975
   MTB-220, September 1975
   MTB-221, September 1975

3. Preliminary Task Schedules.

   MTR-068, October 1974
   MTR-081, March 1975
   MTR-084, April 1975
   MTR-094, Sept 1975

4. Phase I: Command Level.

   When this benchmark is reached, the system can be bootloaded from a Multics system tape, either cold or warm, come up to Initializer command level, and shut down. Only one disk need be used; but it will have a standard label, VTOC, and volume map. Paged I/O will be used for the VTOC. A new version of BOS is required to support the new configuration deck.

   Target date: May 1975. Finished.

5. Phase II: Demonstrable System.

   When this benchmark is reached all functions of the current Multics work in the new system, with the exception of minor bugs and certain metering tools. Since the VTOC is still accessed by means of paged I/O, 1K per volume of page table for VTOC image is wired, plus the 512 words per volume of volume map. No backup or salvager will be implemented. Although much more interesting in terms of function, this stage is not very difficult to accomplish because the system initialization path checks out almost all of the storage

system.

Target date: June 1975. Finished May 1975.

6. 64-word I/O Facility.

This facility will be used to transport VTOC information and volume map data between core and the disk. Changes must be made to the disk DIM, and a sort of "mini page control" must be written for the management of the memory devoted to 64-word data.

Finished.

7. New VTOC Manager.

The new VTOC manager will use the 64-word I/O facility. This change frees 1K per volume of wired core, and decreases the I/O channel time and latency time for requests for data in the VTOC.

Finished.

8. Smaller VTOC Entry.

The interim VTOC entry in use up to this point has 256 words instead of 192, in order to simplify the code for the deactivation of 256K segments. This stage complicates the code but reduces the size of the VTOC by 25%.

Finished.

9. Study and Definition of Backup Problem.

This activity will define the key variables to be optimized in subsequent backup design. See MTB-203 for details.

Finished.

10. New Directory Locking.

This step creates a wired hardcore table with one entry per active directory. The directory lock is kept in this table. Directories need not be modified to be locked and unlocked as a result of this change; this reduces the paging traffic significantly.

Finished.

11. Interim Version of BOS SAVE and RESTOR

An interim version of BOS SAVE and RESTOR has been written which handles single volumes only. It is now being used to

support performance testing.

Finished.

12. Hardcore Partition.

The system bootloading sequence is altered by this task to use a special area of the root physical volume for all paging needed before directory control initialization, as described in MTB-213.

Finished.

CURRENT TASKS

13. Multics Utilities for Pack Maintenance.

These programs will initialize a disk, set up VTOC entries, volume map, and write and check labels. Prototype versions of these programs are now running, using the rdisk_ DIM and PL/I record I/O.

14. Consistent Directory Locking.

This step modifies the operation of the lock primitive and page control so that modified pages of directories which are locked are not written back to disk until the directory is unlocked. This strategy makes the "window" during which a system crash can leave a directory inconsistent very small. Changes to shutdown are also needed. An MTB will be published on this subject.

15. New Configuration Strategy.

The interim configuration mechanisms used by Multics and BOS will be replaced in this step with a new mechanism which reduces the size of the CONFIG deck and fits well with the volume mounting design. See MTB-213 for details.

16. Performance Measurements.

At this stage performance must meet or exceed the performance of the installed Multics. If it does not, we will have to find out why and fix it. An MTB describing the strategy to be used for performance measurements will be published.

Preliminary comparisons are encouraging; additional VTOC reads and writes seem to be counterbalanced by a decrease in paging.

17. Interim Backup.

This task modifies the current backup programs to dump and reload the logical volume ID for a directory. This change allows the current incremental/catchup/complete dumper to be used for backup until a new version is designed and built.

18. Interim Salvager.

This task creates relatively straightforward adaptation of the current system's salvager. When salvaging is necessary, this subsystem will salvage all volumes and then walk the whole hierarchy, like the current system's salvager. See MTB-221 for a discussion of the salvager plans.

19. Design for Backup.

This design will present the long-term plan for the evolution of the system's backup capabilities. Two modes of data recovery will be required, one for reconstruction of the contents of a complete physical volume (or group of physical volumes), and another mode for retrieval of the contents of a single segment.

20. Design for System Recovery Modes.

A monolithic salvager subsystem becomes more and more unwieldy as the size of the configuration increases. The proper solution is to improve the system's in-line error detection and dynamic salvaging code. The emergency shutdown, backup, crawlout, on-line salvager, and directory locking facilities will be redesigned into a coherent and complete package. See MTB-220 for more information.

21. Design for Volume Mount/Demount.

This task adapts the Resource Control Package facilities used to manage tape reels to the management of logical volumes. Both logical and physical volumes will require registration data which must be consulted before a logical volume can be mounted for a user. An MTB will be written on this subject.

22. Phase III: Design Review.

This review covers the design of the error recovery modes, backup, and the volume mounting and demounting modules.

Target date: October 1975.

23. Dynamic Physical Volume Mounting and Accepting.

For this stage, some of the code in initialization which connects a pack to the storage system will be moved to

permanent residence. Privileged gates from ring 1 will be provided so that volumes can be added. This step will complete the implementation of the new configuration strategy described in MTB-213.

24. Implementation of Hardcore Primitives for Backup.

The hardcore primitives to support the new backup system must be able to maintain the list of modified segments on each physical volume for the use of incremental dumping; and to activate and dump or reload a segment by volume ID and VTOC index without referencing the branch.

25. Implementation of Backup Dumping Programs.

The new complete and incremental dumping programs can be much simpler than the current dump programs, since all hierarchy walking and access forcing code is eliminated. The hardcore primitives do most of the work. These programs are easy given the format of the output records to be produced.

26. Implementation of New Reload and Retrieve.

The reloading and retrieval programs will use the output of the dumping programs to reconstruct volumes and to recover the contents of single segments.

27. Specifications for Command Changes.

Many minor changes will have to be made to the command system. Quite a few of these can be done ahead of time if a document setting forth the standards for system commands and subroutines is published.

For example, it will be possible in the new storage system to have a segment which exists, and to which the user has access, but which is not currently on line. Some commands must treat this case as if the segment did not exist. Others must behave just as if the segment did exist. Still other commands must take a new action in this case: and this action can be programmed before the installation of the storage system.

28. Error and Exception Handling Improvements.

This task provides sensible system action for the cases of "no more VTOC entries on the volume" and "no more pages on the volume." To handle the second case the supervisor must move the segment to another volume in the logical volume which has sufficient room.

29. Improved Directory Format.

This task redesigns the directory to be more easily verified for correctness. All storage system modules which reference the directory must be recompiled with the new declaration. The various redundancy checks are not inserted by this task, though. See MTB-221 and MTB-220 for details.

30. Run Mini-Service at CISL.

Until we actually run a "service" of some sort, we will not know what the performance is really like and what operational improvements are required. Installation at CISL also allows other projects to integrate with the new storage system and decreases the number of changes which must be made to two versions of Multics.

The mini-service can be started without the availability of the new backup or the new salvager. The new directory format will be used if possible in order to provide realistic performance estimates.

Target start date: November 1975.

31. New Directory Salvager

Rewrite salvager to operate on a new expanded directory structure, without reference to the VTOC entry.

32. Directory Control Checking.

This task adds to directory control new code for maintenance of the various redundancy fields added to the directory structure, and appropriate in-line checks and repair operations. MTB-220 describes the details of this change.

33. Phase IV: Make System Installable.

Once system reliability and performance are acceptable, the new storage system is ready to be installed at MIT.

The first version of the new storage system to be installed at MIT will not have all the functional improvements which will be provided with release 4.0. In particular, the final salvager system is not required, and the interim backup may suffice, depending on its performance. The ability for a user to request the mounting of a logical volume will not be present in this version of the system. What will be provided is the reformatting of disk storage and directories and the consequent improvements in reliability.

Target date: January 1976.

34. Formalities of Submission.

This step covers filling out submission forms, auditing of all programs, running final performance runs, fixing last-minute problems, etc.

35. Phase V: First Installation at MIT.

Target date: February 1976.

36. Backup Integration.

This task integrates the new backup mechanisms into the running mini-service and ties backup in with salvaging. Target date: March 1976.

37. Master Directory Operations.

This task adds ring-1 support for operations on master directories. User calls are create, delete and list. The create_dir and delete_dir commands need modification for this case.

The ring-1 programs will use the Logical Volume Registration File (LVRF) and the Master Directory Control Segments (MDCSs). Administrative commands to manage these data bases are necessary. For the volume librarian, we need register, unregister, modify, and list. For the volume administrator we need permit, deny, and list.

38. Ring 1 Volume Mount Module.

When a logical volume is to be mounted, the LVRF must be consulted to find the list of physical volumes to be mounted. Calls must then be made to RCP to mount each of the physical volumes, the volume labels must be checked, and the hardcore must be called to tell it that the volumes are accepted.

39. User Request to Connect Logical Volume.

User requests to connect to a logical volume will be passed through RCP. If the user process is permitted to connect to the logical volume, the hardcore will be informed of the connection, a counter associated with the logical volume will be incremented, and a mount request for all physical volumes will be issued, as described above, if they are not already up. The logical volume will be disconnected when the connection count goes back to zero as a result of users unassigning the resource or logging out, or when the operator forces the unassignment.

40. Hardcore Check on Volume Connection.

The hardcore will be changed by this task to require the connection call from ring 1 before allowing a process to initiate a segment on a demountable volume. (The root logical volume is never demountable and other "public" volumes can be declared not demountable.) This insures that RCP is not bypassed, and makes sure that all programs using segments on removable volumes execute independently of whether some other process has caused a pack to be mounted. The list of demountable physical volumes which the process is connected to will be stored in the PDS or some other per-process data base.

41. Phase VI: Follow-up Installation at MIT.

Operational experience will lead us to make many improvements to the interface and behavior of the storage system. Performance measurements under actual load may also show use where to concentrate our programming effort in order to speed the system up; if these improvements are possible we will install them soon.

Target date: April 1976.

42. Command System Changes.

These changes are the ones specified in paragraph 27. In addition to the changes to handle new error and state conditions, the create_dir command must accept and check the new parameter which specifes the logical volume in which storage will reside, and the list and status commands must be modified to show this attribute.

43. Phase VII: Install MR 4.0 at MIT.

Target date: April 1976.

44. Phase VIII: Release MR 4.0

Target Date: June 1976.

FURTHER ENHANCEMENTS

45. Pageable Volume Maps.

Until this point the volume maps have remained wired in core while the volume is online. The cost of this strategy becomes prohibitive when very large configurations are used. The volume map for a DSU191 volume is 512 words, while the map for a MSU450 will be about 2K. This stage modifies the system to use the 64-word I/O facility to transport volume maps to and from core on demand. Careful design must be used so that performance is not degraded severely.

46. Keep Duplicate Copies of Selected Volumes.

Once this task is completed, crucial volumes in the system can be maintained in duplicate; all modified pages will be written out to both devices. In a configuration which places the secondary copy on a different disk subsystem from the primary copy, the cost of maintaining two copies will be very low.

47. Automatic Use of Secondary Volume on Error.

Once the duplicate copy facility is available, the system can be modified so that when a disk record is unreadable, the system automatically switches to the use of the secondary copy.

48. Disk DIM Error Handling Improvements.

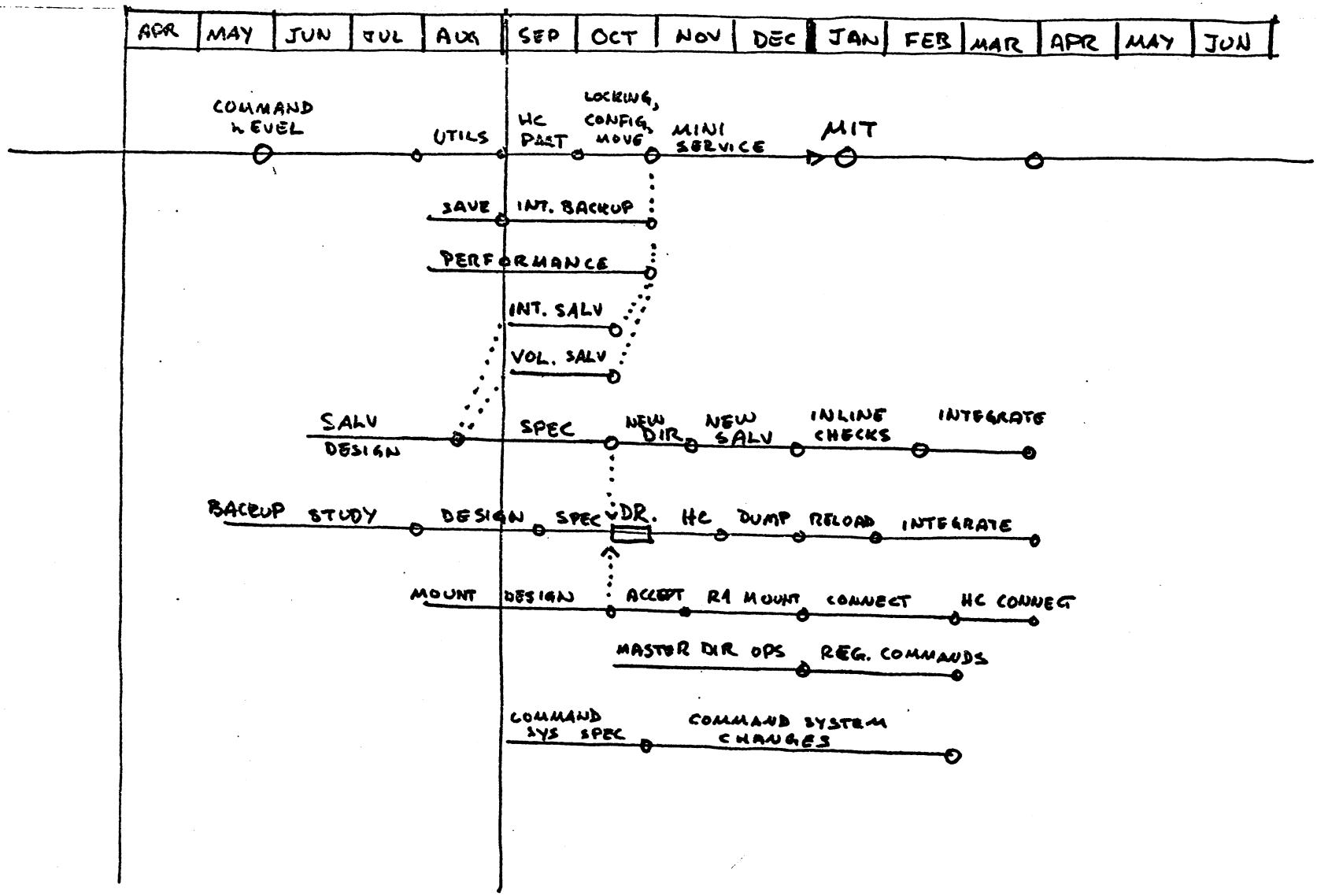
Further improvements are possible to the disk error handling programs. It will be possible for the operator to move a pack which is encountering read errors from one drive to another, without crashing the system.

49. Calls to Initializer Process During Connection.

This step causes RCP to pass all connection requests through the system control process, so that charging can be done, mount messages can be routed, and so that operator commands affecting the request can be issued.

50. Billing.

Modifications must be made to the administrative and billing package to enharce the administrator's ability to manage the system resources. Some of these improvements cannot be specified until we have obtained some operational experience.


APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
JAN
FEB
MAR
APR
MAY
JUN
COMMAND LEVEL
UTILS
HC PART
LOCKING, CONFIG, MOVE
MINI SERVICE
MIT
SAVE
INT. BACKUP
PERFORMANCE
INT. SALV
VOL. SALV
SALV DESIGN
SPEC
NEW DIR
NEW SALV
INLINE CHECKS
INTEGRATE
BACKUP
STUDY
DESIGN
SPEC
DR.
HC
DUMP
RELOAD
INTEGRATE
MOUNT
DESIGN
ACCEPT
RA MOUNT
CONNECT
HC CONNECT
MASTER DIR OPS
REG. COMMANDS
COMMAND SYS SPEC
COMMAND SYSTEM CHANGES