To: Distribution

From: T. H. Van Vleck, Bill Silver

Date: October 8, 1975

Subject: Use of Demountable Logical Volumes

INTRODUCTION

This memorandum discusses the use of demountable logical volumes in the new storage system. MTB-110 (Implementation of Proposed New Storage System) describes how, in the new storage system, the Multics hierarchy is divided into logical volumes, that may consist of part of a physical volume, or several physical volumes. A physical volume may contain storage for only one logical volume. All physical volumes that comprise a logical volume must be mounted or demounted at the same time. Thus a logical volume may not be partially mounted. Logical volumes can be mounted or demounted while the system is running.

A special logical volume, the Root Logical Volume (RLV), cannot be demounted. Except during the booting the system all of the packs that make up the RLV are mounted. The RLV corresponds to the Multics storage system as we know it today. It is in the use of logical volumes other than the RLV that the new storage system provides new user interfaces. This memorandum presents an overview of several new concepts that are involved with the use of demountable logical volumes. Future MTBs will be published that discuss these new concepts in detail. The main new concepts are:

- registration of logical volumes
- master directories
- mounting logical volumes

REGISTRATION OF LOGICAL VOLUMES

The registration of volumes is a concept that has been discussed before, for example in MTB-076 (The Tape Mount Package). The registration of tape reels and disk packs is planned as a future extension to the Resource Control Package (RCP). The use of logical volumes in the new storage system requires that the mechanism for registering these logical volumes be implemented now.

The registration of a logical volume implies that the system maintain data about this logical volume. This data will be kept in a file called the Logical Volume Registration File (LVRF). An entry in the LVRF will be maintained for each logical volume. The LVRF will be located in some permanent system directory that is on the RLV. The LVRF will have ring brackets of 1, 1, 1. Access to the LVRF will be Read for all users and Read-Write for those privileged users that may act as volume librarians.

All of the data needed to maintain the directories on a logical volume and all of the data needed to mount a logical volume will be kept directly or indirectly in the LVRF entry for the logical volume. This data includes:

- A list of the physical packs that comprise this logical volume. Also, for each pack, information needed to mount the pack will be kept.

- The pathname of the Master Directory Control File that is used to maintain the directories on this logical volume. Master directory control files are discussed in the next section.

- The pathname of an Access Control Segment that controls access to the logical volume.

- Default access to this logical volume when no Access Control Segment is available.

- Information about the owner of the logical volume.

The concept of Access Control Segments (ACS) was introduced in MTB-184 (The Resource Control Package). An ACS is a zero length segment whose ACL is used to control access to some resource. In this case the resource is a logical volume. The pathname of the ACS for a logical volume is specified by the owner of the logical volume during volume registration. It can be changed only by volume librarians. The location and ring brackets of the ACS are entirely under the control of the owner of the logical volume. The owner of the logical volume is responsible for creating the ACS, creating any links to it, and setting the ACL to appropriately control access to the logical volume.

Regular ACL commands can be used to set the ACL of a logical volume. The access modes REW will be interpreted in ring 1, however, and they have somewhat different meanings for logical volumes:

- RW (Read/Write) access is needed to mount a logical volume

- E (Executive) access is needed to set user quota on a logical volume.

MASTER DIRECTORIES

In the new storage system all segments contained in the same directory must be placed in the same logical volume. When a directory is created, it normally inherits the "sons_lvid" item from its parent directory. A directory whose "sons_lvid" is different from its parent is called a master directory. Thus a master directory is a directory whose sons reside on a logical volume that is different from the logical volume that holds the segments of the master directory's parent. Master directories provide the mechanism for placing a piece of the hierarchy on a demountable logical volume.

To create a master directory new control arguments must be used with the create_dir command. These arguments specify the name of the logical volume that the master directory will reside on and a quota value. This information is passed to a new ring 1 subsystem called Master Directory Control (MDC). MDC checks access to the logical volume and checks quota on the logical volume before calling ring 0 to actually create the directory. An additional privileged call is made to the hardcore to set the sons_lvid to that of the specified logical volume. In order to create a master directory a user must have the following access:

- SMA access to the parent directory. This, of course, is checked in ring 0.

- RW access to the logical volume on which the master directory will reside. This check is made by MDC in ring 1. This access is obtained from the registration data for this logical volume. If possible it is taken from the ACS for this logical volume. If no ACS is available it is taken from the default access specified in the LVRF entry for this logical volume.

- Sufficient quota on the logical volume for this master directory.

The control of quota in a logical volume is the main function of MDC. Because a disk pack contains so many records, it will be normal for packs to be shared by many users, and for there to be many master directories for a logical volume, each of which is the root of a (possibly large) subtree. The owner of a logical volume will be provided with the resource-control tools to allow him to measure and control the usage of a pack.

The owner of a logical volume has this control by being able the specify which users have "E" access to the logical volume. Users that have "E" access to a logical volume may act as "executives" for the logical volume. An executive for a logical volume can specify the total quota allowed on the logical volume for each user and thus can specify which users may create master directories on the logical volume.

The quota and master directory information for each logical volume is maintained in a file called a Master Directory Control File (MDCF). There is one MDCF for each logical volume. It is created by MDC when the logical volume is registered. MDCFs are created in some permanent system directory that is on that RLV.

Each MDCF contains two sections. The first section is a list of the users that have quota on the logical volume. Both the total quota allowed for a user and the quota that has already been used are kept. These values are checked and updated when a master directory is created. The second section is a list of the master directories that exist on the logical volume. The UID pathname, creator name, and quota is kept for each master directory. Once a directory is created it is almost completely like a non-master directory except that quota moves must be between directories with the same sons_lvid.

When a master directory for a logical volume is deleted, the call is once again intercepted by MDC, which forwards the call to the hardcore. The MDCF second section is updated to show that the master directory has been deleted, and the time-record product for the directory is saved until the end of the accounting period when the master directory entry is deleted. Volume executives may call MDC at any time to list the information about deleted directories. The dynamic usage information is kept in the quota cells for the directories whose sons go on the logical volume, and so a figure accurate to the page-second can be obtained by a program which walks each subtree beginning at a master directory and does not include master directories for other logical volumes. Tools to do this will be constructed from the current system's disk quota accounting programs.

Users will see only a few interface charges. New options for the create_directory command will be provided. Ring 1 entries to MDC for creating, deleting, and listing master directories will be provided. A new error code will be returned for hcs_$quota_move and hcs_$delentry_file if an illegal operation on a master directory is attempted. Finally, a new command will be provided so that a user may interrogate "his" entries in the MDCF for a logical volume, to see if he can create a master directory, etc.

This quota check is made by MDC in ring 1.

MOUNTING LOGICAL VOLUMES

In order to initiate a segment that resides on a logical volume the logical volume, i.e., all of its physical packs, must be mounted. Also, each physical pack must be made useable for paging by being accepted into the PVT via a call the ring 0. MTB-213 (NSS DISK DEFINITION) describes this process in more detail, and explains how permanent volumes are mounted automatically when the system is started up.

For demountable logical volumes it is not feasible to provide demand mounting at segment fault time. Thus the mounting of a logical volume must be the result of an explicit mount request. The basic rule for using logical volumes is:

> In order to initiate a segment that resides on a logical volume a user must have previously issued an explicit request to mount that logical volume.

This rule implies that a logical volume may be mounted for more than one user at a time. A new RCP command and interface will be provided to "mount" logical volumes. This command will mount the same logical volume for multiple users at the same time. When a user mounts a logical volume, the ID of the logical volume is listed in some per-process ring 0 data base (probably the KST). If a user attempts to initiate a segment that resides on a logical volume not listed in his KST, the initiate will fail with a "Logical Volume not Mounted" error. This is true even if the logical volume is already physically mounted and accepted for paging. This rule has two important and useful implications.

- The mounted/dismounted status of a segment becomes deterministic within a process.
- The costs of the resources (disk drives) used by the logical volume can be distributed among all of the user that have concurrently mounted the logical volume.

A logical volume is a new type of resource known to RCP. Logical volumes have characteristics that make dealing with them quite different from device resources. The rule for mounting a logical volume described above implies that the operation of mounting a logical volume is quite different from the operation of mounting a tape. Compare the RCP scenarios for these two types of mounts:

For a tape mount:

- Check to see if the specified tape volume is already mounted for any process. If it is then return with an error.

- Assign a tape device to the requesting process. The user must have access to some appropriate tape device. This access is obtained from the ACS for each device.

- Mount the specified tape volume. Access to the tape reel would be checked and the label verified. These access checking operations are not done now.

- Complete the attachment to the tape device for the requesting process.

For a logical volume mount:

- Check to see if the specified logical volume is already mounted for the requesting process. If it is then return with no error.

- Check to see if the caller has the required access to mount the logical volume. A user must have RW access to the logical volume as specified in the LVRF and ACS for this logical volume.

- Check to see if the logical volume is already mounted for any process. If it is make this logical volume known for this process by calling into ring 0 to list this logical volume in the KST. RCP will add this process to a list of processes that have mounted this logical volume. This successfully completes the mount for this process.

- If the logical volume is not mounted for any process then RCP will mount it. RCP will physically mount each pack that comprises this logical volume. RCP will assign an appropriate disk device for each pack. It will mount the pack on the drive. It will make the pack useable for paging by having it accepted in ring 0. If any packs of the logical volume cannot be mounted then the mount of the logical volume will fail.

Not only is the concept of mounting different for a logical volume, but also the algorithm for assignment of the packs and disk drives used by a logical volume is different from the assignment of a disk drive for use as an I/O disk. The disk drives used by a mounted logical volume are not assigned to any process. They are assigned to the "storage system". The user whose mount request results in the actual mounting of the physical packs of the logical volume is just involved in an

accident of fate. It is the fact that all user requests to mount a logical volume look the same to the user that makes the mounted/dismounted state of a logical volume deterministic.

A new RCP command and interface will be provided to "demount" a logical volume. Demounting a logical volume involves the following steps for RCP:

- Remove the logical volume name from the ring 0 list of logical volumes mounted for this process.
- Remove this process from the list of process that have mounted this logical volume.
- If the demount for the logical volume for this process results in no process having the logical volume mounted then the logical volume will be physically demounted.
- To physically demount a logical volume RCP will call ring 0 to disable paging on all the physical packs of the logical volume. RCP will then unassign all of the disk devices used by this logical volume.

In order to provide complete security for new storage system volumes, a label checking mechanism will be provided during the mounting of both storage system packs and I/O packs. The label of every pack mounted will be checked. For storage system volumes, obviously, the volume label must be completely correct. In addition, in order to prevent a user from mounting an I/O pack and writing a counterfeit label on it, in the hope that some future operator error will cause the pack to be mounted instead of a storage system pack, we will put information in the label of each storage system pack that is only known to RCP. We will also require that any I/O pack which is mounted will have its label record read. If the label looks like a valid label, and the label claims that the volume mounted is not the volume requested, then the volume will not be mounted. This check insures that an operator cannot accidentally mount a storage system pack as an I/O pack; and thus protects the storage system from destruction. Special functions, such as pack initialization, and privileged users, such as the volume librarian, may bypass this check.

The system's charging tools will not charge for storage on demountable packs by the same method used for permanent storage. Permanent storage will continue to be controlled by quota and charged for by the page-second. For demountable storage, the system will charge for the number of minutes that the pack was mounted, just as is done for tapes and for I/O packs. (This facility may not be available in the initial release.) Since storage system packs are shared, the system will charge each of the users that have mounted a logical volume an equal fraction of the total per-minute rate.

Directories for segments on a private pack will have quota and time-page-products just like other directories, but since these data refer to a share of a different resource they cannot be added to the quota or time-page-product data for segments on the system's permanent storage. The owner of a pack will be able to determine the total time-page-product usage of storage on his pack, so that he can "retail" storage space to other users.

The eventual plan for the development of RCP includes a resource-reservation facility that will allow a user to negotiate with the system for future availability of specifiec combinations of resources. Since the number of free disk drives at a site is likely to be relatively small and competition for them severe, we may need to implement interim measures such as time limits on the duration of a mount and special RCP policies in order to permit installations to make the most effective use of their disk drives.