To: Distribution

From: T. H. Van Vleck
Harris A. Berman

Date: 08/30/72

Subject: User Tapes

This memorandum describes a set of changes to the Multics tape-handling facilities which will enable us to allow all users to be able to use tapes. This plan concentrates on volume management and protection, rather than logical data management services; new ways of organizing and using tape data within Multics, such as tape archives and offline segments, will also be required but are not within the scope of this proposal, which is concerned with the elementary system resource management facilities which are a prerequisite for such applications.

Specific features of the implementation described below are:

1. The use of code in ring 1 to intercept attach and detach requests.

2. Ring 1 maintenance of a Tape Reel Description Segment (TRDS), which provides logical protection of each tape reel.

3. An orderly communication path between ring 1 of a tape user's process and the system tape operator, which allows the user to inform the system of tape mount and dismount requests.

4. Tape label checking (and/or authentication code checking), performed by ring 1 of the tape user's process.

5. Tape accounting.

Overview of Tape Management

This section gives a brief overview of the tape management facilities which will be implemented.

Information Protection

Two levels of information protection must be supplied for tape data: logical protection, which defines which users of the system may use the data and how the data may be used, just like the protection provided by the Multics storage system; and physical protection, which arises from the detachable nature of tapes, and which insures that the physical reel being used is indeed the correct one to which the logical protection applies.

Logical protection will be provided primarily by an extended access control list like the standard Multics ACL for a message segment. This access will be managed in ring 1, by special tape-handling primitives. Since some uses of tapes bypass the normal Multics access control mechanisms (such as use by BOS and by the reloader), these stand-alone uses must have some simplified mechanism for access control, to insure that they do not accidentally bypass the system's logical protection mechanism. Tapes may, therefore, also have an accessibility indicator, which will be checked by the stand-alone mechanisms if the TRDS is not available. This indicator will be stored in the TRDS and written on the tape label. The values for this indicator will be:

blank	- User tape. Stand-alone mechanisms will not use.
S	- System tape. For bootloading only.
R	- Reload tape. Backup may write, cold boot may read.
B	- BOS tape. May be read or written.

During normal Multics operation, the accessibility indicator will not be checked, although it will be written into the tape label from the TRDS. The value of the indicator in the TRDS may be changed by a privileged operation. Note that a user may read or write a tape during Multics operation, without regard to the value of the indicator, since his access is controlled by the ACL on the TRDS; note also that only labeled tapes may have a non-blank accessibility indicator.

An optional retention date (in the TRDS) will also be associated with the data on each tape reel, to specify the earliest date that the contents of the tape may be rewritten.

Physical protection will be provided by code which will be installed in ring 1. For Multics standard format tapes, this code will check the tape label of the tape and insure that the tape mounted by the operator is in fact the tape requested. For non-standard tapes, the ring 1 code will check to make sure that the tape either has a correct label, or no label. For tapes which are not labeled, the operator will be required to specify an authentication code when he mounts the tape. He will be told the number of the tape to mount, but not the authentication code; when the tape is mounted, the operator will be required to type in the three-character code (which will be on a sticker on the tape reel). The ring 1 program will reject the tape if the authentication characters are not correct. The authentication code is a non-obvious function of the tape reel number, and can be calculated by the system.

Reel Management

Besides insuring data protection, the tape management facility must provide the facilities necessary to allow installations to manage their inventory of tape reels. This will include administrative procedures for labeling and registering tapes, for obtaining a list of all tapes, or all tapes for a given user, and for producing accounting figures which can charge a user for reel and slot rental and for tape mounts and mount time.

Internal Data Bases

Each reel of tape at a Multics installation will be associated with a ring-1 segment known as a Tape Reel Data Segment (TRDS). These segments will be kept in a special directory in the file system hierarchy. The extended access control list on each TRDS will give the logical access control for the data contained on the tape reel.

The information contained in the TRDS will include the reel name, coded as part of the path name of the segment; the authentication and label-checking switches; the accessibility indicator; and information about the density, recording mode, and number of tracks. Other information may also be useful, such as error count. Most TRDS's will be zero-length segments, to conserve storage; default values for all data which might be contained in a TRDS will be chosen if its length is zero. A data base write-up for the TRDS is attached as an appendix.

It is desirable, for maximum physical protection, for most tapes at an installation to be labeled. The label must contain the tape identification, so that the system can verify that the reel which has been mounted is the same as the one which has been requested. Another memorandum will propose a new tape-labeling scheme which will conform to ANSI standards.

The hardcore I/O Assignment Table (IOAT) will continue to be used to record the association between a process and the channels or devices associated with it.

One additional data base will be provided, for communication between the ring-1 attach/detach programs and the operator process responsible for tape management. This data base will record all mount requests, from the time they are made until the tape is dismounted or the mount rejected, so that a list of all currently-mounted tapes can be extracted from it. The request state, the user's process ID, the tape label, and the time of the request are examples of data in this segment. A data base write-up for the communications segment is attached as an appendix.

User Interface

Users who are now using Multics standard tapes will not be required to change any of their operational procedures, if they are using system-provided tape software. The user will continue to provide, as the third argument to ios_$attach, the tape reel identifier of the tape which he wants mounted. The system will look up the TRDS for the tape in the tape directory, and, if access is correct, issue the mount request. After the tape is mounted, the user may read and write tape just as he does now. When the user detaches a tape device, the system will rewind and unload the tape (usually), and will request the operator to re-file the tape.

Since the authentication code is a function of the tape reel number, unlabeled tapes can be handled in the same manner. That is, when the user requests the attachment of a tape reel which is registered as a unlabeled tape, the system will calculate the authentication code from the reel number, and include the authentication characters in the table entry for the mount request. The operator will be instructed to reply with the tape authentication characters when he mounts the tape, and the mount will be rejected if the authentication characters do not match.

In order to read in tapes from other systems, we may occasionally need to be able to handle tapes which have no TRDS. These functions will be supported by requiring a special syntax in the tape reel identifier, so that a user might request the mount of

```
unregistered,9TRACK,NL,comment="reel 37A5"
```

meaning that he wants an unregistered reel mounted, on a nine-track drive, that the tape has no label, and that the operator should look for a reel labeled 37A5. See the memo on tape identification and labeling for more information on this subject. In order to insure the integrity of the logical protection mechanism, no tape which has a TRDS may ever be mounted by this method. Special privileged intervention by the operator will be required to verify the user's permission to use the tape in this case.

Several other special tape identifiers can be constructed for the convenience of users and system operations. The identifier "scratch" will refer to any available scratch tape. The identifier "next_backup_tape" will refer to the next reel for an incremental dump, and the system will assign a tape and report the reel number used back to the daemon.

Operator Interface

The system control tape management function may execute in the initializer process, or in some other process. Its mount and dismount requests will be sent through the system control terminal management facility to whatever physical console is designated as the destination for mount and dismount messages.

The tape management function will be invoked by event call whenever a user request is made. The event-call message will contain a reference to the communication segment, where the details of the mount request will be found.

The mount request will cause a message to be typed to the operator which looks like this:

```
   1035.1 tape 49 mount reel 275 on drive 3 for Smith.Proj1
   --> tape
```

The number "49" in the message is an internal table index, which the operator will use in his reply. Indexing the messages in this way permits the system to have more than one mount pending at once, while retaining the ability to associate the operator's reply with the proper request. Note that, in general, we cannot have more than one request for each drive on the system pending.

The operator will use the "reply" command to tell the tape function of the success or failure of the mount request.

```
   reply tape 49 ok bvn
```

This example is an affirmative response, in which the operator also typed in the tape authentication characters from the sticker on the tape. Authentication might not have been required, if the tape were a standard Multics labeled tape; the system will check the authentication if it is supplied, for additional redundancy.

The operator will be given a new command called "tape" which can list all the requests in the communication segment if the operator types

```
      tape list
```

or which can print the contents of a TRDS if the operator or user types

```
      tape status 300
```

This command can also be used to register a new tape reel or unregister an old reel, if the operator types

```
      tape register 3701,9TRACK Doe.Multics
```

Suitable defaults will be applied to the arguments to "tape", and the star convention will be interpreted where appropriate. This command may also be used from other processes, if the user has the appropriate access to the ring-1 gate which provides the support for the command. A write-up for the operator commands which will be available is attached as an appendix.

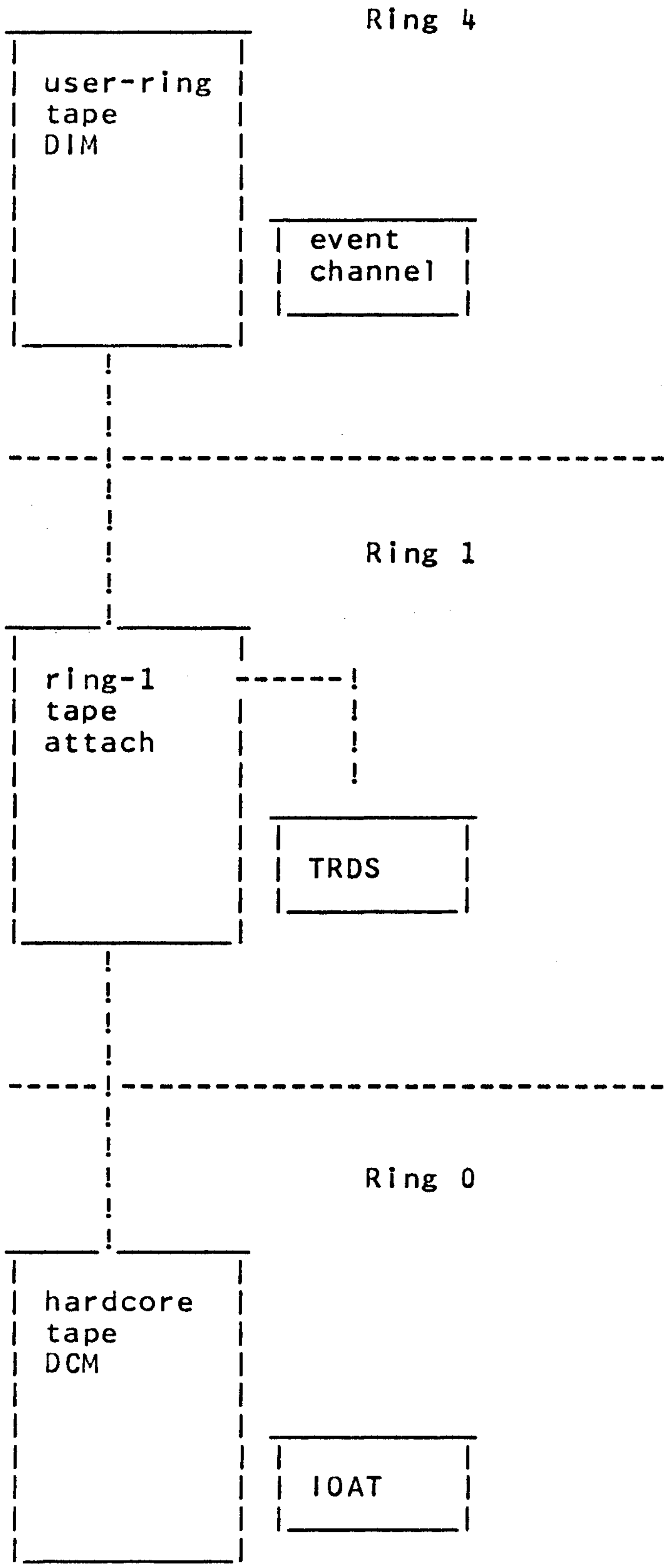

Figure 1 - User calls to attach tape

When the user makes a call to attach a tape, the user-ring tape DIM calls into a new gate segment into ring 1, which

1. Obtains and checks the tape reel identifier.

2. Interprets conventional identifiers like "scratch".

3. Looks up the TRDS for the tape requested.

4. Checks that the user has access to the TRDS.

5. Obtains all tape attributes from the TRDS or assigns default attributes if TRDS length is zero. Retention date is checked here if write attachment is specified.

6. Calls the hardcore DCM to get a drive assigned. This drive cannot be used by the process from ring 4 yet, since the validation level for the drive is set to 1.

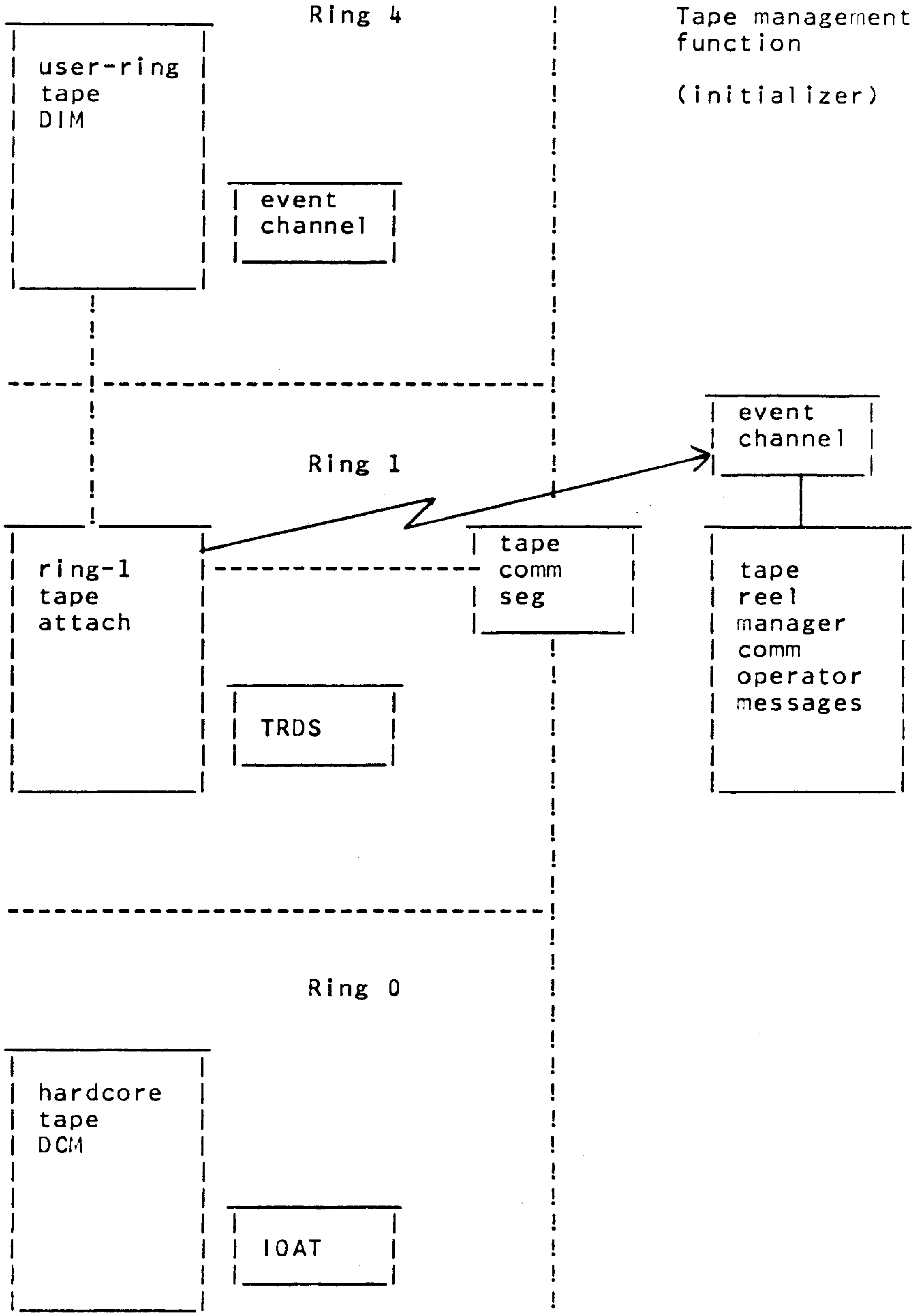

Figure 2 - User signals tape manager

If a drive is successfully assigned, the ring-1 tape attachment module obtains a free slot in the tape communications segment, and fills in the mount request information. This will include the tape identification, authentication if required, and perhaps a user message. The user's ring-4 event channel identification and process ID will also be placed in the request, so that the tape manager may reply to the user.

The user process will then signal the tape management function over a public event channel. The message sent over the channel will contain the index in the communications segment of the mount request.

The user process then returns to the user-ring tape DIM, which blocks on the user's event channel for the tape.

The tape manager, when awakened, performs any necessary validation of the mount request and then displays a message to the operator of the form

```
   1324.2 tape 19 mount reel 3702 on drive 6, auth
   --> tape
```

The tape manager then changes the state indicator in the tape communications segment's request block to indicate that operator action is pending, and returns.

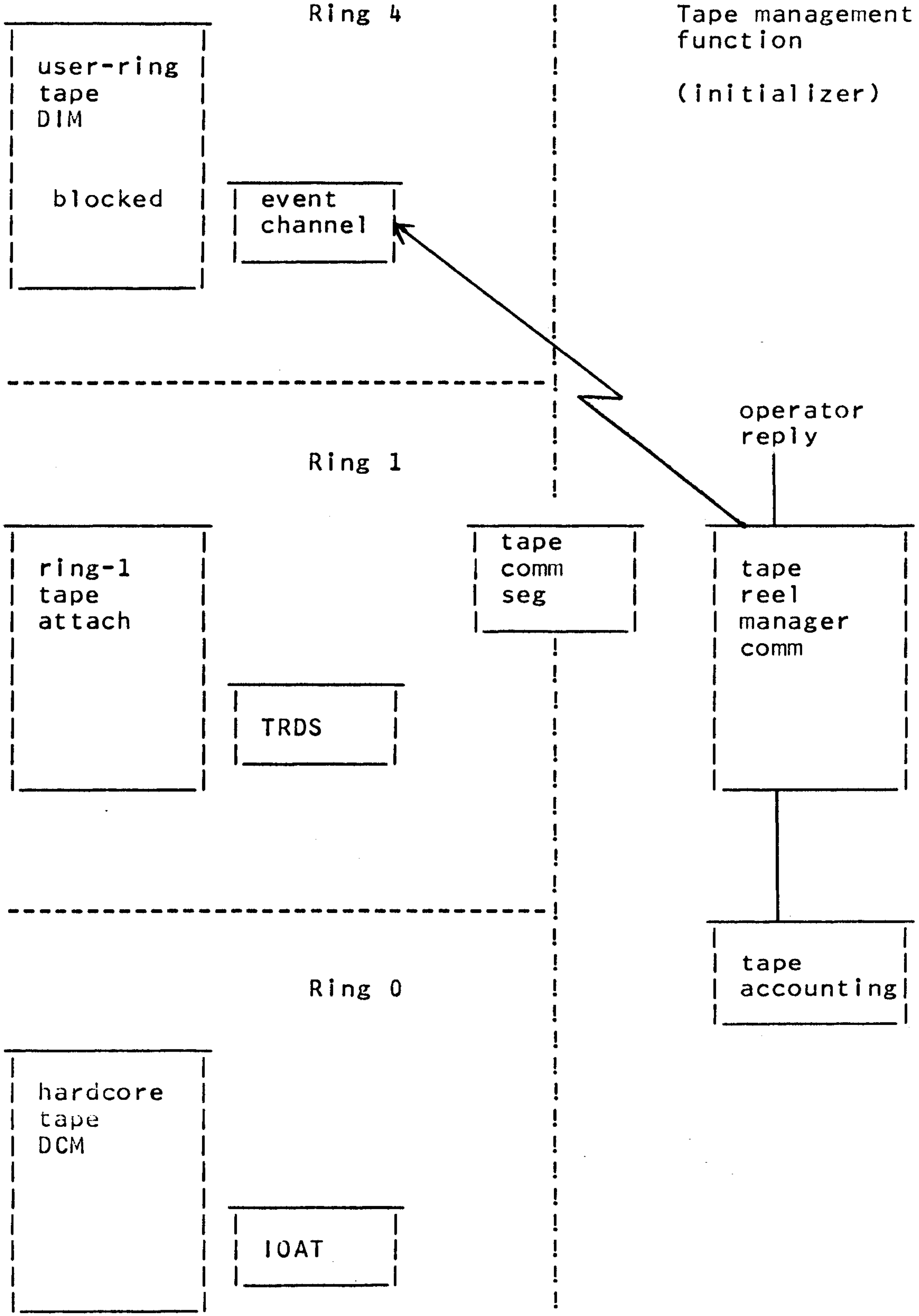

Figure 3 - Operator replies to mount request

The operator locates and mounts the tape. (The current "special interrupt" provided by the tape controller when a tape is readied will no longer automatically wake the user process.)

When the tape is mounted, the operator gives a command of the form

```
   reply tape 19 ok qmn
```

where the "19" is the slot number in the tape communications segment, and "qmn" is the authentication code from the sticker on the tape. The tape manager verifies the authentication, and may reject the mount if the authentication code is invalid.

If the operator cannot locate the user's tape, or if the user is not supposed to use the tape, the operator may make a negative response, like

```
   reply tape 19 notape
```

to indicate that the tape cannot be located.

The operator will be reminded of a request if he does not respond in 5 minutes. If the authentication for a reel does not match the expected authentication, the operator will be given one or two chances to retry his verification. The operator will be able to request a list of all current drive assignments and all pending mount requests.

If the mount is successful, the tape manager informs the accounting package that the user is signed on to a tape.

After the operator replies, the tape manager sends a signal to the user process, over its event channel, to tell it that it may proceed.

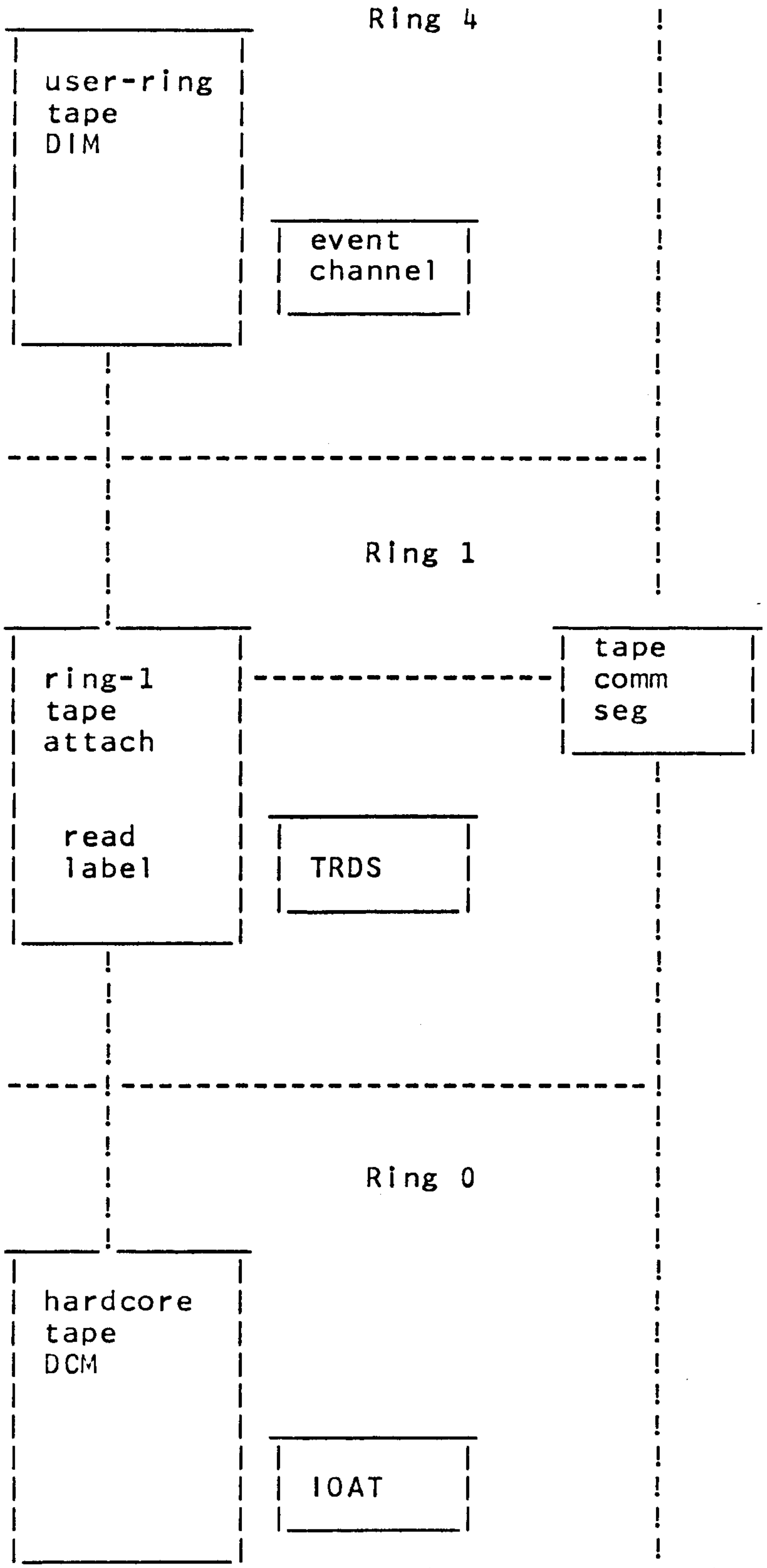

Figure 4 - User is awakened and checks mount

When the user process is awakened, it calls from the user ring into ring 1 again to check the mount.

The ring-1 program first examines the tape communication segment to make sure that the tape was successfully mounted.

The ring-1 program will then attempt to rewind the tape drive to determine write-protect status. If the tape is mounted for writing and a file-protect condition is sensed, the tape is unloaded and an error is returned to the user ring. The tape reel manager communicator will be signalled to have the operator remount the tape. The operator will give the same reply as he did for initially mounting the tape. If the tape is mounted for reading and a ring is sensed, a "Set-File-Protect" command will be issued to the tape drive to place it in File-Protect mode.

If label checking is required, the ring-1 module then reads the tape label and checks that the tape is the one requested. If this test fails, the tape is unloaded and an error is returned to the user ring.

If the tape is mounted for writing, and the "not_yet_labeled" switch is on, a label in the appropriate format will be written on the tape.

If the label is correct, or if the tape is a non-standard tape with no label, the ring-1 module calls the hardcore tape DCM to raise the validation level for the drive to the caller's ring number, so that the user may make subsequent read and write calls without entering ring 1.

Control is then returned to the user ring, which then calls directly into the hardcore for any tape reading or writing required, just as we do currently.

Miscellaneous topics

Once we have established this framework, the tape package can be extended to handle multi-file volumes and multi-reel files in a straigntforward fashion, since the ring 1 program can discover the information necessary for such operations from the appropriate TRDS.

In addition, an "append" mode for tape files should be allowed, so that users can append to the end of a tape even if they do not have permission to rewrite data already on the tape. This ability can be implemented at some later date, after the basic mechanism is working; the key to this is the ability to instruct the DCM that backward repositioning of a tape is not permitted, and a simple addition to the label-checking routine which positions the tape at the end of the file at attach time.

The transition from the current mode of operation to the new method can be done in several steps, without affecting user or system operation. The new ring 1 programs and the tape manager can be checked out completely before we force the tape DIM's to use them. Once they are checked out, we can place the ring-1 pattern in service by simply changing the user-ring tape DIMs to call the new code instead of the old; the old DCM calls to attach and detach a tape can be removed at a later date, when we are sure that the new pattern is operational.

The initial version of the ring-1 attach code need not be delayed by the requirement that every reel in an installation have a TRDS created for it. The access-checking code can be constructed so that, if a TRDS does not exist for a tape reel, the operator is asked whether one should be created. The operator may then specify special attributes (say in the case of a backup tape), may simply let the request go through (for a normal user tape), or may make a negative response (in the case of an error).

We need not relabel all of our tapes, either, in order to get the new facility on the air. This can be done by requiring authentication for all tapes mounted. IPC already has procedures for assigning authentication codes and numbering tape reels operating on the batch system.

Code to be written

The implementation plan for the tape improvements separates into several phases.

Phase 1 - hardcore preparation

1. Changes to the tape DCM.
   The attach entry to the DCM will be modified to save the caller's validation level, and all other entries will reject calls from a higher validation level than the level at which the drive was attached. This simple change will not affect current tape usage, since all tapes are now attached from the ring from which they will be used. However, if the user calls the new attachment function, and the drive is attached from ring 1, he will be unable to read or write from ring 4. A new entry point will also be provided, which can "promote" a drive to a higher validation level.

2. New gates and gate entries.
   A gate for the new attachment function and an additional entry in hcs_ for the promotion function will be added. Dummy versions of the ring-1 code will be installed as place-holders. They will probably just return error codes.

Phase 2 - new mount pattern

1. Tape Manager
   This system control function can be invoked by a user signal, to type out a message to the operator, or by an operator reply. It will manipulate the information in the tape communication segment.

2. Tape accounting
   This program will have sign-on and sign-off entry points, and may have an entry for an accounting update. It will charge the user for tape mounting, and for tape connect time.

3. Ring-1 tape attachment code.
   This program will respond to three user requests:
   a) Mount requests
   b) Mount-check requests (after operator mounts)
   c) Dismount requests
   The program accesses the TRDS segments for tapes, and makes calls to the hardcore tape DCM for channel and device assignment, label reading and writing, and channel unassignement. The program also accesses the tape manager's tape communications segment, and signals the tape manager when a mount request is made.

Phase 3 - using the new mount protocol and preparation for cutting the system over.

Code to be written

1. User-ring DIM changes
   New versions of the user-ring tape DIMs can now be tested, while the old ones continue to work. The changes consist of modifying the attach and detach calls to call ring 1 instead of the hardcore, and the move of the label reading and writing code from "tape_", in the user ring, into the ring-1 label code.

2. Tape registration and administration
   Special programs which can generate a new TRDS and label a tape will be required. Other programs may be needed, to do things like prepare a list of all tapes registered.

Phase 4 - enforcement of the new protocol

1. Once the user-ring DIMs have been thoroughly tested, and all tapes have volume tags and TRDS's, we will install new versions of "tape_" and "nstd_", and request any users who are communicating with the tape DCM directly to install new versions of their DIMs. If problems arise, we will be able to drop back from this new protocol immediately, by reinstalling the old DIMs.

Phase 5 - removal of old code

1. Once the full mount protocol is in smooth operation, we can remove the "tdcm_attach", "tdcm_detach", and "tdcm_message" entries from hcs_. The tdcm_message code and the code to handle special interrupts can be removed from the hardcodre tape DCM.

Phase 6 - further enhancements

1. BOS and bootload changes
   At some later time, we will modify BOS to examine the tape label of a tape it intended to use, to insure that the tape's accessibility indicator permits the kind of use intended.

2. Append mode
   Modifications to the tape DCM to allow an append-only mode for a tape reel may be done after the rest of the scheme is in operation. This will involve modifying the label-checking code to position the tape after the end of data instead of after the label, and will require the modifications to the DCM which prevent backward repositioning.

3. Assign drives in ring 1.

Two new gate segments will be added to the system to support tape operations. One will be callable by any user, and will support the user-ring DIM calls and any calls made by user utility programs. It will be called "trm_".

A privileged gate, "priv_trm_", will be required to support the operator and administrator commands.

Sample calls to each entry in the gates follow. The details of these calls will probably change during coding, but the list should give the flavor of what is intended.

1. Call made by user DIM to mount tape.

   ```
   call trm_$mount (tape_id, evchn, message, tcsindex, ec);
   ```

   This call returns the index in tcs of the request block set up. The user process should block until signalled by the tape mount function.

2. Call made by user DIM to check mount.

   ```
   call trm_$mount_check (tcsindex, ec);
   ```

   This call is made when the user is awakened by the mount function.

3. Call made when user DIM detaches tape.

   ```
   call trm_$dismount (tcsindex, ec);
   ```

   Perhaps an index of 0 means all tapes for this process.

4. Call made to return contents of a request block in tcs.

   ```
   call trm_$tape_status (tcsindex, p, ec);
   ```

   This fills in some block of words pointed to by p with the contents of the request block in tcs. Perhaps we should pass an area pointer and have the data allocated.

5. Calls pertaining to access control.

   ```
   call trm_$acl_replace (tape_id, aclp, n, ec);
   call trm_$acl_add (tape_id, aclp, n, ec);
   call trm_$acl_add1 (tape_id, uname, mode, brackets, ec);
   call trm_$acl_replace (tape_id, aclp, n, ec);
   call trm_$acl_delete (tape_id, aclp, n, ec);
   call trm_$acl_list (tape_id, aclp, n, uap, ec);
   ```

These calls are constructed on the pattern of the corresponding calls for segment ACL's. They should be made to look like the new access control primitives when documentation on these is published.

6. TRDS management.

```
call trm_$read_trds (tape_id, p, ec);
call trm_$write_trds (tape_id, p, ec);
```

These calls allow the user to inspect and modify tape attributes. Some information in the TRDS may not be available to the user, and some combinations of attributes may be illegal. This entry performs whatever checking is appropriate.

7. Reservation of resources.

```
call trm_$reserve (type, from, to, tcsindex, ec);
call trm_$unreserve (tcsindex, ec);
call trm_$list_reservations (tcsindex_array, n, ec);
```

These calls can be implemented later to support resource reservation.

8. Privileged calls

```
call priv_trm_$register (tape_id, owner_id, trdsp, ec);
call priv_trm_$unregister (tape_id, ec);
call priv_trm_$read_trds (tape_id, p, ec);
call priv_trm_$write_trds (tape_id, p, ec);
call priv_trm_$acl_list (tape_id, aclp, n, uap, ec);
call priv_trm_$acl_replace (tape_id, aclp, n, ec);
call priv_trm_$acl_add (tape_id, aclp, n, ec);
call priv_trm_$acl_delete (tape_id, aclp, n, ec);
call priv_trm_$list_tcs (tcsx, p, ec);
call priv_trm_$modify_tcs (tcsx, p, ec);
call priv_trm_$force_dismount (tcsx, ec);
```

The function of these calls is fairly obvious. Note that the tape mounting function does not have to be operating in ring 1, if appropriate entry points are provided in this privileged gate. The overhead involved should be investigated.

When the operator replies to the tape management function concerning a mount message, he types

```
reply tape NN KEY ARG
```

where "tape" is the identification of the system control function, used by the "reply" command to discover where to send the message; NN is the request index typed in the mount message; KEY is the keyword which indicates the success of the mount or the reason for failure; and ARG is an optional argument.

The following values are legal for KEY:

ok <auth>	mount successful, authentication is "auth"
notape	Tape reel cannot be located.
nopermit	User is not permitted to mount reel.
nosys	System or operations difficulty.
alrdy	Tape already mounted for another user.
redun	Tape already mounted for this user.

Name: tcs

The Tape Communication Segment (tcs) is an administrative-ring segment which is used for communication between the administrative-ring tape mount and dismount programs in a tape user's process and the tape mount and dismount request handling function in a system process. An entry is made in this table whenever a tape mount is requested. The entry is freed when the tape is dismounted.

Usage:

```
dcl 1 tcs based aligned,
    2 nents fixed bin,
    2 mount_proc bit (36),
    2 mount_chan fixed bin (71),
    2 n_mounted fixed bin,
    2 n_pending fixed bin,
    2 freep fixed bin,
    2 pad (25) fixed bin,
    2 array (100),
      3 fill (128) fixed bin;

dcl 1 tape_status_block based aligned,
    2 state fixed bin,
    2 procid bit (36),
    2 evchn fixed bin (71),
    2 uname char (24),
    2 uproj char (12),
    2 tape_reel_id char (16),
    2 drive_id fixed bin,
    2 channel_id fixed bin,
    2 devx fixed bin,
    2 authentication char (4),
    2 pdtep ptr,
    2 time_requested fixed bin (71),
    2 time_mounted fixed bin (71),
    2 time_dismount fixed bin (71),
    2 flags,
      3 not_multics_standard bit (1) unal,
      3 os_label bit (1) unal,
      3 ans_label bit (1) unal,
      3 no_label bit (1) unal,
      3 not_yet_labeled bit (1) unal,
      3 no_label_check bit (1) unal,
      3 no_authenticate bit (1) unal,
      3 small_reel bit (1) unal,
      3 seven_track bit (1) unal,
      3 not_high_density bit (1) unal,
      3 has_retention_date bit (1) unal,
      3 has_access_indicator bit (1) unal,
    2 mount_message char (120) aligned,
    2 operator_reply char (16) aligned,
    2 tape_label char (80);
```

EXPLANATION OF VARIABLES

1. nents is the number of entries in the table

2. mount_proc is the prccess id of the mount-handling process

3. mount_chan is the event channel used for mount requests

4. n_mounted is the current number of reels mounted

5. n_pending is the current number of mounts pending

6. freep is the index of the beginning of the free chain

7. pad is padding

8. array is the array of per-request entries

9. fill reserves storage for each table entry

10. tape_status_block is the description of a single request

11. state is the state of the request request

12. procid is the user process id

13. evchn is the user event channel

14. uname is the user's person id

15. uproj is the user's project id

16. tape_reel_id is the tape reel identification

17. drive_id is the drive id

18. channel_id — is the tape channel number

19. devx — is the device index, for hardcore calls

20. authentication — is the authentication code for the tape reel

21. pdtep — is a pointer to the user's PDT entry, for accounting

22. time_requested — is the time the mount request was made

23. time_mounted — is the time the operator replied to the mount request

24. time_dismount — is the dismount time

25. flags — are flags, copied from the TRDS, describing the tape reel

26. not_multics_standard — is "1"b if the tape is not in Multics standard format

27. os_label — is "1"b if OS/VS standard label

28. ans_label — is "1"b if ANSI standard label

29. no_label — is "1"b if the tape is unlabeled

30. not_yet_labeled — is "1"b if the tape is blank

31. no_label_check — is "1"b if the tape label should not be checked

32. no_authenticate — is "1"b if authentication is not required

33. small_reel — is "1"b if the tape reel is not the standard size

34. seven_track — is "1"b if the tape is seven-track

35. not_high_density — is "1"b if the tape is not recorded at the highest density

36. has_retention_date — is "1"b if tape is date protected

37. has_access_indicator is "1"b if special access control will be required for standalone uses

38. mount_message is the text of the user's mount request

39. operator_reply is the operator's reply

40. tape_label is the label as read from the tape

(END)

Name: trds

A Tape Reel Description Segment (trds) is created for each tape reel which is registered at an installation. The trds contains information about the attributes of the tape reel, and its access control list controls which users may access the tape reel. Each trds has a segment name of the form "reelid.trds"; all of the trds's are kept in a special system directory. The trds's are accessed by the administrative-ring tape mount and dismount programs when a user requests a tape mount.

Usage:

```
dcl 1 trds based aligned,
    2 flags,
      3 not_multics_standard bit (1) unal,
      3 os_label bit(1) unal,
      3 ans_label bit(1) unal,
      3 no_label bit (1) unal,
      3 not_yet_labeled bit (1) unal,
      3 no_label_check bit (1) unal,
      3 no_authenticate bit (1) unal,
      3 small_reel bit (1) unal,
      3 seven_track bit (1) unal,
      3 not_high_density bit (1) unal,
      3 has_retention_date bit (1) unal,
      3 has_access_indicator bit (1) unal,

    2 version fixed bin,
    2 retention_date fixed bin (71),
    2 read_error_count fixed bin,
    2 write_error_count fixed bin,
    2 density fixed bin,
    2 accessibility_indicator char (1);
```

EXPLANATION OF VARIABLES

1. not_multics_standard is "1"b if the tape is not in standard form (See MPM Reference Guide Section 5.3.)

2. os_label is "1"b if OS/VS standard label

3. ans_label is "1"b if ANSI standard label

4. no_label is "1"b if the tape is unlabeled

5. not_yet_labeled is "1"b if the label has not been written

6. no_label_check
   is "1"b if the label is not to be checked

7. no_authenticate
   is "1"b if the authentication is not required

8. small_reel
   is "1"b if the reel is a small size

9. seven_track
   is "1"b if the tape is seven-track

10. not_high_density
    is "1"b if "density" value has meaning

11. has_retention_date
    is "1"b if "retention_date" has meaning

12. has_access_indicator
    is "1"b if the tape has a special value (i.e. not blank) for "accessibility_indicator" below.

13. version
    is the version number of this declaration (currently 1)

14. retention_date
    is the date before which the tape cannot be written

15. read_error_count
    is the count of read errors on this tape

16. write_error_count
    is the count of write errors on this tape

17. density
    is the recording density in BPI

18. accessibility_indicator
    is blank for normal user tapes. This indicator is written in the tape label, where it will be checked by some special programs which access tapes when the TRDS and its access control list are not available.

(END)