TO: C. Clingen
F. Corbató
R. Feiertag
J. Gintell
N. Morris
R. Roach
J. Saltzer
T. Van Vleck
V. Voydock
S. Webber

FROM: Mike Spier

DATE: October 17, 1971

SUBJECT: Suggested implementation schedule for the Multics tape management facility.

TAPE REEL MANAGER IMPLEMENTATION

This is a suggested schedule for the implementation of the proposed tape management facility; familiarity with the proposed design is assumed. The objectives of this schedule are,

1) Effect an inexpensive initial implementation sufficient to allow users to manipulate their own tapes, while protecting system tapes from unauthorized access by users.

2) Make the initial implementation such that any subsequent upgrading will not affect the user interface.

3) Leave all present tape manipulating subsystems (e.g., backup, MST generator etc.) virtually unaffected to insure against possible bugs and also in order to avoid a "flag-day" type transition.

4) Assure that all future improvements of the facility will only be in the domain of protection, and cause no further modification to the functional interfaces.

The proposed design calls for the implementation of a ring-1 Tape Reel Manager (TRM) which performs all 'attach' and 'detach' operations in conformity with the tape protection scheme. The current implementation processes the 'attach' and 'detach' calls in the user ring. Moreover, the proposed design calls for a new interface using a Tape Reel Descriptor Segment (TRDS) for device-ID as opposed to the current implementation in which the device-ID is the actual (impossible to validate) reel number. Lastly, the tape protection scheme relies upon strict adherence to a tape control header checking discipline which is currently inexistent. A further significant change is the addition of an 'append' mode to the currently implemented 'read' and 'write'.

From an implementation point of view, the tape header checking discipline is contingent upon the availability of the ring-1 TRM. The system relies upon the smooth functioning of the current tape subsystems (especially the backup/retrieval mechanism) which must not be converted prematurely to use a potentially defective TRM. The suggested implementation schedule allows the gradual development of a TRM in parallel to the existing tape primitives and calls only for non functional modifications to the existing tape subsystems to make their tape header formats compatible with the projected TRM; they will be incorporated into the new tape facility at some point in the future when the TRM has proven itself reliable.

step-1: make tape subsystems bypass the i/o switch. The reason for this is that the switch is incapable of selectively choosing different tape packages depending upon the purpose of the tape. The new header requires additional parameters to the 'attach' call which cannot be elegantly accomodated within the rigid format of the standard i/o calling sequence.

A relatively trivial editing of these subsystems (which use tapes exclusively and therefore do not need the i/o switching option) will direct all i/o calls to the tape DSM. Moreover, the 'attach' calls will be directed to new dedicated entrypoints such as tape_xtach_$backup or tape_xtach_$mst allowing the passage of implicit parameters concerning the tapes' functional designation which is crucial if the protection scheme is to be implemented. The current 'attach' handlers (tape_xtach_, tdsm_xtach_) will be modified to produce the new tape header format and selectively generate the functional designations implied by the various special entrypoints.

step-2: move the entire tape package into ring-1 providing a universal gate for the regular 'attach' call and restricted gates to the special entrypoints. This implies that in the future all system tapes will be generated by members of privileged projects only, such as "SysLib", "SysMaint", "SysDaemon" etc.

step-3: modify the ring-0 TDCM to unload and detach a tape following a "rewind" request if that request did not come from ring-1. If a user is allowed to attach a tape for writing, rewind the tape and generate his own control header then the protection mechanism is completely invalidated.

step-4: code a new 'attach' handler named 'trm_attach_' which uses a TRDS pathname as device-id and which always reads the tape's control header before writing a new one. This is a partial TRM in that it does not do systemwide reel management nor validates a user's right to specify a TRDS. In effect, this very first TRM will be a transfer vector that generates a dummy TRDS and then calls the TRM proper specifying the fabricated TRDS as device-id. The TRM will also recognize the new 'append' attribute for tapes, positioning the tape after the last data record. When checking the attached tape's header it will reject any attempt to access a system tape as identified by the functional designation in its control header.

The new TRM will be installed in ring-1 and made known to the i/o switch as the standard tape DSM. All user requests will be directed to it. The older tape procedures will remain in effect but be accessible only to privileged users for the purpose of generating system tapes.

At this point we have effectively segregated system tapes from user tapes, while maintaining the current users' i/o interface. Users may still specify any random reel identifier and possibly access one another's tapes, however they are incapable at this point of accessing system tapes. The next step will assure that a user may only access his own allocated tapes.

step-5: write an administrative command to allocate tapes to users. This command, executing in some administrative process will generate a TRDS in the user's directory, in ring-1. The TRDS will contain the identifier of the allocated reel (reel number). Following the successful installation of this command, a user "flag-day" will be announced following which tapes could only be attached by their TRDS "device-id". This change in user interface is relatively minor. Currently, a user designates the tape of his choice by specifying the ASCII representation of the actual reel number. In the new scheme, he will merely use a different ASCII string, namely the TRDS's pathname. The only inconvenience in the changeover would be administrative, notifying users of their tapes' new identifier.

The TRM will be able to access the TRDS only if the requesting user has "rwa" access to it in ring-1, access which can only be granted by the tape administrator using some dedicated command. At this point users may no longer manipulate tapes which do not belong to them.

step-6: The TRM is upgraded to interpret the extended access attributes of the TRDS to allow sharing of tapes among users. New commands are provided to allow users to a) modify the contents of their TRDS, and b) grant other users extended access to their tapes.

This stage completes the implementation of the tape protection mechanism, but does not provide for systemwide reel management. That facility may be included at some future date to enable administrators to automatically allocate and de-allocate reels of tape, as opposed to the current practice of manual bookkeeping. Other administrative commands, such as a procedure to write headers on new tapes are also needed and may be implemented somewhere between steps 3 and 5.

The privileged tape subsystems (backup etc.) may be converted to use the TRM at any point following step-5, provided that it is judged sufficiently safe for system use. Any further upgrading of the tape facility will have no effect on the functional (i.e., tape i/o) interfaces and will only provide more elaborate administrative tools.