Date: 11/01/76

From: C. D. Tavares

Subject: Proposal for Standard Tape-Archival Facility

For some time, various users have expressed a desire for a standard facility to support archival of files to tape, the most recent such expression being a System Change Proposal generated by HLSUA. This MTB proposes the design of such a facility.

Your comments on this MTB are requested. Multics mail on the MIT machine is the preferred medium.

Experiences with the privately-maintained tape_archive facility

A private version of a tape archive facility has been available for use since about 1970. It provides a relatively simple method of performing all the major functions of tape archival, and was several times considered for installation. However, the actual coding is structured in a data-modular manner (rather than a procedurally-structured manner) which makes it difficult to maintain, and almost impossible to enhance.

In addition, the users of this facility have commented on certain problems they have encountered. These include:

Strict two-tape operation. Unless all outstanding requests are extractions from the tape, the facility mounts two tapes, copying files from one to the other and appending and replacing files on the fly. Logically, file appending could also be treated as a one-tape operation.

Limitation of the facility to one physical reel.

Inability of the facility to handle multi-segment files.

The command arrangement which requires a separate request to mount the tapes and perform the requests just made. At times, users simply forget to perform the actual tape processing.

The lack of a recovery facility which would allow all the readable data on the archival tape to be recovered in the event of damage to a part of the tape, or in the event of the online tape directory segment being deleted or becoming inconsistent. This latter has been a problem in the past where a successful archival processing has taken place shortly before a major file system disaster, with the result that the on-line segment becomes inconsistent with respect to what is actually on the tape. A major complication to the whole problem is that the tapes are in a format which is only readily decodable by the tape archive facility.

The inability to transfer files between Multics sites, or to another operating system, using the facility. (In all fairness, the original facility was not designed for this; but it did present problems when users of tape_archive at various sites attempted to move the archived files to other sites.)

Features of the new tape archival facility

The proposed new tape archival facility implements these desired features and provides additional capability.

Like the current tape_archive command, an on-line tape directory will be maintained. The user will issue requests to append, replace, and extract files from the archive. These requests will be queued in the online segment and will be performed when the tapes are next mounted. Requests will provide archival functions similar to those provided by the archive command keys a, ad, adf, r, rd, rdf, u, ud, udf, x, xd, and xdf.

In addition to the separate commands to queue requests and perform processing, a mode will be available which will read successive request lines from user_input (in the manner of reorder_archive) specifying file extractions, replacements, and so on. Exiting from this mode by typing "." would automatically cause the tape mounting and file processing. While in this mode, commands would be available to allow the user to quit without having any of his previous requests queued, or to cause the requests to be queued but not immediately perform the processing.

The facility will make use of the structured tape I/O modules. Standard ANSI tapes will be used to archive files. If the user wishes, the facility may also be used to archive files using IBM format.

Files on the tape will be named with meaningful unique names (not from unique_chars_) to aid in transporting to non-Multics systems. These names will be kept in the on-line table and will be available on request to the user.

The on-line table will be appended to the archive after any processing involving writing to the archive. A facility will be available to extract the table on request. Aside from the obvious utility as a backup mechanism, this feature will provide easy transportability of a set of files between Multics sites. The ANSI filename of the on-line file will be one of a set of readily identifiable names, different from any possible user file name. The use of a set of file names rather than one particular canned name insures that, should tape appends be interrupted by a system catastrophe which also destroys the on-line file, the last copy of the on-line file will still be available on the tape for reloading.

The facility will determine, via two cheap heuristics, whether a file must be recorded in the binary mode or is probably ASCII. The user will be able to override this heuristic explicitly, plus being able to specify EBCDIC encoding.

File copying from one set of tapes to another will not be performed except at the user's request. Until such a request, file replacement and deletion will simply result in dead space on the tape. A facility will be available for copying the volume set to a new volume set, compacting out the dead space in the process. User-accessible meters will be kept as an aid in determining when compaction would be advantageous.

Since it would be relatively cheap to maintain (in the on-line segment) a chain linking files with previous versions of themselves (dead space files), it would be reasonable to provide to the user the alternative of reloading an older version of a file, should the latest version become damaged. Compaction will reset this facility, since the previous copies go away.

Like the current facility, all interesting properties of the file (names, ACL, and so on) will be kept on the archival tape. The information for each file archived will be kept in a separate file on the tape. (For transferability purposes, it is not kept in the same physical file.)

The user will be able to dynamically add to the volume set when his archive requires an additional reel. This feature is a free consequence of using the tape_ansi_ and tape_ibm_ I/O modules.

Unresolved questions

The current tape archival facility uses a command syntax parallel to that of the archive command. Although this syntax no longer conforms to system standard practice, there are advantages to using a syntax that almost everyone is comfortable with.

Much the same remarks apply to the "Input" mode of operation which is designed solely to provide a recognizable point in time at which a user is ready to have the "automatic" tape processing performed. The system standard solution of making the user create a file containing his requests is too unsatisfying. However, this mode of operation would also be provided, since it is a handy way of performing certain "canned" sequences of operations (such as generating periodic "carry" tapes from a set of files.)