To: Distribution

From: Bill Silver, Max Smith

Date: August 6, 1974

Subject: Multics Tape Facility

INTRODUCTION

This document discusses the Multics tape facility as a whole. This facility is currently undergoing great change. These changes are being caused by the new I/O system (iox_) and by the new hardcore I/O interfacer (ioi_). In addition, the Multics tape facility is being changed to add necessary features which it did not support. This document discusses the old Multics tape facility which it defines to be those system supported tape functions which existed before iox_ and ioi_. It discusses the major factors which have forced this tape facility to change. It attempts to explain why these changes are necessary. It will show that these changes will lead to a coordinated and comprehensive new tape facility. The new tape facility is defined as those tape functions which will be available when the old I/O switch (ios_) and the old hardcore interface for tapes (tdcm) have been deleted from the system. The major functions that will be supported in the new Multics tape facility are:

1. Support for Multics, ANSI, IBM, and raw tape formats.

2. An I/O module that can process a tape with any supported format and without the user having to know the actual tape format.

3. Complete support by all tape I/O modules for all iox_ entry points which process sequential files.

4. Complete support for all iox_ entry points which process unstructured files for tapes with the Multics tape format. Support for unstructured files for other tape formats will be provided as needed.

5. Improved performance for tapes at the hardcore level and by all tape I/O modules.

6. Complete and useful interfaces at all levels of the tape facility. In this document a distinction is made between true user interfaces and system programmer interfaces. The most recognizable difference is that user interfaces are MPM documented and system programmer interfaces are PLM documented.

7. Drive and volume accounting, reservation and security.

8. On-line T&D support for tapes.

9. A complete set of command level tape utility packages.

This document is necessarily general. Following it will be a series of MTBs describing in detail the various building blocks of the new tape facility. Design reviews will be held to discuss each of these MTBs as they become available. Comments on the Multics tape facility as a whole will be appreciated by the authors.

Some existing MTBs are pertinent and the reader should be familiar with them.

MTB-028 and MTB-056 on the I/O Interfacer (partly obsolete)

MTB-074 on the I/O Buffer Manager (partly obsolete)

MTB-051 on a proposed new tape DCM (now obsolete)

MTB-076 on a proposed tape mount package (now obsolete)

MTB-061 and MTB-077 on the new I/O switch (iox_)

MTB-090 on the ANSI format tape I/O module

MTB-096 on the IBM format tape I/O module

OLD MULTICS TAPE FACILITY

For the purpose of discussion, let us divide the whole tape facility into three levels. These three levels are:

1. User Level Facility: This part of the tape facility consists of programs usually called from command level. They interface with tape I/O modules via an I/O switch. This part of the tape facility includes both user programs and system maintained tape tools.

2. Support Level Facility: This part of the tape facility consists mainly of tape I/O modules. They support calls from the I/O switch. They interface with the security kernel via gates. At this level, system programmers may supply their own programs.

3. Security Level Facility: This part of the tape facility is solely under control of the Multics Security Kernel. Its function is to control the use of

tape resources and to perform the physical I/O on the tape drives.

The structure of the old Multics tape facility (see Figure I) seems simple and clear. The problem is, that at each level, the facility is incomplete. We will discuss each level of the old Multics tape facility. We will try to point out the problems that exist at each level.

Old User Level Facility

All user tape programs interface with tapes directly through an I/O switch. The prefered alternative, for PL/I programs, is to use PL/I I/O. Unfortunately, in the old tape facility, none of the tape I/O modules are available through PL/I I/O.

The major problem with the system maintained tape tools is the absence of tape tools which Multics should have. A list of the basic tape tools which should be available is presented in the section on the new user level facility.

Old Support Level Facility

This level of the old Multics tape facility consists of tape I/O modules, each of which has its individual bad points. The current tape I/O modules are:

User I/O modules: The main problem with user tape I/O modules, as we shall see later, is that they interface on one side with ios_ and on the other side with tdcm.

tape_: This tape I/O module supports the Multics tape format. It does not support structured files and does not support all of the features required by iox_ for unstructured files.

nstd_: This tape I/O module supports a sort of raw tape format. However, there are many physical tape formats which it does not support. In addition, its maximum record size is too small.

astd_: This tape I/O module supports ANSI and IBM tape formats. However, it is not a system supported I/O module and is not fully operational.

In addition to the problems with each individual tape I/O module, there are general problems with all of these tape I/O modules. All these tape I/O modules provide a complete interface to ios_. Since all tape I/O modules must currently use tdcm, they all do their own buffering. Each must perform its own read

ahead, write behind, and error recovery. Each must know about physical DCW operation codes and physical tape status. In short, developing and maintaining this type of tape I/O module is a very difficult task.

Old Security Level Facility

The two main problems with the old Multics tape facility at this level are the absence of control over the use of tape volumes and drives, and tdcm itself. As the only interface to tapes, tdcm has several major disadvantages:

1. The tdcm interface is at a physical level. Thus all tape I/O modules have to know about the physical characteristics of tape drives.

2. Despite the fact that an I/O module must know about tapes at a physical level, the tdcm interface does not allow a tape I/O module to have very much control over the tape drives.

3. The tseg structure used by tdcm forces a particular buffering strategy on the I/O module.

4. tdcm is inefficient in that it performs at least one and sometimes three copies of data.

5. tdcm will wait in ring zero even when operating the tape asynchronously. This results in a traffic controller eligibility slot being wasted on a process which cannot run. This causes inefficiency not only for the process that is waiting but also for the whole system.

6. Since tdcm waits for I/O operations to complete, it does not allow the I/O module to WORK in an asynchronous manner.

REASONS FOR CHANGE

The problems with the old Multics tape facility that have been identified above are not, in themselves, the prime causes of the changes that must be made to the facility. These problems have existed for a long time and probably could prevail awhile longer. The real cause of change is the changing environment in which the Multics tape facility exists, namely, the new iox_ and ioi_ interfaces.

The advent of the new I/O switch, iox_, will eventually result in changes to all tape I/O modules and all user level tape

programs. Eventually, the inefficient ios_ writearounds that are needed to support these old interfaces will be deleted. The system maintained tape I/O modules must be changed so they suppport the iox_ entry points for sequential files and in some cases unstructured files.

The advent of ioi_ was fostered by the requirements of on-line T&D. Although it is not an absolute necessity that ioi_ become the standard ring zero tape interface, there are many practical reasons why it should be. Two of the most important are:

1. On-line T&D cannot use tdcm. It must use ioi_. They cannot both be in control of the tape channels at the same time. This implies that if tdcm were kept as the standard system tape interface, when tape T&D had to be run, all other tape I/O would have to be suspended.

2. ioi_ solves many of the tdcm problems noted above. It does not wait in ring zero, it does not restrict user control over the tape drives, etc. Tape I/O modules that use ioi_ will be more efficient due to the elimination of unnecessary copying of data.

NEW MULTICS TAPE FACILITY

Figure II depicts what is hoped will be the new Multics tape facility. Note that the whole facility is again divided into three levels. The changes which have been discussed should be aimed at producing this new coordinated and comprehensive tape facility. There are two major goals of this new tape facility:

1. The capabilities at each level should be so complete that most users will be able to perform most of their tape processing via the system maintained facility.

2. If the case arises that a subsystem inplementer must develop his own tape software, he should be able to easily insert his software at any level within the tape facility. (The security level is excepted of course.) This really means that the interfaces at each level must be as complete and useful as possible.

New User Level Facility

The major change at this level is the new I/O switch, iox_. One of the objectives of the new Multics tape facility is to have as many user programs as possible use tapes via PL/I I/O. Those programs which need more control over their tape I/O should use iox_.

The development of a complete set of system maintained tape tools must be an important part of the new Multics tape facility. A user should not be forced to develop tape tools which he could reasonably expect the system to provide. Some of the most obvious tape tools that need to be developd are:

1. A good tape archive system,

2. A dump tape command which can display the contents of physical tape records.

3. A tape copy command which can copy files from a tape written in any of the supported tape formats to files on a tape written in any one of the supported tape formats. The input and output tape formats should not have to be the same.

4. Commands that will reserve tape drives and tape volumes before they actually have to be used. Thus user subsystems may know before they start a phase of their operation that they have all of the resources necessary to complete that phase.

5. Commands that control the registration and access attributes of tape volumes and possibly tape drives.

New Support Level Facility

The user (MPM documented) interface at this level consists of tape I/O modules. In addition, there will be system programmer (PLM documented) interfaces to internal tape utility programs used by all tape I/O modules and the resource control package.

There will be system maintained tape I/O modules which support all of the common tape formats. In addition, the I/O module tape_file_ will be available. tape_file_ is not a complete I/O module on its own. It is used to perform attachments to one of the other tape I/O modules. It will choose the tape I/O module based on a tape format argument given in its attach description. If no tape format is specified in the attach description, it will determine the tape format from information obtained from the resource control package. If the resource control package cannot provide a tape format, a default tape format will be used. The default will probably be the Multics tape format.

The system maintained tape I/O modules that will be provided are:

tape_file_ - performs an attachment to one of the other tape I/O modules.

tape_ansi_ - processes ANSI format tapes.

tape_ibm_ - processes IBM format tapes.

tape_mult_ - processes Multics format tapes. Note, this I/O module supports unstructured files.

tape_raw_ - processes tapes in a raw format, i.e., as physical records. This tape I/O module can be used to read a tape that was written in any of the other logical tape formats. It allows users to block and deblock data in any logical format. Thus, instead of having to develop a new tape I/O module in order to process a tape in some special format, a user can use tape_raw_. This allows a user to solve his problem at the user level rather than at the support level.

The tape I/O modules described above should be sufficient for almost all tape processing. There may, of course, be reasons why a user still must develop his own tape I/O module. The main reason would be that he has to do a large amount of processing of tapes that have a format not acceptable to any of the standard tape I/O modules. In this case, he might not want to suffer the inefficiency associated with using the tape_raw_ I/O module.

In order to allow a user and system programmers to more easily develop and maintain tape I/O modules, the new Multics tape facility will include programs that perform functions needed by tape I/O modules. These programs can be called by any tape I/O module. They will be strictly an internal interface and documented only in PLMs. An analysis of the internal functions performed by all tape I/O modules has led to the specification of the following tape I/O module building blocks:

1. tape I/O modules: Each of the different tape I/O modules should be concerned only with its own tape formating problems. As much as possible, the functions which are common to all tape I/O modules should be maintained as part of some separate interface which is used by all tape I/O modules.

2. tape_ioi_ is the standard internal interface to ioi_. It will be called by all system maintained tape I/O modules. It may be called in all rings higher than ring zero. It manages the ioi_ workspace buffer and builds all necessary DCWs. It performs read ahead, write behind, and simple error recovery upon request.

3. tape_util_ is an internal interface which performs utility functions that are needed by tape I/O modules. Although described here as one program, it may in fact be implemented as several programs. To perform some of its functions it will call tape_ioi_. Some of these functions might be:
   a) backing up a tape in order to synchronize the logical and physical positions,
   b) rewinding a tape volume,
   c) setting the density of a drive,
   d) performing detailed decoding of tape status,
   e) performing comprehensive error recovery,
   f) parsing attach descriptions.

4. If it becomes necessary for tape I/O modules other than tape_mult_ to process unstructured files, a common program might be developed to perform this function.

Not all tape I/O modules will have to use all of the features provided by tape_ioi_ and tape_util_. However, even if tape I/O modules use only part of these capabilities, they should be significantly less complicated than existing tape I/O modules. Each of these interfaces should be correct enough so that users are not forced to replace them with their own procedures. However, nothing has been done to prohibit a user or system programmer from replacing any part or the user or support level tape facility.

New Security Level Facility

There are two new elements at this level of the tape facility. They are ioi_ and the resource control package. ioi_ allows us to remove tdcm from ring zero. The interface provided by ioi_ allows the physical I/O of a tape drive to be controlled from a higher ring.

The resource control package will control the allocation and access to all devices and volumes which can be assigned through ioi_. For tapes, this means that it will control the use of tape drives and tape reels. It will also maintain per reel information such as tape format, tape density, number of tracks, statistical information, etc.

THE PLAN OF ACTION

The old Multics tape facility will be changed into the new Multics tape facility in many steps. The implementation steps listed below are ordered according to what must be done first. In many cases, these steps may overlap. The actual implementation plan and completion dates for the various steps will be the subject of future Multics Task Reports.

1. One of the first goals of the new Multics tape facility is to eliminate all access to tapes through the hcs_ gate. The name tdcm_ has been added to the hcs_ gate. This will allow users to immediately begin changing their tape I/O modules so that they will not call tdcm through hcs_. All calls to hcs_$tdcm_xxxx must be changed to calls to tdcm_$tdcm_xxxx.

2. ioi_ and the initial version of the resource control package will be installed. The resource control package will allow users to actually assign devices through ioi_. For tapes, it will perform the drive attach and detach functions now performed by tdcm.

3. Since tapes cannot be used through both tdcm and ioi_ at the same time, tdcm will be deleted from ring zero. The name tdcm_ will be taken off the hcs_ gate. All the hcs_ gate entries to tdcm will be deleted. At this point in time all tape I/O modules must have been changed to call tdcm_. A ring four tdcm simulator will be installed. Its name will be tdcm_ and it will call the resource control package and ioi_.

4. On-line T&D for tapes will be installed. These programs will call the resource control package and ioi_ directly.

5. In order to provide some access to tapes via iox_, and thus through PL/I I/O, a new tape I/O module, ntape_, has been developed. It is just a temporary writearound to the ios_ tape I/O module, nstd_. ntape_ is currently installed.

6. The current ANSI format tape I/O module, astd_, will become a system maintained product. It will support all iox_ entry points for sequential files. Its initial implementation will not support the input/output opening mode.

7. All the current tape I/O modules that are available through ios_ will remain until they are no longer used.

8. Install tape_ioi_ and tape_util_.

9. Create all of the system maintained tape I/O modules that will be part of the new Multics tape facility. Note, these I/O modules will be just rewrites of the existing ios_ type tape I/O modules which support the corresponding tape formats. All of these new tape I/O modules, except tape_mult_, will support all of the iox_ entry points for sequential files. The tape_mult_ I/O module will support all of the iox_ entry points for unstructured files. All of the new tape I/O modules will call the resource control package. They will also call tape_ioi_ and tape_util_. They will in no way depend upon the tdcm interface.

10. The tape_file_ I/O module should be installed.

11. Users should have converted their tape I/O modules to use the tape_ioi_ interface rather than the tdcm interface. If there are still tape I/O modules which use the tdcm interface, the ring 4 tdcm simulator will still be available.

12. The full resource control package capabilities for tapes will be installed. This should result in operational and command level user changes but no changes to any tape I/O modules should be necessary.

13. The tape_mult_ I/O module will be made to support all iox_ calls for sequential files.

14. If there is enough demand, some of the other tape I/O modules (tape_ansi_ and tape_ibm_) may be made to support the iox_ entry points for unstructured files.

15. The command level tape tools discussed previously should be implemented. They will all use iox_.

16. A drive and tape volume reservation command should be installed. This may be part of a more general resource reservation command.

17. Hopefully, at some point in time, all user tape I/O modules either will be unnecessary or will have been converted to use the tape_ioi_ interface rather than the tdcm interface. The ring 4 tdcm simulator will be deleted.

18. Also, at some point in time, it is hoped that all system and user programs which use tapes will have been changed to call iox (either directly or through PL/I I/O) rather than ios_. The old ios_ tape I/O modules will be deleted.

Figure I
Old Multics Tape Facility

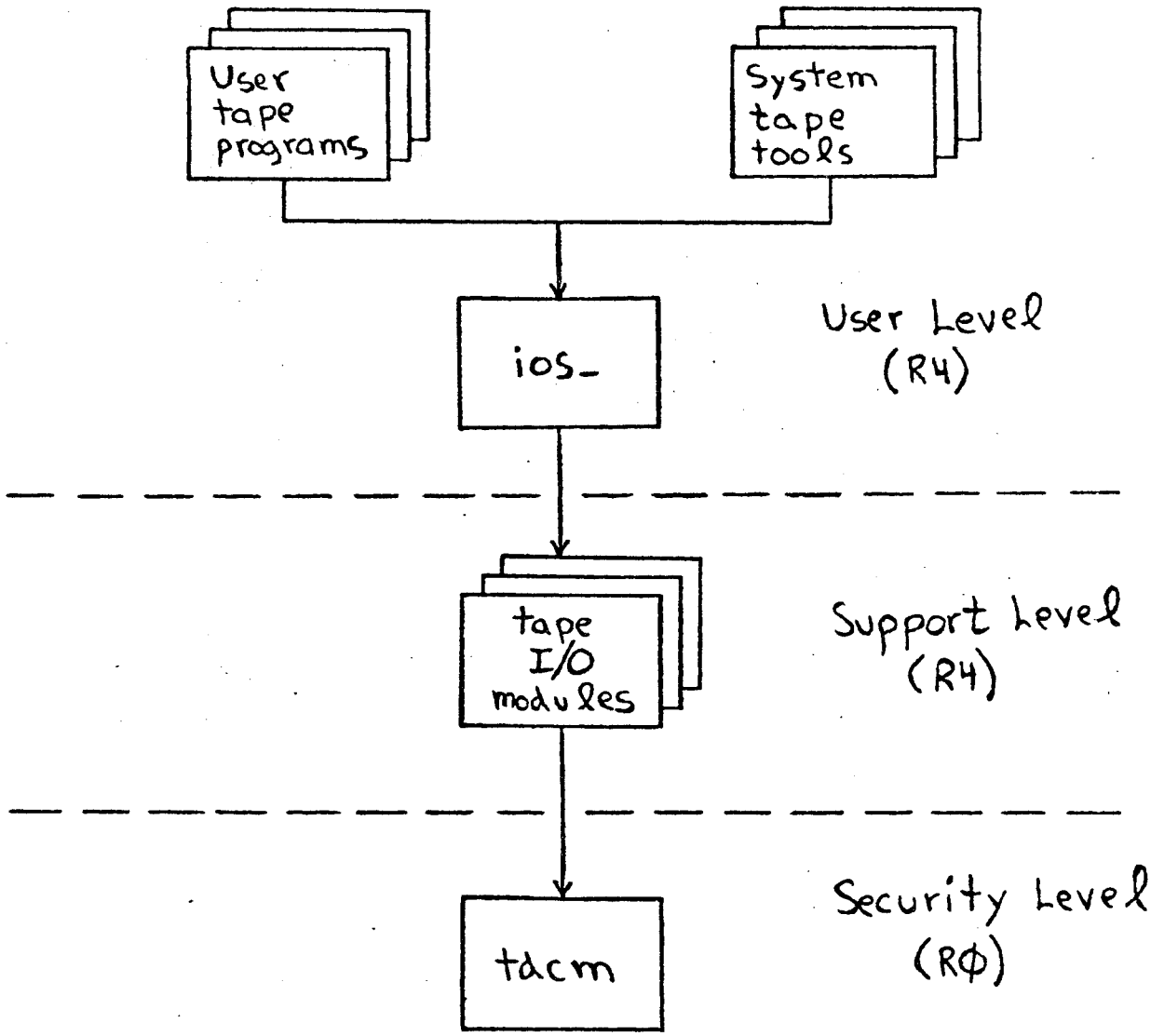

Figure II
New Multics Tape Facility