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

FROM: Mike Spier

Date: September 1, 1971

SUBJECT: The proposed design for the new off-line segment storage facility.

THE MULTICS OFF-LINE SEGMENT FACILITY

The Multics off-line segment facility extends the file system storage capabilities of Multics by providing for the orderly movement of segments between on-line (random access) and off-line (dismountable) storage media. The current implementation uses magnetic tapes for dismountable storage.

Before proceeding, it is necessary to clearly delineate the scope of this facility and draw the appropriate distinction between it and other magnetic tape oriented services in Multics.

The file system provides permanent storage for user files. The off-line segment facility, being an extension of the on-line file system, also provides for permanent storage of user files. In order to guarantee this permanency against accidental destruction, there exist on Multics several backup mechanisms which are used to save copies of user files on magnetic tape. These copies are temporary in nature, to be discarded within a predetermined period of time (for example, one year).

Additionally, Multics offers its users the convenience of magnetic tape drives, to be used as optional input/output devices under users' control.

We have, then, three distinct application which happen to share a common denominator, namely usage of magnetic tapes; they are nonetheless unrelated, both logically and functionally and should not be mistaken for one another. Following is a table of Multics' magnetic tape oriented services:

Input/Output System	File System	
	Permanent	Temporary
tape i/o	off-line segments	backup: incremental (logical) dump (logical) save (physical) retrieval: retrieve (logical) reload (logical) restore (physical)

Figure-1 is a graphical representation of the data flows between the various modules listed above.

As is described further on, both the off-line segment facility and the logical backup/retrieval facility use similiar data- and tape-record formats and consequently share, within the system's supervisor, common formatting procedures. It is important therefore to remember that they are nevertheless functionally distinct.

Design Overview

The off-line segment facility provides a means by which terminal sub-trees of the Multics file system hierarchy may be physically copied onto magnetic tape storage in order to relieve congested on-line storage space. The terminal sub-tree is copied onto tape, and the on-line copy is subsequently deleted. Regardless whether the copied item is a terminal data- (non-directory) or a directory segment, in either case a straightforward copying of that segment's binary image is effected. For the sake of simplicity we shall henceforth refer to such terminal sub-trees as "off-line segments".

The directory branch pointing to the copied off-line segment is maintained on-line, and is appropriately modified to indicate the segment's new device address (e.g., tape reel identifier). The user thus retains direct access to the segment's attributes (names, bitcount, access control list etc.), and is given the impression that the segment is directly available to him (albeit after some delay in time). The user may freely manipulate the branch; display it (e.g., by using commands such as 'list' 'status' etc.), modify it (e.g., 'rename' 'set_acl' etc.) and even delete it from his directory without ever having to resort to the physical reading of the off-line storage medium.

As an implementation restriction the user may not move the branch from one directory to another.

The segment's format on tape consists of the segment's binary image preceded by its file system name, in the form of a variable length condensed tree-name. This name is generated by concatenating all the unique identifiers of the branches which constitute the segment's tree-name. Directory segments, which are a conglomeration of various data structures, are considered for the purpose of output to consist of a single unstructured sequence of bits.

(Note: in the current system, a branch's unique identifier is regenerated during a complete disk reload; this will have to be modified, so as to preserve the original identifier and guarantee its original value for the branch's entire lifetime).

When an off-line segment is to be restored onto on-line storage, its address within the hierarchy is determined by its condensed tree-name; any break (i.e., unmatched component) in

the sequence of unique identifiers is interpreted to mean that the segment has been deleted, either explicitly (e.g., terminal branch pointing to off-line segment was deleted) or implicitly (e.g., a higher level sub-tree within which this segment is contained was deleted).

If the segment's address proved valid, restoration takes either of two forms depending upon whether or not the off-line segment is a directory. If it is a non-directory, it is simply copied back. If it is a directory, a check is made to determine whether or not it is of the current format. In the affirmative case the directory is simply copied back, otherwise a special reformatting procedure is invoked to convert the inconsistent format into the current one. For that purpose, every data structure within a directory carries a self designating code. Whenever a data structure is modified, its associated code is updated, and a dedicated reformatting procedure is added to the off-line segment restoration package to effect conversion from the previous format into the current one. The reformatting procedure may call upon yet other reformatting procedures if more than a single directory restructuring has taken place since the directory in question was moved off-line. The number of such conversion routines which may co-exist concurrently at any given time is, however, limited; as will be seen later on, it is guaranteed that the format of an off-line directory may not be older than a predetermined period of time (normally, one year), and consequently the system needs be burdened only by those conversion routines necessary in order to cover a year's worth of directory format modifications.

User Interface

Two basic user commands are provided,

```
move_off_line path1 -path2- ... -pathn-
```

and

```
restore_on_line path1 -path2- ... -pathn-
```

respectively abbreviated to 'mol' and 'rol'. The first effects the off-line migration of terminal sub-trees 'pathi', the latter restores them onto on-line storage. For reasons of implementation, it is forbidden to move off-line a sub-tree containing a nested off-line segment. Also, in order to achieve the orderly restoration of an off-line sub-tree, outputing has to proceed in a top to bottom sequence, starting with the root of the sub-tree. Thus, when the command 'mol' is invoked it first scans the entire sub-tree 'pathi' down to its terminal branches in order to detect possible off-line segments nested within, and if it finds them it displays an appropriate error message and quits.

In the absence of nested off-line segments, outputing commences at the segment pointed at by branch 'pathi', and if that segment is a directory, the command then proceeds to output the segments pointed at by the branches which are contained within that directory; this process is repeated until the entire sub-tree has been output.

As mentioned above, an off-line segment has a tape address stored in its branch. Therefore, when outputing a directory, the branches contained within that directory have first to be updated to reflect their respective segments' tape address, before the directory proper is copied onto tape. This predetermined tape address may sometimes prove to be wrong, as in the case when an "end-of-reel" condition is detected prior to completion of the entire output operation. This problem is relatively minor, because it is assumed that the majority of off-line segments are going to be terminal data segments and that the probability of dumping a directory is not very large, so that the probability of encountering an "end-of-reel" condition while outputing a directory is even smaller. The technique employed in correcting such an error condition is to declare all the tape copies of the already output portion of the sub-tree as deleted (by means of an indicator in the off-line segment table (OST)), and to restart the entire dumping process on a fresh reel of tape.

Once output has begun, the command must execute in an uninterruptable mode because any abnormal termination would leave the file system in an inconsistent state. The command is therefore designed to operate in two phases; the first scanning, error detecting and preparatory phase during which quits are allowed and recovery from an abnormal termination is possible, and the second output phase during which quits are inhibited and which is coded for efficiency in order to minimize its time requirements.

Off-line Segment Backup

Off-line segments must be backed up in the same manner in which the on-line file system is backed up. Every branch of the file system contains a tape address designating the last backed-up copy of that segment. The 'mol' command first initiates an incremental backup of the segment to be moved off-line, updating that segment's branch to contain the current tape address of the backed-up copy, and only then does it perform the actual output operation.

Backup tapes are kept for a predetermined period of time (one year) after which they are recirculated. In order to assure that off-line segments will be permanently backed up, it is necessary to repeat the backing up process of older off-line segments once every year. Additional reasons, such as the above-mentioned reformatting of obsolete directory formats and

the logistics of efficient tape reel management all support the advisability of a scheme by which old off-line segments are periodically restored on-line to be immediately output again on the current tape reel, in the current directory format and with a new backed-up copy on the current incremental backup tape reel.

This periodic update must be effected before the corresponding backup tape (containing the copy of the off-line segment) is recirculated. Tape reel management is controlled by the system which is cognizant of the contingencies involved, and which will not release a backup tape reel until the appropriate off-line segment update was performed.

Synchronization

Outputing a segment involves the synchronization of the requesting user's process with two asynchronously operating tape drives (one for the off-line storage tape and one for the incremental backup tape) which may prove to be an operational bottleneck since they serve the entire user community. The delays involved may be quite substantial. Each of the tape drives is controlled by an independent dedicated daemon process which services queued-up requests sequentially. The appearance of fast response to the interactive user is guaranteed through the implementation of an intermediate transitory state of the affected segments, by which they are declared (through a special purpose indicator in the segment's branch) to be logically off-line, even though they are still physically on-line. The physical movement of these segments to the off-line storage is then performed sometime in the future, at which point the on-line storage space which they had previously occupied is freed.

This implementation technique carries the implication that a logical off-line segment cannot be retrieved; the 'rol' command checks the "logical" indicator of a branch before accepting a restoration request. Multics guarantees service on the 'mol' and 'rol' commands within a predetermined period of time (say, 24 hours).

Salvaging

Given the relatively substantial delays in time which elapse between the initiation of the 'mol' and 'rol' commands and the actual accomplishment of the physical movement of segments between the on- and off-line storage media, these commands maintain an internal scratch pad memory in which all intented transactions are recorded. A segment is actually removed from on-line storage only when its tape copies (backup and off-line storage) have successfully been produced. This internal memory resides in a dedicated segment and contains sufficient information to enable the appropriate salvaging of segments whose

input/output movements were disturbed because of some system failure.

Backup

The logical backup facility (incremental backup and logical dump) uses the same tape formats and input reformatting procedures as the off-line storage facility. Every branch in the file system contains a tape address for the most recently backed up copy; to guard against mishap, a second tape address designates the previously backed-up copy as well. Backup is initiated periodically, either for selected recently used segments (incremental) or for the entire hierarchy (dump). A user command,

backup_retrieve path1 -path2- ... -pathn-

(abbreviated to 'bur') initiates the retrieval of the copy designated by the branch's backup tape address. Options may be specified to use the previous copy rather than the current one, or to search back n generations to a much older copy. Analogous to the 'rol' command, this request is put on the backup daemon process' queue and processed sometime in the future.

The Off-line Segment Table

The OST resides in a dedicated segment, in the administrative ring, and has an entry for every tape reel assigned to the off-line segment facility, regardless whether that reel is currently in use or not. A hashing algorithm allows access to a reel's entry by its unique (within the system) reel identifier.

An entry consists of a header followed by an array of n binary indicators, corresponding to the n records on the tape. The header contains control information; an indicator as to whether or not it is in current use, the tape's creation date, a record count, separate counts of both valid and invalid records on it, a count of the number of segments on it, checksums for the array of indicators etc.

The indicators designate the validity of their corresponding tape records, "1"b for a valid record and "0"b for an invalid one. When a record is appended to a tape, its indicator is set; when a record is read into memory in the process of retrieving an off-line segment, its indicator is reset.

The segment's tape address, in the branch, consists of the tape reel number, the record number and a record count. In order to accelerate retrieval, special tape-mark characters are written on the tape every so and so many (e.g., hundred) records. Knowing

the initial record number of the off-line segment, the tape may then be skipped --at high speed-- to the proximity of that record before the slower tape read operation begins. As mentioned before, the segment's on-line address is written on tape in the form of a condensed tree-name, thus the OST provides no clue as to the file system address associated with a given record.

If a user decides to delete an off-line segment, the 'delete' command simply resets that segment's associated record indicator(s).

Tape Reel Management

Tape reels, in a Multics installation, may typically be grouped according to their functional designation, as follows:

1) New unused tapes
2) Scratch tapes
3) Off-line segment storage tapes
4) Incremental backup tapes
5) Logical dump tapes
6) Physical save tapes
7) System bootload tapes
8) Proprietary tapes assigned to various users

Tape management is done, exclusively, by the system; no human being (e.g., operator) may arbitrarily decide to overwrite a given reel of tape, excepting the case of an owner of a proprietary tape.

Protection of tapes is guaranteed through rigorous adherence, by all system modules, to a tape-header checking discipline. Every reel of tape in the system is written, upon its introduction into the system, with a header record containing that reel's unique identifier and its functional designation. The system maintains a systemwide, administrative ring, database known as the Tape Reel Table (TRT) which contains an entry per tape reel. Entries may be accessed by reel identifier through application of a hashing algorithm.

As a special case, BOS checks tape reel headers to make sure that a given reel does not contain valid information; when writing a tape, BOS produces its own distinct tape header.

Any request for a tape reel is first validated by the tape reel manager by consulting the TRT. A valid request results in the display, upon some operator console, of a message directing the operator to mount a specific reel on a specific drive. Control is not returned to the requesting procedure until the reel's identity is validated by reading its header.

The tape reel manager is also responsible for the periodic deletion of obsolete tapes (e.g., incremental backup tapes which are older than one year); periodically, it scans the TRT in quest of such tapes and initiates the appropriate actions. As mentioned above, the tape reel manager is cognizant of functional relationships and contingencies among different tapes, and proceeds to delete them in proper order.

Tapes may further be protected by using color coded labels for the various categories mentioned above.

The TRT entry for a given tape reel contains that reel's identifier, its functional designation, the date on which it was written, the tape drive on which it was written, the number of records, statictical information regarding its frequency of use and frequency of error, etc.

The TRT may also contain statistical information concerning the usage and performance of tape drives.

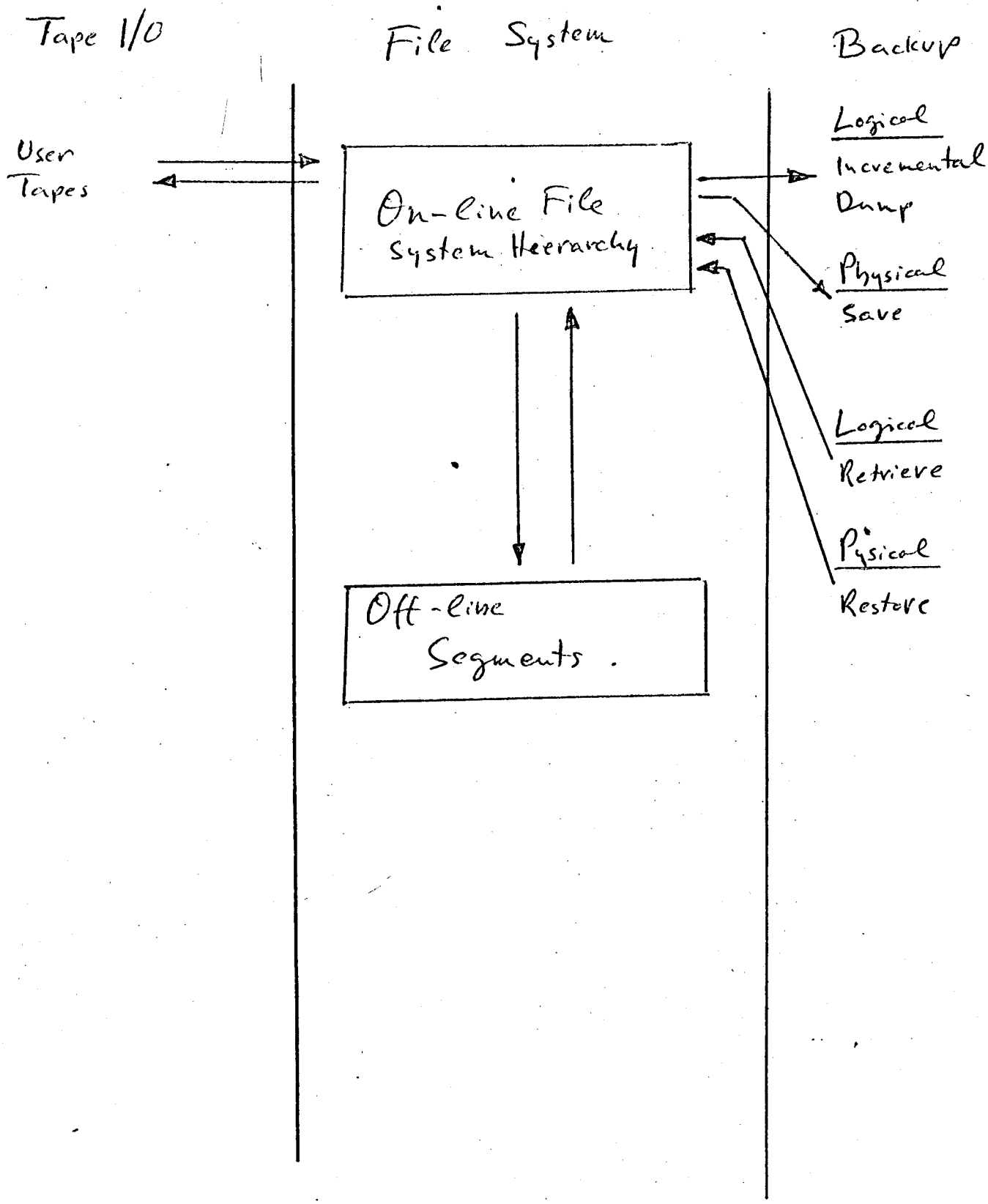

Figure-1: Data flow between tape oriented services.

Directory Format Changes

1) Directory Header

a) Version numbers for all include files used within this directory. These constant items are filled in by the procedure which initially creates the directory.
b) Single global version number for the entire directory, which is updated whenever any one or more include files within the directory are modified. It is a constant set by the procedure which initially creates the directory.
c) Count of off-line segments within this directory.

Note: The above items facilitate the global checking for nested off-line segments, and/or for reformatting during input.

2) Branch

a) Indicator that segment is off-line.
b) Indicator that segment is logically off-line, but physical(1) still on line.
c) Indicator that restoration request was issued, but that segment has not yet been read into on-line storage.
d) Tape address for off-line segment.
e) Two tape addresses for last backed up copies.

Figure-2: Suggested directory format changes