May 19, 1965.

PART I

A Proposal for the GE 636 File System

R.C. Daley

Introduction

This paper is a proposal for a multi-level file system for use within the Time-Sharing system to be implemented on the GE 636. The ideas presented here are an extension and modification of the current file system. The system proposed here will be upwardly compatible with the current file system from a logical point of view.

Brief Review

The file system will be designed to accomodate any configuration of I/O channels and/or devices. This system will provide a machine independent and flexible means of data handling through a standard interface to all users.

The smallest piece of information which can be manipulated by the file system is an element. An element may be a machine word, a character or a bit. For the GE 636 an element will be a 36-bit machine word. For a machine such as the IBM 360 an element might be an eight-bit character.

A file is defined as an ordered sequence of elements. Every file will have a unique name which is used to identify that file to the user. An element in a file is referenced by specifying the file name and the linear index. For example, the element "i" in file "a" is referred to as a(i). Files may be created, modified or destroyed by a user only through the use of the file system.

A file appears to the user to be a block of contiguous storage which may be referenced through normal sequential addressing conventions. However, the physical structure of the file is independent of the logical structure which the user experiences. The user may refer to a file only through the symbolic file name and should have no notion of where or how the file is stored. The number of elements which make up a file is arbitrary, and in fact, a file may exist with no elements.

A file directory is a special file which is maintained for the user by the file system. This file contains a list of file names and a short description of each file in the directory. The file description includes such information as the time this file was created or last modified, the time this files was last referred to, the "AUTHOR" of the file, the length of the file and a pointer to the file itself.

When a user is "Attached" to a file directory, all references to files are assumed to be defined within that directory unless stated otherwise by the user. The file

directory to which the user is currently attached is referred to as the user's "Current" file directory.

The Master File Directory (MFD) is a file directory which contains entries for other file directories. Initially, the MFD is the only file whose physical location in secondary storage is known to the file system. The expression d(a) might be used to uniquely define the file "a" which is defined in the file directory "d" which in turn is defined in the Master File Directory.

Structured File Directories

In the current file system all file directories must be defined in the Master File Directory. In the new file system it should be possible for a user to create new file directories which will be defined in the file directory to which he is currently attached. When a new directory is created, the directory in which the new directory is defined is said to be "superior" to the new directory. The new directory is said to be "inferior" to the directory in which it is defined. The process of defining inferior file directories may be continued to any desired depth. This means that the entire contents of secondary storage may be looked upon as a tree structure of files and file directories. The Master File Directory may be considered the "trunk" of this tree and all other directories will be inferior to it.

The highest level file directory to which a particular user may refer will be called the user's base file directory. This directory will be determined by the manner in which the user signs on to the system and by the administrative policies of the particular installation. The user's base file directory and all directories inferior to it will be known as user's "domain of influence". The user may refer directly only to those files which are defined within his domain of influence. He may, however, refer indirectly to any file or directory by means of the "LINK" feature which will be described later. A user may, of course, have several different domains of influence which may be selected by the manner in which he signs on to the system.

The file system will provide a means by which inferior file directories can be created and deleted. The file system will permit a file directory to be deleted only after it has become empty. The concept of what it means to delete a file directory will therefore be left to system commands and subroutines.

File Modes

A characteristic common to both files and file directories is the mode. The mode of a file is made up of a list of properties which may affect the usage of the file. The following is a list and a short description of these properties.

READ ONLY - A file with this property may only be read. An attempt to write into a read-only file will result in an error condition.

APPEND-ONLY - Information may only be written at the end of a file with this property. Any attempt to read or rewrite information which already exists in this file will result in an error condition.

EXECUTE-ONLY - A file of this property may only be referenced by a procedure which has been declared to have the authority to read execute only procedure segments. For example the "page turning algorithm" or the "segment linker" would most likely have permission to read execute-only files.

PRIVATE - A private file may only be referenced by the "AUTHOR" of the file i.e., the user who created or last modified this file.

LINK-FORBID - A file of this property may not be referred to by means of the "LINK" feature.

TRAP - When a file with this property is referenced, a "Trap" will be effected. The file system will call a procedure whose name is defined in the file directory entry for the file containing the trap. Any number of additional parameters may be defined within this file directory entry and will be passed as constants to this procedure. In addition, all pertinent information concerning the file and

the calling sequence which resulted in the "trap" will be passed as arguments to the procedure.

When returning to the file system, this procedure may specify one of the following options.

1. Continue processing the call which caused the "trap"
2. Ignore the call causing "trap" and return
3. Give calling procedure an error return specifying that access to the file referenced has been denied.

The user may inhibit the file trap process. If the file trap process is inhibited, all references to files with this property will cause an error return to the calling procedure. The file trap mechanism is useful for file monitoring, compiling source programs as referenced and placing additional constraints on file references.

PROTECTED - The mode of a file with this property may only be changed by the "AUTHOR" of the file. In addition a protected file may not be deleted. In effect, this property serves as lock on the file mode and the author has the only key.

To give a file particular property, the file system will set the appropriate binary bit in the file mode to "1". The mode of a file directory is taken to apply to all files

and directories which are inferior to it. The effective mode of a file is defined as the inclusive OR of the modes of the file and directory in which the file is defined.

For example, if a file directory is "protected" all inferior files and directories are said to be protected. The inferior files may add additional restrictions but may not remove any restrictions set by superior directories.

The "LINK" Feature

The Link feature provides a means by which a user may refer to files and directories which are defined outside of his domain of influence. The user may accomplish this by defining a "link" in his current file directory. Each link will have a mode and a name which is unique to the directory in which it is defined. The name of the link need not correspond to the name of the file or directory to which the link is made.

Once a link is established, the user may refer to it as if it were a real file or directory. If the link is made to a file directory the user may think of it as a directory which is inferior to the directory in which the link is defined. The mode of the link is used to place additional restrictions on the user of the link. For example, assume a user has a "read only" link called "ALPHA" which refers to the "protected" file, "BETA". The user may only refer to ALPHA as a read-only and protected file. However, the user who may refer directly to the file BETA will be able to

write into the file as well as read from it.

The Link Commands

The entry in the file system for creating a link will be a restricted entry and will not be available to most users. To facilitate the use of the link mechanism, the following set of commands are proposed. These commands are not an integral part of the file system and as such may be replaced or modified without modification to the basic file system.

1. PERMIT - This command will provide a means by which a user may declare to the system which files may be referred to by means of the link mechanism. When a user issues a PERMIT command, an entry will be made in the "permission" file which is maintained in the user's current file directory by the PERMIT command. When the user grants permission to link to a particular file he may also declare the MODE of the link. A possible form of the PERMIT command might be the following.

   PERMIT NAMELIST USERLIST MODE

   NAMELIST is a list of file or directory names which may be linked to by the users specified in USERLIST. All subsequent links to any of these files by any of these users will have the mode specified by MODE. If NAMELIST is "*", this is taken to mean all files which are defined in the user's current directory. If

USERLIST is "*", this is taken to mean all users who have access to the system. When a user allows a file directory to be linked to, he has given access to all files and directories which are inferior to that directory. A possible convention for specifying lists such as NAMELIST and USERLIST might be the following.

(NAME(1) NAME(2) ... NAME(n))

2. FORBID - This command will allow a user to revoke permission previously granted with the PERMIT command. The FORBID command has the same format as the PERMIT command. If possible FORBID will delete the corresponding entry from the permission file. If this is not possible, it will create an entry in another file, called the "exception" file. This feature will allow a user to give universal permission with the PERMIT command and list exceptions with the FORBID command. The FORBID command will also remove all links to files for which permission had previously been granted but is now revoked.

3. LINK - The LINK command provides a means by which a user may create a link to a file providing permission has been granted by the owner of the file. This command will check the permission and exception files and will create a link if permission has been granted. The LINK command might have the following form.

LINK LINKNAME FILENAME MODE

LINKNAME will be the name of the link and FILENAME is the file to which link will be made. The mode of the link is constructed by taking the inclusive "or" of the mode given by the LINK command and the mode specified in the permission file. Thus, the user may add additional restrictions to his use of the link but may not take away any restriction imposed by the permission file. The LINK command will maintain a "link" file which is a record of all links which have been made by means of the LINK command. This file is kept so that links may be deleted when permission is revoked by means of the FORBID command. The link mechanism is also useful in that it allows a user to create a shorthand notation for cross-referencing files within his domain of influence. A user needs no permission to create links to files which are defined within his domain of influence. It will be possible to create a link to another link as long as the latter link does not contain the LINK-FORBID property in its mode.

4. UNLINK - This command is used to delete a link which has been established by the link command. The UNLINK command might have the following form

UNLINK LINKNAME

The link specified by LINKNAME will be deleted and removed from the link file.

Password Protection of Files and File Directories

It may be desirable to have additional locks on certain files or directories. One method for accomplishing this would be to have the system ask for a "password" when a particular file or directory is referenced. This scheme might be implemented by the use of the following system commands.

LOCK NAME KEY

NAME is the name of the file or directory and KEY is the password to be associated with it. The LOCK command will place the file in "TRAP" mode, giving the name of a procedure to be called when the file is referenced. The KEY will also be recorded in the file directory entry as a parameter to be given to this procedure when it is called by the file system. When this procedure is called, it will ask the user who referenced the file to provide the password. If he fails to produce the correct password, control will be returned to the file system specifying that this user should be denied access to this file. If he does provide the correct password, control will be returned to the file system allowing the user to proceed. To remove the password interlock from a file, the following command might be used.

UNLOCK NAME

Before the password is removed, the user will be required to verify it. If the file is also "protected" only the AUTHOR of the file will be able to create and remove passwords for that file. It should be stressed that the above commands are not part of the file system and may be replaced or modified without modification to the file system.

Secondary Storage Allotments

It is hoped that the file system for the GE 636 will be invariant from installation to installation while accounting procedures may vary a great deal. It is, therefore, important to draw a clean dividing line between the file system and accounting procedures.

When a user first signs on the system, the file system should be given an "account" name or number. All files created by this user will be labelled with his account name. When a user wishes to increase his usage of secondary storage, the file system will call upon a "system accounting" procedure giving the user's account name, and the amount and class of storage desired. The accounting procedure will be responsible for maintaining records of secondary storage usage and allotments.

If the user has been allotted enough secondary storage to cover the request, the accounting module needs only to record the additional usage and return to the file system.

If the request for additional storage exceeds the user's allotment, the accounting module may return to the file system specifying one of the following options.

1. Request for storage denied
2. Request for storage granted
3. Request granted but account overdrawn

If the third option is to be taken, the file system will take steps to reduce the user's usage of secondary storage. This will be accomplished by a procedure known as the "demon" which runs asynchronously with respect to other processes in the system. To aid in this operation the demon should be able to request at any time the amount by which a user's usage of a certain class of storage should be reduced.

Hierarchies of Secondary Storage

In most cases a user will not need to know how or where a file is stored by the file system. A user's primary concern is that a file be readily available to him when he needs it. With the exception of files explicitly stored by the user on certain classes of removable storage (tapes, disk packs, etc.) only the file system will know on which device a file resides.

The file system will be designed to accomodate any configuration of secondary storage devices. These devices may cover a wide range of speeds and capacities. All

considerations of speed and efficiency of storage devices will be left to the file system.

All permanent secondary storage devices will be assigned a level number which will be assigned according to the relative speed of the device. The devices with the highest transmission and access rates will be assigned the highest level numbers.

As files become active, they will automatically be moved to the highest level secondary storage device available. As more space is needed on high level storage, the least active files will be moved to a lower level storage medium. The lowest level medium to which a file may be moved will be magnetic tape. Files can only be deleted by a user. The file system will not automatically delete any file.

When a user refers to a file which has been removed to tape backup storage, the user will be notified. At this time the user may decide to wait until the file is automatically retrieved from tape or may issue a request to retrieve the file while he works on something else.