1108 Hard Disk File System

Introduction:

The 1108 hard disk file system is designed to provide Interlisp-D users with a flexible mechanism for storing and accessing files. Like the file systems for the 1100 and 1132, the 1108 file system supports features like random access and version numbers on files. In addition the 1108 local file system supports a hierarchical naming structures for files. Though the first release of the 1108 file system does not support direct access from within Interlisp-D to MESA and STAR file partitions, it is possible to have partitions for MESA and STAR on the same disk used by Interlisp-D. More complete compatibility with MESA and STAR is planned for future releases of this software.

Partitioning the Disk:

The hard disk used with an 1108 may be partitioned into up to six regions called logical volumes. Each of these volumes acts like a separate device. Some may be used to hold files used with the Interlisp-D software, while others may be reserved for STAR and MESA files. You can partition the disk using the *Installation Utility* floppy, described elsewhere. Since partitioning the hard disk erases all its contents, you are advised to partition the disk appropriately before storing anything on it; otherwise, you will have to offload all files from the disk, partition, and then copy the files back to the disk.

In order to run the Interlisp-D software, a logical volume (usually named LISP) is needed to contain the system virtual memory and microcode. It should be 15,600 pages long; if shorter, it must be the last logical volume on the disk. This logical volume should not be used for user file space.

If you wish to store user files on the hard disk, a logical volume (preferably named DSK) should be allocated for this purpose. This volume can be used from within Interlisp-D just like the {DSK} device on the 1100 and 1132 except that it supports a hierarchical naming structure for files. This logical volume may be of any desired size (but probably should not be less than 1,000 pages). You can in fact use several logical volumes to store Lisp user files. If you do not have a lisp user file logical volume called DSK, the {DSK} device will be emulated by a coredevice, which is fine except that (a) it provides limited scratch space for some system programs and (b) when running GREET, Interlisp will fail to find {DSK}INIT.LISP and will have to prompt the user for an init file.

Installing a File Directory on a Logical Volume:

An Interlisp-D file directory must be installed on an 1108 volume before any files can be written to it. This is done using the function MKDIR:

(MKDIR <logical-volume-name>) [function]

MKDIR installs a file directory on the named logical volume. For example, (MKDIR ’DSK) will install a file directory on the logical volume named DSK. When MKDIR finishes, a message is printed stating that the logical volume now has a directory.

One must take some care in installing an Interlisp directory on a volume not to inadvertently destroy information already stored there. For instance, if the logical volume already has MESA or STAR files on it, access to these files will be lost. If the logical volume already has a LISP directory on it, that directory and hence access to the files it contains will be lost if a new directory is installed. The space used by the directory and files will also be lost. Installing an Interlisp directory is something that should only have to be done the first time the logical volume is used. After that, the system will automatically recognize and open access to the logical volumes that have Interlisp directories on them. If a logical volume already has an Interlisp directory on it, the MKDIR function will request confirmation before proceeding.

Accessing Logical Volumes From Interlisp-D:

Once an Interlisp directory has been installed on a logical volume, any program runnning in the system has access to the volume. The logical volume as a whole is treated as a device. Thus the logical volume DSK is referred to as the device {DSK} from within Interlisp-D. (The braces surrounding the name DSK indicate the reference is to a device.) As an example, consider the loading of an init file done during GREET. The system looks for the file {DSK}INIT.LISP, and if one exists loads it. A new file can be created on the DSK logical volume by executing the function call (OPENFILE '{DSK}MYFILE 'OUTPUT).

Unlike the local disk systems on the 1100 and 1132, the 1108 hard disk system supports a hierarchical naming convention for files. There isn't really a subdirectory structure, but the naming convention makes it appear as if one exists, and lets users group files. The characters "<" and ">" are used to delimit levels of the hierarchary. The "<" character is used only at the top level. The remaining levels are denoted by the ">" character. For example consider a two level structure on the device {DSK}. Let the top level be a user's name and the next level be a group of utility programs. The user's name will be WIZARD and the the utility level will be named UTILS. The file MOVEFILES.DCOM stored at the utility level will thus be referenced as {DSK}<WIZARD>UTILS>MOVEFILE.DCOM.

As a convenience, the 1108 local disk system supports connecting to any level in the name hierarchy. Thus if one wanted to work with WIZARD's utilities for awhile, the command CONN {DSK}<WIZARD>UTILS> would set up a default path that makes typing anything but the actual file name unnecessary. So (LOAD 'MOVEFILES.DCOM) would in fact load the file {DSK}WIZARD>UTILS>MOVEFILES.DCOM.

The Interlisp-D file system provides for device dependent file attributes, which can accessed and set by the functions OPENFILE, SETFILEINFO, and GETFILEINFO. The attributes accepted by the 1108 local disk system are as follows:

In the function OPENFILE, setting the LENGTH attribute sets the initial allocation of space for the file (in bytes). Setting the CREATIONDATE to a string overides the default of using the current time.

In the function GETFILEINFO, the LENGTH attribute returns the current length of the file in bytes. AUTHOR returns the author of the file as an atom. CREATIONDATE returns the creation time for the file as a string, READDATE returns the time at which the file was last opened for reading, and WRITEDATE returns the time the file was last opened for write. ICREATIONDATE, IREADDATE, and IWRITEDATE return the corresponding times as integers that GDATE will convert to string form.

In the function SETFILEINFO, setting the LENGTH attribute will change the size of the file. Setting AUTHOR as an atom or string will change the author of the file. (Only the first 40 characters are retained). Setting CREATIONDATE to a string changes the creation date for the file. Setting ICREATIONDATE to an integer (returned by IDATE) changes the creation date for the file.

Other Useful Functions:

(VOLUMEDISPLAY <flag>) can be used to create a status window for the local disk showing the names of the volumes, the amount of space allocated to each volume and the remaining space on each volume. (VOLUMEDISPLAY 'ON) opens the window in the place it was last left on the screen. (VOLUMEDISPLAY 'OFF) closes the window. Manually closing the status window using the right-button menu automatically does (VOLUMEDISPLAY 'OFF).

If the Volume Status window is on, it is updated as space for files is allocated and released during the course of an Interlisp-D session. In addition, an "*" character is displayed in the file system status window to the left of each volume name that has an open Interlisp file directory installed on it. This notation indicates which of the logical volumes on the local disk are accessible from Interlisp-D.

(DFSVOLUMES) returns a list of the names of the logical volumes on the 1108.

(DFSLISPVOLUMEP <logical-volume-name>) returns non-NIL iff the named volume has an Interlisp file directory open for Interlisp access.

(CURRENTVOLUME) returns the volume name of the volume containing the currently-running virtual memory.

Limitations:

Future releases of the 1108 hard disk file system will allow Pilot and Lisp files to coexsist on the same logical volume and share the same directory. In this release, however, Pilot and Lisp logical volumes are incompatible.

It should also be noted that at the current time there is no provision for directly installing a SYSOUT made by the Interlisp-D file system on an 1108 local disk. Doing this requires more compatibility with the MESA disk handling system than is now implemented. SYSOUTS can be made directly to floppy disks, however. It is also possible to make a SYSOUT directly to a file server (installation of Interlisp-D from a file server is supported) or to make the SYSOUT on a local disk volume and copy it to the server from Interlisp-D via the function COPYFILE.

Erasing an Interlisp File System Volume:

At some point in time it may become necessary to erase an entire Interlisp local file system volume. For example, you may wish to use that volume for storing and executing an Interlisp virtual memory image. Alternatively, you may have accidently used MKDIR on an already-existing file volume, and you wish to reclaim the space that was lost. Unfortunately, there is currently no easy way to do this from within Interlisp. To erase an entire Interlisp local file system volume, do the following:

1) Before leaving Interlisp, call the function

(MAKEPILOT <logical-volume-name>) [function]

This will remove the file device associated with the volume and will make non-Lisp tools (such as the Installation Utility floppy and the Lisp Installation Tool) able to recognize the volume.

[If you are a Xerox internal user, skip steps 2 and 3 and instead boot Othello and use the "Erase Volume" command to erase the volume in question.]

2) Boot the Installation Utility floppy. This will present a menu of operations. Choose the operation "Physical Volume Scavenge". This will ask for a confirmation. Note that this operation will reset ALL Interlisp file system volumes on the 1108. If you have multiple file volumes, their contents should be saved before doing this.

3) Boot the Lisp Installation Tool, and use the "Erase!" command to erase the bad volume. (You will not be able to boot the Lisp Installation Tool

until you have done the physical volume scavenge described in step 2.)

4) Restart your Interlisp memory image.

Scavenging:

This implementation of the 1108 hard disk file system includes a partial scavenger should your file system crash. If your local file system directory becomes smashed, call the function

(SCAVENGEVOLUME <logical-volume-name>) [function]

Provided the file maps are still intact, this will rebuild your directory with new file names. You can then rename the files appropriately by using SEE to look at contents, and sort out version numbers by using GETFILEINFO to compare creation dates.

Future implementations of the file system will include a scavenger that will rebuild file maps as well.