1108 Local Hard Disk File System
Harmony Release

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 structure for files. Though the current 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.

Partitioning the Disk

The hard disk used with an 1108 may be partitioned into up to ten regions called logical volumes. Some logical volumes may be used to hold Interlisp virtual memories, some to hold Interlisp files, while others may be reserved for Star and Mesa files. You can partition the disk with the Installation Utility floppy. 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, repartition, 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 16,200 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 you should allocate at least one logical volume (usually named LispFiles) for this purpose. Such volumes can be of any desired size, but are recommended to be between 1000 and 40000 disk pages. So long as there is such a volume, and you have at some point created a Lisp directory on it (see next section), you will have a local disk device called {DSK}. This device 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.

If you do not have a volume with a Lisp directory on it, Interlisp will emulate the {DSK} device by a core device, which is fine except: (a) The core device provides limited scratch space for some system programs, (b) when running GREET, Interlisp will fail to find {DSK}INIT.LISP and will have to prompt the user for an init file, and (c) since the core device is contained in virtual memory, it (and the files stored on it) can last only as long as you keep your virtual memory image.

Installing a File Directory on a Logical Volume

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

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

DFSCREATEDIRECTORY installs a file directory on the named logical volume. For example, (DFSCREATEDIRECTORY ’LispFiles) will install a Lisp directory on the logical volume named LispFiles. When DFSCREATEDIRECTORY finishes, it prints a message 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.

File Name Conventions

Each logical volume with a Lisp directory on it serves as a directory of the device {DSK}. Files are referred to as

{DSK}<*logical-volume-name*>*file-name*

Thus the file Init.lisp on the volume LispFiles would be called {DSK}<LISPFILES>INIT.LISP.

In addition, you can simulate subdirectories in file names by using the **>** character to delimit subdirectory names. There isn’t really a subdirectory structure, but the naming convention makes it appear as if one exists, and lets users group files. Files with subdirectories are written

{DSK}<*logical-volume-name*>*subdir1*>...>*subdirN*>*file-name*

For example, suppose you had a file LRdesign.tedit on the subdirectory ParserGenerator on the subdirectory Compiler on the directory (logical volume) LispFiles of the hard disk device; it’s name would be written {DSK}<LISPFILES>COMPILER>PARSERGENERATOR>LRDESIGN.TEDIT.

You can default file names for the 1108 hard disk in an unusual but simple way. That is: if the file does not have a subdirectory and you leave out the directory (logical volume) name, the directory will default to the next logical volume which has a Lisp directory on it after the volume containing the currently-running virtual memory. Thus if your disk has logical volumes Lisp, Tajo, and LispFiles, and the Lisp volume contains the running virtual memory, and the LispFiles volume has a Lisp directory on it (and the Tajo volume doesn’t), then {DSK}INIT.LISP will refer to the file {DSK}<LISPFILES>INIT.LISP. This defaulting convention is necessitated by several parts of the Interlisp system which create scratch files on the device {DSK} without specifying a directory (logical volume).

Note: Currently, the local file system does not follow the defaulting convention described above in every situation. For example, DIR {DSK}FOO* will not list all of the files starting with the characters FOO on the local file system volume, but DIR {DSK}<LISPFILES>FOO* will. In general, the user should supply the "full" device name {DSK}<LISPFILES> when using the local file system. It is also possible to connect to the local file system with CONN {DSK}<LISPFILES> and manipulate files without specifying the device.

Accessing Files From Interlisp-D

Once an Interlisp directory has been installed on a logical volume, any program runnning in the system has access to files on the the volume. Access is provided through the usual device independent file interface: CONN (to connect to any directory or subdirectory on the local disk), OPENFILE, CLOSEF, DELFILE, GETFILEINFO, SETFILEINFO, BIN, BOUT, LOAD, etc.

Other Useful Functions

(VOLUMEDISPLAY <flag>) [function]

can turn on a dynamically updated 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. <flag> = 'ON turns the status display on; <flag> = 'OFF turns it off. (Manually closing the status window will also turn it 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 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.

(VOLUMES) [function]

returns a list of the names of the logical volumes on the 1108.

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

returns the atom LISPFILE if the named volume has an Interlisp file directory open for Interlisp access, and the atom PILOT otherwise.

(DISKPARTITION) [function]

as on the 1100 and 1132, returns the name of the volume containing the currently-running virtual memory.

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 DFSCREATEDIRECTORY on an already-existing 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

(DFSPURGEDIRECTORY <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.

2) LOGOUT from Interlisp, and boot the Installation Utility floppy. This will display a menu of options. Select the option which says to erase the local file system logical volume.

3) Restart your Interlisp memory image.

Limitations

Only partial scavenging service is provided. There are two lookup structures stored on each lisp file system volumes: One, the directory, maps literal filenames onto file ID numbers; the other, the volume file map, maps file numbers onto runs of file pages. If the directory gets smashed, it can be rebuilt; see the documentation for the DlionFSScavenge package. However, in the unlikely event that the volume file map gets smashed, there is no recourse. It is expected that a volume file map scavenger will be included in the Intermezzo release of Interlisp-D.

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 Pilot 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.