NAME

fsdb - FFS debugging/editing tool

SYNOPSIS

fsdb [**-d**] [**-f**] [**-r**] *fsname*

DESCRIPTION

The **fsdb** utility opens *fsname* (usually a raw disk partition) and runs a command loop allowing manipulation of the file system’s inode data. You are prompted to enter a command with **fsdb (inum X)>** where *X* is the currently selected i-number. The initial selected inode is the root of the file system (i-number 2). The command processor uses the editline(3) library, so you can use command line editing to reduce typing if desired. When you exit the command loop, the file system superblock is marked dirty and any buffered blocks are written to the file system.

The following options are available:

-d Enable additional debugging output (which comes primarily from fsck(8)-derived code).

-f Left for historical reasons and has no meaning.

-r Open the file system read/only, and disables all commands that would write to it.

COMMANDS

Besides the built-in editline(3) commands, **fsdb** supports these commands:

help Print out the list of accepted commands.

inode *i-number*
Select inode *i-number* as the new current inode.

back Revert to the previously current inode.

clri *i-number*
Clear *i-number*.

lookup *name*
cd *name*
Find *name* in the current directory and make its inode the current inode. *Name* may be a multi-component name or may begin with slash to indicate that the root inode should be used to start the lookup. If some component along the pathname is not found, the last valid directory

encountered is left as the active inode. This command is valid only if the starting inode is a directory.

active
print Print out the active inode.

blocks
Print out the block list of the active inode. Note that the printout can become long for large files, since all indirect block pointers will also be printed.

findblk *disk_block_number ...*
Find the inode(s) owning the specified disk block(s) number(s). Note that these are not absolute disk blocks numbers, but offsets from the start of the partition.

uplink
Increment the active inode's link count.

downlink
Decrement the active inode's link count.

linkcount *number*
Set the active inode's link count to *number*.

ls List the current inode's directory entries. This command is valid only if the current inode is a directory.

rm *name*
del *name*
Remove the entry *name* from the current directory inode. This command is valid only if the current inode is a directory.

ln *ino name*
Create a link to inode *ino* under the name *name* in the current directory inode. This command is valid only if the current inode is a directory.

chinum *dirslot inum*
Change the i-number in directory entry *dirslot* to *inum*.

chname *dirslot name*
Change the name in directory entry *dirslot* to *name*. This command cannot expand a directory

entry. You can only rename an entry if the name will fit into the existing directory slot.

chtype *type*
Change the type of the current inode to *type*. *Type* may be one of: *file*, *dir*, *socket*, or *fifo*.

chmod *mode*
Change the mode bits of the current inode to *mode*. You cannot change the file type with this subcommand; use **chtype** to do that.

chflags *flags*
Change the file flags of the current inode to *flags*.

chown *uid*
Change the owner of the current inode to *uid*.

chgrp *gid*
Change the group of the current inode to *gid*.

chgen *gen*
Change the generation number of the current inode to *gen*.

btime *time*
mtime *time*
ctime *time*
atime *time*
Change the creation (birth), modification, change, or access time (respectively) on the current inode to *time*. *Time* should be in the format *YYYYMMDDHHMMSS[.nsec]* where *nsec* is an optional nanosecond specification. If no nanoseconds are specified, the *birthnsec*, *mtimensec*, *ctimensec*, or *atimensec* field will be set to zero. Note that **btime** is available on UFS2 file systems only.

quit, **q**, **exit**, *<EOF>*
Exit the program.

SEE ALSO

editline(3), fs(5), clri(8), fsck(8)

HISTORY

The **fsdb** utility appeared in 4.3BSD-Tahoe. It used the source code for fsck(8) to implement most of the file system manipulation code. The remainder of **fsdb** appeared in NetBSD 1.1 written by John T.

Kohl. It first appeared in FreeBSD 2.1.5 ported by Peter Wemm.

BUGS

Manipulation of ‘‘short’’ symlinks has no effect. In particular, one should not try changing a symlink’s type.

You must specify modes as numbers rather than symbolic names.

There are a bunch of other things that you might want to do which **fsdb** does not implement.

WARNING

Use this tool with extreme caution--you can damage an FFS file system beyond what fsck(8) can repair.