NAME

ffs - Berkeley fast file system

SYNOPSIS

In the kernel configuration file:
options FFS
options QUOTA
options SOFTUPDATES
options SUIDDIR
options UFS_ACL
options UFS_DIRHASH
options UFS_EXTATTR
options UFS_EXTATTR_AUTOSTART
options UFS_GJOURNAL

In fstab(5):

```
/dev/disk0a         /mnt ufs rw 1 1
```

DESCRIPTION

The Berkeley fast file system provides facilities to store file system data onto a disk device. **ffs** has been optimized over the years for speed and reliability and is the default FreeBSD file system.

Quotas

options QUOTA

This option allows system administrators to set limits on disk usage on a per-user basis. Quotas can be used only on file systems mounted with the **quota** option; see quota(1) and edquota(8).

Soft Updates

options SOFTUPDATES

The soft updates feature tracks writes to the disk and enforces metadata update dependencies (e.g., updating free block maps) to ensure that the file system remains consistent.

To create a new file system with the soft updates enabled, use newfs(8) command:

newfs -U *fs*

fs can be either a mount point listed in fstab(5) (e.g., */usr*), or a disk device (e.g., */dev/da0a*).

It is possible to enable soft updates on an *unmounted* file system by using tunefs(8) command:

tunefs -n enable *fs*

Soft updates can also add journaling that reduces the time spent by fsck_ffs(8) cleaning up a filesystem after a crash from several minutes to a few seconds. The journal is placed in an inode named *.sujournal*, and is kept as a circular log of segments containing records that describe metadata operations.

To create a new file system with both the soft updates and soft updates journaling enabled, use the following command:

newfs -j *fs*

This runs tunefs(8) command after newfs(8) command with **-U** flag enabled. It is possible to enable soft updates journaling on an *unmounted* file system by using tunefs(8) command:

tunefs -j enable *fs*

This flag automatically enables the soft updates feature when it is not enabled. Note that this tunefs(8) command will fail if a file *.sujournal* already exists before enabling the soft updates journaling.

File Ownership Inheritance

options SUIDDIR

For use in file sharing environments on networks including Microsoft Windows and Apple Macintosh computers, this option allows files on file systems mounted with the **suiddir** option to inherit the ownership of its directory, i.e., "if it’s my directory, it must be my file."

Access Control Lists

options UFS_ACL

Access control lists allow the association of fine-grained discretionary access control information with files and directories. This option requires the presence of the UFS_EXTATTR option, and it is recommended that UFS_EXTATTR_AUTOSTART is included as well, so that ACLs are enabled atomically upon mounting the file system.

In order to enable support for ACLs, two extended attributes must be available in the EXTATTR_NAMESPACE_SYSTEM namespace: *posix1e.acl_access*, which holds the access ACL, and *posix1e.acl_default*, which holds the default ACL for directories. If you are using file system extended attributes, the following commands may be used to allocate space for and create the necessary EA backing files for ACLs in the root of each file system. In these examples, the root file system is used; see *Extended Attributes* for more details.

```
mkdir -p /.attribute/system
cd /.attribute/system
extattrctl initattr -p / 388 posix1e.acl_access
extattrctl initattr -p / 388 posix1e.acl_default
```

On the next mount of the root file system, the attributes will be automatically started if UFS_EXTATTR_AUTOSTART is included in the kernel configuration, and ACLs will be enabled.

Directory Hashing

options UFS_DIRHASH

Implements a hash-based lookup scheme for directories in order to speed up accesses to very large directories.

Extended Attributes

options UFS_EXTATTR

Extended attributes allow the association of additional arbitrary metadata with files and directories, which can be assigned and retrieved from userland as well as from within the kernel; see extattrctl(8).

options UFS_EXTATTR_AUTOSTART

If this option is defined, **ffs** will search for a *.attribute* subdirectory of the file system root during the mount operation. If found, extended attribute support will be automatically started for that file system.

GEOM-based Journaling

options UFS_GJOURNAL

Implements a block level journaling of a UFS file system, which is for both data and metadata. To enable this, create a gjournal(8) GEOM provider for a block device by using the following command:

```
gjournal label da0
```

In this example, */dev/da0* is used as the target block device, and */dev/da0.journal* is created. Then create a new file system by using newfs(8) with the block level journaling flag and mount it:

```
newfs -J /dev/da0.journal
mount -o async /dev/da0.journal /mnt
```

async option is not mandatory but recommended for better performance because the journaling guarantees the consistency of an **async** mount.

It is also possible to enable the block level journaling on an existing file system. To do so, use gjournal(8) utility to label the underlying block device and tunefs(8) utility to enable the block level journaling flag:

```
gjournal label da0
tunefs -J enable /dev/da0.journal
mount -o async /dev/da0.journal /mnt
```

sysctl(8) MIBs

The following sysctl(8) MIBs are defined for use with **ffs**:

vfs.ffs.doasyncfree	Asynchronously write out modified i-node and indirect blocks upon reallocating file system blocks to be contiguous. (Default: 1).
vfs.ffs.doreallocblks	Enable support for the rearrangement of blocks to be contiguous. (Default: 1).

HISTORY

The **ffs** manual page first appeared in FreeBSD 4.5.

SEE ALSO

quota(1), acl(3), extattr(3), edquota(8), extattrctl(8), fsck_ffs(8), sysctl(8), tunefs(8)

M. McKusick, W. Joy, S. Leffler, and R. Fabry, "A Fast File System for UNIX", *ACM Transactions on Computer Systems*, 2, 3, 181-197, August 1984.

M. McKusick, "Soft Updates: A Technique for Eliminating Most Synchronous Writes in the Fast Filesystem", *Proceedings of the Freenix Track at the 1999 Usenix Annual Technical Conference*, 71-84, June 2000.

M. McKusick and J. Roberson, "Journaled Soft-updates", *BSD Canada Conference 2010 (BSDCan)*, May 2010.