NAME

tunefs - tune up an existing UFS file system

SYNOPSIS

tunefs [**-A**] [**-a enable** | **disable**] [**-e** *maxbpg*] [**-f** *avgfilesize*] [**-j enable** | **disable**] [**-J enable** | **disable**] [**-k** *held-for-metadata-blocks*] [**-L** *volname*] [**-l enable** | **disable**] [**-m** *minfree*] [**-N enable** | **disable**] [**-n enable** | **disable**] [**-o space** | **time**] [**-p**] [**-s** *avgfpdir*] [**-S** *size*] [**-t enable** | **disable**] *special* | *filesystem*

DESCRIPTION

The **tunefs** utility is designed to change the dynamic parameters of a UFS file system which affect the layout policies. The **tunefs** utility cannot be run on an active file system. To change an active file system, it must be downgraded to read-only or unmounted.

The parameters which are to be changed are indicated by the flags given below:

-A The file system has several backups of the super-block. Specifying this option will cause all backups to be modified as well as the primary super-block. This is potentially dangerous - use with caution.

-a enable | **disable**

Turn on/off the administrative POSIX.1e ACL enable flag.

-e *maxbpg*

Indicate the maximum number of blocks any single file can allocate out of a cylinder group before it is forced to begin allocating blocks from another cylinder group. Typically this value is set to about one quarter of the total blocks in a cylinder group. The intent is to prevent any single file from using up all the blocks in a single cylinder group, thus degrading access times for all files subsequently allocated in that cylinder group. The effect of this limit is to cause big files to do long seeks more frequently than if they were allowed to allocate all the blocks in a cylinder group before seeking elsewhere. For file systems with exclusively large files, this parameter should be set higher.

-f *avgfilesize*

Specify the expected average file size.

-j enable | **disable**

Turn on/off soft updates journaling.

Enabling journaling reduces the time spent by fsck_ffs(8) cleaning up a filesystem after a crash to a few seconds from minutes to hours. Without journaling, the time to recover after a crash is a function of the number of files in the filesystem and the size of the filesystem. With journaling, the time to recover after a crash is a function of the amount of activity in the filesystem in the minute before the crash. Journaled recovery time is usually only a few seconds and never exceeds a minute.

The drawback to using journaling is that the writes to its log adds an extra write load to the media containing the filesystem. Thus a write-intensive workload will have reduced throughput on a filesystem running with journaling.

Like all journaling filesystems, the journal recovery will only fix issues known to the journal. Specifically if a media error occurs, the journal will not know about it and hence will not fix it. Thus when using journaling, it is still necessary to run a full fsck every few months or after a filesystem panic to check for and fix any errors brought on by media failure. A full fsck can be done by running a background fsck on a live filesystem or by running with the **-f** flag on an unmounted filesystem. When running fsck_ffs(8) in background on a live filesystem the filesystem performance will be about half of normal during the time that the background fsck_ffs(8) is running. Running a full fsck on a UFS filesystem is the equivalent of running a scrub on a ZFS filesystem.

-J enable | **disable**
Turn on/off gjournal flag.

-k *held-for-metadata-blocks*
Set the amount of space to be held for metadata blocks. When set, the file system preference routines will try to save the specified amount of space immediately following the inode blocks in each cylinder group for use by metadata blocks. Clustering the metadata blocks speeds up random file access and decreases the running time of fsck(8). While this option can be set at any time, it is most effective if set before any data is loaded into the file system. By default newfs(8) sets it to half of the space reserved to minfree.

-L *volname*
Add/modify an optional file system volume label. Legal characters are alphanumerics, dashes, and underscores.

-l enable | **disable**
Turn on/off MAC multilabel flag.

-m *minfree*
Specify the percentage of space held back from normal users; the minimum free space threshold.

The default value used is 8%. Note that lowering the threshold can adversely affect performance:

- Settings of 5% and less force space optimization to always be used which will greatly increase the overhead for file writes.
- The file system's ability to avoid fragmentation will be reduced when the total free space, including the reserve, drops below 15%. As free space approaches zero, throughput can degrade by up to a factor of three over the performance obtained at a 10% threshold.

If the value is raised above the current usage level, users will be unable to allocate files until enough files have been deleted to get under the higher threshold.

-N enable | disable
Turn on/off the administrative NFSv4 ACL enable flag.

-n enable | disable
Turn on/off soft updates.

-o space | time
The file system can either try to minimize the time spent allocating blocks, or it can attempt to minimize the space fragmentation on the disk. Optimization for space has much higher overhead for file writes. The kernel normally changes the preference automatically as the percent fragmentation changes on the file system.

-p Show a summary of what the current tunable settings are on the selected file system. More detailed information can be obtained from the dumpfs(8) utility.

-s *avgfpdir*
Specify the expected number of files per directory.

-S *size*
Specify the softdep journal size in bytes. The minimum is 4M.

-t enable | disable
Turn on/off the TRIM enable flag. If enabled, and if the underlying device supports the BIO_DELETE command, the file system will send a delete request to the underlying device for each freed block. The trim enable flag is typically set when the underlying device uses flash-memory as the device can use the delete command to pre-zero or at least avoid copying blocks that have been deleted.

Note that this does not trim blocks that are already free. See the fsck_ffs(8) **-E** flag.

At least one of these flags is required.

FILES

/etc/fstab read this to determine the device file for a specified mount point.

SEE ALSO

fs(5), dumpfs(8), gjournal(8), growfs(8), newfs(8)

M. McKusick, W. Joy, S. Leffler, and R. Fabry, "A Fast File System for UNIX", *ACM Transactions on Computer Systems 2*, 3, pp 181-197, August 1984, (reprinted in the BSD System Manager’s Manual, SMM:5).

HISTORY

The **tunefs** utility appeared in 4.2BSD.

BUGS

This utility does not work on active file systems. To change the root file system, the system must be rebooted after the file system is tuned.

You can tune a file system, but you cannot tune a fish.