NAME

fsck_ffs, **fsck_ufs** - file system consistency check and interactive repair

SYNOPSIS

fsck_ffs [**-BCdEFfnpRrSyZz**] [**-b** *block*] [**-c** *level*] [**-m** *mode*] *filesystem ...*

DESCRIPTION

The specified disk partitions and/or file systems are checked. In "preen" or "check clean" mode the clean flag of each file system's superblock is examined and only those file systems that are not marked clean are checked. File systems are marked clean when they are unmounted, when they have been mounted read-only, or when **fsck_ffs** runs on them successfully. If the **-f** option is specified, the file systems will be checked regardless of the state of their clean flag.

The kernel takes care that only a restricted class of innocuous file system inconsistencies can happen unless hardware or software failures intervene. These are limited to the following:

Unreferenced inodes
Link counts in inodes too large
Missing blocks in the free map
Blocks in the free map also in files
Counts in the super-block wrong

These are the only inconsistencies that **fsck_ffs** with the **-p** option will correct; if it encounters other inconsistencies, it exits with an abnormal return status and an automatic reboot will then fail. For each corrected inconsistency one or more lines will be printed identifying the file system on which the correction will take place, and the nature of the correction. After successfully correcting a file system, **fsck_ffs** will print the number of files on that file system, the number of used and free blocks, and the percentage of fragmentation.

If sent a QUIT signal, **fsck_ffs** will finish the file system checks, then exit with an abnormal return status that causes an automatic reboot to fail. This is useful when you want to finish the file system checks during an automatic reboot, but do not want the machine to come up multiuser after the checks complete.

If **fsck_ffs** receives a SIGINFO (see the "status" argument for stty(1)) signal, a line will be written to the standard output indicating the name of the device currently being checked, the current phase number and phase-specific progress information.

Without the **-p** option, **fsck_ffs** audits and interactively repairs inconsistent conditions for file systems. If the file system is inconsistent the operator is prompted for concurrence before each correction is

attempted. It should be noted that some of the corrective actions which are not correctable under the **-p** option will result in some loss of data. The amount and severity of data lost may be determined from the diagnostic output. The default action for each consistency correction is to wait for the operator to respond yes or no. If the operator does not have write permission on the file system **fsck_ffs** will default to a **-n** action.

The following flags are interpreted by **fsck_ffs**:

-B   A check is done on the specified and possibly active file system. The set of corrections that can be done is limited to those done when running in preen mode (see the **-p** flag). If unexpected errors are found, the file system is marked as needing a foreground check and **fsck_ffs** exits without attempting any further cleaning.

-b   Use the block specified immediately after the flag as the super block for the file system. An alternate super block is usually located at block 32 for UFS1, and block 192 for UFS2.

        See the **-N** flag of newfs(8).

-C   Check if file system was dismounted cleanly. If so, skip file system checks (like "preen"). However, if the file system was not cleanly dismounted, do full checks, as if **fsck_ffs** was invoked without **-C**.

-c   Convert the file system to the specified level. Note that the level of a file system can only be raised. There are currently four levels defined:

        0   The file system is in the old (static table) format.

        1   The file system is in the new (dynamic table) format.

        2   The file system supports 32-bit uid's and gid's, short symbolic links are stored in the inode, and directories have an added field showing the file type.

        3   If maxcontig is greater than one, build the free segment maps to aid in finding contiguous sets of blocks. If maxcontig is equal to one, delete any existing segment maps.

        In interactive mode, **fsck_ffs** will list the conversion to be made and ask whether the conversion should be done. If a negative answer is given, no further operations are done on the file system. In preen mode, the conversion is listed and done if possible without user interaction. Conversion in preen mode is best used when all the file systems are being converted at once. The format of a file system can be determined from the first line of output from dumpfs(8).

This option implies the **-f** flag.

-d   Enable debugging messages.

-E   Clear unallocated blocks, notifying the underlying device that they are not used and that their
           contents may be discarded. This is useful for filesystems which have been mounted on systems
           without TRIM support, or with TRIM support disabled, as well as filesystems which have been
           copied from one device to another.

           See the **-E** and **-t** flags of newfs(8), and the **-t** flag of tunefs(8).

-F   Determine whether the file system needs to be cleaned immediately in foreground, or if its
           cleaning can be deferred to background. To be eligible for background cleaning it must have
           been running with soft updates, not have been marked as needing a foreground check, and be
           mounted and writable when the background check is to be done. If these conditions are met,
           then **fsck_ffs** exits with a zero exit status. Otherwise it exits with a non-zero exit status. If the
           file system is clean, it will exit with a non-zero exit status so that the clean status of the file
           system can be verified and reported during the foreground checks. Note that when invoked with
           the **-F** flag, no cleanups are done. The only thing that **fsck_ffs** does is to determine whether a
           foreground or background check is needed and exit with an appropriate status code.

-f   Force **fsck_ffs** to check 'clean' file systems when preening.

-m   Use the mode specified in octal immediately after the flag as the permission bits to use when
           creating the *lost+found* directory rather than the default 1777. In particular, systems that do not
           wish to have lost files accessible by all users on the system should use a more restrictive set of
           permissions such as 700.

-n   Assume a no response to all questions asked by **fsck_ffs** except for 'CONTINUE?', which is
           assumed to be affirmative; do not open the file system for writing.

-p   Preen file systems (see above).

-R   Instruct fsck_ffs to restart itself if it encounters certain errors that warrant another run. It will
           limit itself to a maximum of 10 restarts in a given run in order to avoid an endless loop with
           extremely corrupted filesystems.

-r   Free up excess unused inodes. Decreasing the number of preallocated inodes reduces the
           running time of future runs of **fsck_ffs** and frees up space that can allocated to files. The **-r**
           option is ignored when running in preen mode.

-S   Surrender on error. With this flag enabled, a hard error returned on disk i/o will cause **fsck_ffs** to
abort instead of continuing on and possibly tripping over more i/o errors.

-y   Assume a yes response to all questions asked by **fsck_ffs**; this should be used with great caution
as this is a free license to continue after essentially unlimited trouble has been encountered.

-Z   Similar to **-E**, but overwrites unused blocks with zeroes. If both **-E** and **-Z** are specified, blocks
are first zeroed and then erased.

-z   Clear unused directory space. The cleared space includes deleted file names and name padding.

Inconsistencies checked are as follows:

1. Blocks claimed by more than one inode or the free map.
2. Blocks claimed by an inode outside the range of the file system.
3. Incorrect link counts.
4. Size checks:
    Directory size not a multiple of DIRBLKSIZ.
    Partially truncated file.
5. Bad inode format.
6. Blocks not accounted for anywhere.
7. Directory checks:
    File pointing to unallocated inode.
    Inode number out of range.
    Directories with unallocated blocks (holes).
    Dot or dot-dot not the first two entries of a directory or having the wrong inode number.
8. Super Block checks:
    More blocks for inodes than there are in the file system.
    Bad free block map format.
    Total free block and/or free inode count incorrect.

Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence,
reconnected by placing them in the *lost+found* directory. The name assigned is the inode number. If the
lost+found directory does not exist, it is created. If there is insufficient space its size is increased.

The full foreground **fsck_ffs** checks for many more problems that may occur after an unrecoverable disk
write error. Thus, it is recommended that you perform foreground **fsck_ffs** on your systems periodically
and whenever you encounter unrecoverable disk write errors or file-system-related panics.

FILES

/etc/fstab contains default list of file systems to check.

EXIT STATUS

The **fsck_ffs** utility exits 0 on success, and >0 if an error occurs.

Specific non-zero exit status values used are:

1   Usage error (missing or invalid command arguments).

2   The **-p** option was used and a SIGQUIT was received, indicating that the system should be returned to single user mode after the file system check.

3   The file system superblock cannot be read. This could indicate that the file system device does not exist or is not yet ready.

4   A mounted file system was modified; the system should be rebooted.

5   The **-B** option was used and soft updates are not enabled on the file system.

6   The **-B** option was used and the kernel lacks needed support.

7   The **-F** option was used and the file system is clean.

8   General error exit.

16   The file system could not be completely repaired. The file system may be able to be repaired by running **fsck_ffs** on the file system again.

DIAGNOSTICS

The diagnostics produced by **fsck_ffs** are fully enumerated and explained in Appendix A of *Fsck - The UNIX File System Check Program*.

SEE ALSO

fs(5), fstab(5), fsck(8), fsdb(8), newfs(8), reboot(8)

HISTORY

A **fsck** utility appeared in 4.0BSD. It became **fsck_ffs** in FreeBSD 5.0 with the introduction of the filesystem independent wrapper as **fsck**.