NAME

lockfs - change or report file system locks

SYNOPSIS

/usr/sbin/lockfs [**-adefhnuw**] [**-c** *string*] [*file-system*]...

DESCRIPTION

lockfs is used to change and report the status of file system locks. **lockfs** reports the lock status and unlocks the file systems that were improperly left locked.

Using **lockfs** to lock a file system is discouraged because this requires extensive knowledge of SunOS internals to be used effectively and correctly.

When invoked with no arguments, **lockfs** lists the **UFS** file systems that are locked. If *file-system* is not specified, and **-a** is specified, **lockfs** is run on all mounted, **UFS** type file systems.

OPTIONS

The options are mutually exclusive: **wndheuf**. If you do specify more than one of these options on a **lockfs** command line, the utility does not protest and invokes only the last option specified. In particular, you cannot specify a flush (**-f**) and a lock (for example, **-w**) on the same command line. However, all locking operations implicitly perform a flush, so the **-f** is superfluous when specifying a lock.

You must be super-user to use any of the following options, with the exception of **-a**, **-f** and **-v**.

The following options are supported.

-a

Apply command to all mounted, **UFS** type file systems. *file-system* is ignored when **-a** is specified.

-c *string*

Accept a string that is passed as the comment field. The **-c** only takes affect when the lock is being set using the **-d**, **-h**, **-n**, **-u**, or **-w** options.

-d

Delete-lock (**dlock**) the specified *file-system*. dlock suspends access that could remove directory entries.

-e

Error-lock (**elock**) the specified *file-system*. elock blocks all local access to the locked file system and returns **EWOULDBLOCK** on all remote access. File systems are elocked by **UFS** on detection of internal inconsistency. They may only be unlocked after successful repair by **fsck**, which is usually done automatically (see mount_ufs(8)). elocked file systems can be unmounted.

-f

Force a synchronous flush of all data that is dirty at the time **fsflush** is run to its backing store for the named file system (or for all file systems.)

It is a more reliable method than using sync(8) because it does not return until all possible data has been pushed. In the case of **UFS** filesystems with logging enabled, the log is also rolled before returning. Additional data can be modified by the time **fsflush** exits, so using one of the locking options is more likely to be of general use.

-h

Hard-lock (**hlock**) the specified *file-system*. hlock returns an error on every access to the locked file system, and cannot be unlocked. hlocked file systems can be unmounted.

-n

Name-lock (**nlock**) the specified *file-system*. nlock suspends accesses that could change or remove existing directories entries.

-u

Unlock (**ulock**) the specified *file-system*. ulock awakens suspended accesses.

-v

Enable verbose output.

-w

Write-lock (**wlock**) the specified *file-system*. wlock suspends writes that would modify the file system. Access times are not kept while a file system is write-locked.

OPERANDS

The following operands are supported.

file-system

A list of path names separated by whitespace. Note that *file-system* can be a directory rather than the specific name of a file system, such as **/** or **/usr**. For example, if you specify **/export/home** as an argument to a **lockfs** command and **/export/home** is mounted on the root (**/**) file system, the **lockfs** command will take effect on the root file system.

USAGE

See largefile(7) for the description of the behavior of **lockfs** when encountering files greater than or equal to 2 Gbyte (2^31 bytes).

EXAMPLES

Example 1 Using **lockfs -a**

In the following examples, *filesystem* is the pathname of the mounted-on directory (mount point). **Locktype** is one of "**write**," "*name*," "*delete*," "*hard*," or "*unlock*". When enclosed in parenthesis, the lock is being set. **Comment** is a string set by the process that last issued a lock command.

The following example shows the **lockfs** output when only the **-a** option is specified.

```
  example#  /usr/sbin/lockfs -a

Filesystem Locktype Comment
/          unlock
/var       unlock

  example#
```

Example 2 Using **lockfs -w**

The following example shows the **lockfs** output when the **-w** option is used to write lock the **/var** file system and the comment string is set using the **-c** option. The **-a** option is then specified on a separate command line.

```
  example#  /usr/sbin/lockfs -w -c "lockfs: write lock example" /var
  example#  /usr/sbin/lockfs -a

Filesystem Locktype Comment
/          unlock
/var       write    lockfs: write lock example

  example#
```

Example 3 Using **lockfs -u**

The following example shows the **lockfs** output when the **-u** option is used to unlock the **/var** file system and the comment string is set using the **-c** option.

```
  example#  /usr/sbin/lockfs -uc "lockfs: unlock example" /var
  example#  /usr/sbin/lockfs /var

Filesystem Locktype Comment
/var       unlock   lockfs: unlock example

  example#
```

SEE ALSO

kill(1), mount_ufs(8), sync(8), attributes(7), largefile(7), ufs(4FS),

DIAGNOSTICS

file system: **Not owner**

You must be root to use this command.

file system **:Deadlock condition detected/avoided**

A file is enabled for accounting or swapping, on *file system*.

file system: **Device busy**

Another process is setting the lock on *file system*.