

NAME

nfsd - special filesystem for controlling Linux NFS server

SYNOPSIS

mount -t nfsd nfsd /proc/fs/nfsd

DESCRIPTION

The **nfsd** filesystem is a special filesystem which provides access to the Linux NFS server. The filesystem consists of a single directory which contains a number of files. These files are actually gateways into the NFS server. Writing to them can affect the server. Reading from them can provide information about the server.

This file system is only available in Linux 2.6 and later series kernels (and in the later parts of the 2.5 development series leading up to 2.6). This man page does not apply to 2.4 and earlier.

As well as this filesystem, there are a collection of files in the **procfs** filesystem (normally mounted at **/proc**) which are used to control the NFS server. This manual page describes all of these files.

The *exportfs* and *mountd* programs (part of the *nfs-utils* package) expect to find this filesystem mounted at **/proc/fs/nfsd** or **/proc/fs/nfs**. If it is not mounted, they will fall-back on 2.4 style functionality. This involves accessing the NFS server via a systemcall. This systemcall is scheduled to be removed after the 2.6 kernel series.

DETAILS

The three files in the **nfsd** filesystem are:

exports

This file contains a list of filesystems that are currently exported and clients that each filesystem is exported to, together with a list of export options for that client/filesystem pair. This is similar to the **/proc/fs/nfs/exports** file in 2.4. One difference is that a client doesn't necessarily correspond to just one host. It can respond to a large collection of hosts that are being treated identically.

Each line of the file contains a path name, a client name, and a number of options in parentheses. Any space, tab, newline or back-slash character in the path name or client name will be replaced by a backslash followed by the octal ASCII code for that character.

threads

This file represents the number of **nfsd** thread currently running. Reading it will show the number of threads. Writing an ASCII decimal number will cause the number of threads to be changed (increased or decreased as necessary) to achieve that number.

filehandle

This is a somewhat unusual file in that what is read from it depends on what was just written to it. It provides a transactional interface where a program can open the file, write a request, and read a response. If two separate programs open, write, and read at the same time, their requests will not be mixed up.

The request written to **filehandle** should be a client name, a path name, and a number of bytes. This should be followed by a newline, with white-space separating the fields, and octal quoting of special characters.

On writing this, the program will be able to read back a filehandle for that path as exported to the given client. The filehandle's length will be at most the number of bytes given.

The filehandle will be represented in hex with a leading `'\x'`.

The directory **/proc/net/rpc** in the **procfs** filesystem contains a number of files and directories. The files contain statistics that can be display using the *nfsstat* program. The directories contain information about various caches that the NFS server maintains to keep track of access permissions that different clients have for different filesystems. The caches are:

auth.domain

This cache maps the name of a client (or domain) to an internal data structure. The only access that is possible is to flush the cache.

auth.unix.ip

This cache contains a mapping from IP address to the name of the authentication domain that the ipaddress should be treated as part of.

nfsd.export

This cache contains a mapping from directory and domain to export options.

nfsd.fh This cache contains a mapping from domain and a filesystem identifier to a directory. The filesystem identifier is stored in the filehandles and consists of a number indicating the type of identifier and a number of hex bytes indicating the content of the identifier.

Each directory representing a cache can hold from 1 to 3 files. They are:

flush When a number of seconds since epoch (1 Jan 1970) is written to this file, all entries in the cache that were last updated before that file become invalidated and will be flushed out. Writing 1 will flush everything. This is the only file that will always be present.

content

This file, if present, contains a textual representation of every entry in the cache, one per line. If an entry is still in the cache (because it is actively being used) but has expired or is otherwise invalid, it will be presented as a comment (with a leading hash character).

channel

This file, if present, acts as a channel for request from the kernel-based nfs server to be passed to a user-space program for handling.

When the kernel needs some information which isn't in the cache, it makes a line appear in the **channel** file giving the key for the information. A user-space program should read this, find the answer, and write a line containing the key, an expiry time, and the content. For example the kernel might make

```
nfsd 127.0.0.1
```

appear in the **auth.unix.ip/content** file. The user-space program might then write

```
nfsd 127.0.0.1 1057206953 localhost
```

to indicate that 127.0.0.1 should map to localhost, at least for now.

If the program uses [select\(2\)](#) or [poll\(2\)](#) to discover if it can read from the **channel** then it will never see an end-of-file but when all requests have been answered, it will block until another request appears.

In the **/proc** filesystem there are 4 files that can be used to enable extra tracing of nfsd and related code. They are:

```
/proc/sys/sunrpc/nfs_debug  
/proc/sys/sunrpc/nfsd_debug  
/proc/sys/sunrpc/nlm_debug  
/proc/sys/sunrpc/rpc_debug
```

They control tracing for the NFS client, the NFS server, the Network Lock Manager (lockd) and the underlying RPC layer respectively. Decimal numbers can be read from or written to these files. Each number represents a bit-pattern where bits that are set cause certain classes of tracing to be enabled. Consult the kernel header files to find out what numbers correspond to what tracing.

SEE ALSO

[nfsd\(8\)](#), [rpc.nfsd\(8\)](#), [exports\(5\)](#), [nfsstat\(8\)](#), [mountd\(8\)](#) [exportfs\(8\)](#).

AUTHOR

NeilBrown