NAME

nfssvc - NFS services

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <sys/param.h>
#include <sys/mount.h>
#include <sys/time.h>
#include <nfs/rpcv2.h>
#include <nfsserver/nfs.h>
#include <unistd.h>

int
nfssvc(int flags, void *argstructp);
```

DESCRIPTION

The **nfssvc**() system call is used by the NFS daemons to pass information into and out of the kernel and also to enter the kernel as a server daemon. The *flags* argument consists of several bits that show what action is to be taken once in the kernel and the *argstructp* points to one of three structures depending on which bits are set in flags.

On the client side, nfsiod(8) calls **nfssvc**() with the *flags* argument set to NFSSVC_BIOD and *argstructp* set to NULL to enter the kernel as a block I/O server daemon. For NQNFS, mount_nfs(8) calls **nfssvc**() with the NFSSVC_MNTD flag, optionally or'd with the flags NFSSVC_GOTAUTH and NFSSVC_AUTHINFAIL along with a pointer to a

```
struct nfsd_cargs {
        char            *ncd_dirp;      /* Mount dir path */
        uid_t           ncd_authuid;    /* Effective uid */
        int             ncd_authtype;   /* Type of authenticator */
        int             ncd_authlen;    /* Length of authenticator string */
        u_char          *ncd_authstr;   /* Authenticator string */
        int             ncd_verflen;    /* and the verifier */
        u_char          *ncd_verfstr;
        NFSKERBKEY_T ncd_key; /* Session key */
};
```

structure. The initial call has only the NFSSVC_MNTD flag set to specify service for the mount point.

If the mount point is using Kerberos, then the mount_nfs(8) utility will return from **nfssvc**() with *errno* == ENEEDAUTH whenever the client side requires an ‘‘rcmd’’ authentication ticket for the user. The mount_nfs(8) utility will attempt to get the Kerberos ticket, and if successful will call **nfssvc**() with the flags NFSSVC_MNTD and NFSSVC_GOTAUTH after filling the ticket into the ncd_authstr field and setting the ncd_authlen and ncd_authtype fields of the nfsd_cargs structure. If mount_nfs(8) failed to get the ticket, **nfssvc**() will be called with the flags NFSSVC_MNTD, NFSSVC_GOTAUTH and NFSSVC_AUTHINFAIL to denote a failed authentication attempt.

On the server side, **nfssvc**() is called with the flag NFSSVC_NFSD and a pointer to a

```
struct nfsd_srvargs {
        struct nfsd         *nsd_nfsd;        /* Pointer to in kernel nfsd struct */
        uid_t               nsd_uid;  /* Effective uid mapped to cred */
        uint32_t   nsd_haddr;         /* Ip address of client */
        struct ucred        nsd_cr;           /* Cred. uid maps to */
        int                 nsd_authlen;      /* Length of auth string (ret) */
        u_char              *nsd_authstr;     /* Auth string (ret) */
        int                 nsd_verflen;      /* and the verifier */
        u_char              *nsd_verfstr;
        struct timeval      nsd_timestamp;    /* timestamp from verifier */
        uint32_t   nsd_ttl;   /* credential ttl (sec) */
        NFSKERBKEY_T nsd_key;  /* Session key */
};
```

to enter the kernel as an nfsd(8) daemon. Whenever an nfsd(8) daemon receives a Kerberos authentication ticket, it will return from **nfssvc**() with *errno* == ENEEDAUTH. The nfsd(8) utility will attempt to authenticate the ticket and generate a set of credentials on the server for the ‘‘user id’’ specified in the field nsd_uid. This is done by first authenticating the Kerberos ticket and then mapping the Kerberos principal to a local name and getting a set of credentials for that user via getpwnam(3) and getgrouplist(3). If successful, the nfsd(8) utility will call **nfssvc**() with the NFSSVC_NFSD and NFSSVC_AUTHIN flags set to pass the credential mapping in nsd_cr into the kernel to be cached on the server socket for that client. If the authentication failed, nfsd(8) calls **nfssvc**() with the flags NFSSVC_NFSD and NFSSVC_AUTHINFAIL to denote an authentication failure.

The master nfsd(8) server daemon calls **nfssvc**() with the flag NFSSVC_ADDSOCK and a pointer to a

```
struct nfsd_args {
        int         sock;       /* Socket to serve */
        caddr_t     name;       /* Client address for connection based sockets */
        int         namelen;/* Length of name */
```

```
};
```

to pass a server side NFS socket into the kernel for servicing by the nfsd(8) daemons.

RETURN VALUES

Normally **nfssvc**() does not return unless the server is terminated by a signal when a value of 0 is returned. Otherwise, -1 is returned and the global variable *errno* is set to specify the error.

ERRORS

[ENEEDAUTH] This special error value is really used for authentication support, particularly Kerberos, as explained above.

[EPERM] The caller is not the super-user.

SEE ALSO

mount_nfs(8), nfsd(8), nfsiod(8)

HISTORY

The **nfssvc**() system call first appeared in 4.4BSD.

BUGS

The **nfssvc**() system call is designed specifically for the NFS support daemons and as such is specific to their requirements. It should really return values to indicate the need for authentication support, since ENEEDAUTH is not really an error. Several fields of the argument structures are assumed to be valid and sometimes to be unchanged from a previous call, such that **nfssvc**() must be used with extreme care.