

Example: mounting a remote server

- get fhandle (via MOUNTPROC_MNT rpc to mountd)
- fill in struct nfs_args
 - ◆ struct nfs_args na
- call mount(2) syscall
 - ◆ mount("/mnt", flags, "nfs", &na, sizeof(na))

CSE506: Operating Systems

7

Contents of struct nfs_args

```

NA->addr {sockaddr_in} (len=16) =
  "02000801803b14640000000000000000"
NA->addr.sin_family = "2"
NA->addr.sin_port = "264"
NA->addr.sin_addr = "803b1464"
NA->hostname = "opus"
NA->namlen = 255
NA->filehandle =
  "008000f4000000002000a00000000000026e
  065b6c000a000000000000026e065b6c"
NA->version = 3
NA->flags = 0x0
NA->rsz = 4096
NA->wsz = 4096
NA->bsize = 0
NA->timeo = 7
NA->retrans = 3
NA->acregmin = 3
NA->acregmax = 60
NA->acdirmin = 30
NA->acdirmax = 60

```

CSE506: Operating Systems

8

NFS V.2 (1984)

- Built on top of UDP
 - 17 calls
- | | | | |
|----------------|---|-------------|----|
| NFS_NULL | 0 | NFS_CREATE | 9 |
| NFS_GETATTR | 1 | NFS_REMOVE | 10 |
| NFS_SETATTR | 2 | NFS_RENAME | 11 |
| NFS_ROOT | 3 | NFS_LINK | 12 |
| NFS_LOOKUP | 4 | NFS_SYMLINK | 13 |
| NFS_READLINK | 5 | NFS_MKDIR | 14 |
| NFS_READ | 6 | NFS_RMDIR | 15 |
| NFS_WRITECACHE | 7 | NFS_READDIR | 16 |
| NFS_WRITE | 8 | NFS_STATFS | 17 |
- (why no lseek?)

CSE506: Operating Systems

9

Ex: NFS_READ Call

```

struct readargs {
    fhandle file;
    unsigned offset;
    unsigned count;
    unsigned totalcount;
};

union readres switch (stat status) {
    case NFS_OK:
        fattr attributes;
        nfsdata data;
    default:
        void;
};

```

CSE506: Operating Systems

10

NFS V.3 (1994)

- TCP and UDP
- 64 byte file handles
- files > 2GB
- ACLs supported
- Kerberos authentication type
- All ops return old/new attributes
 - ◆ saves on most popular call, getattr (update client caches faster)

CSE506: Operating Systems

11

NFS V.3 Protocol

- Removed: ROOT and WRITECACHE
- Added:
 - ◆ READDIRPLUS: 17
 - ✦ also returns file handles
 - ✦ saves on NFS_LOOKUPS
 - ◆ FSSTAT: 18
 - ◆ FSINFO: 19
 - ◆ PATHCONF: 20
 - ◆ COMMIT: 21
 - ✦ Saves cached data to disk

CSE506: Operating Systems

12

NFS V.4 (2004)

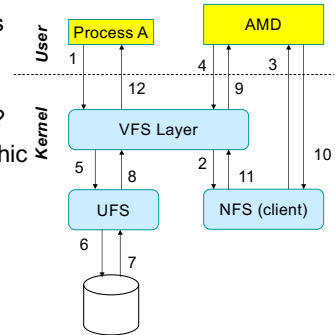
- IETF design, not Sun
- Integrated file locking and mount protocol
- Stronger security w/ negotiation
 - ◆ Public file handles
 - ◆ Works with firewalls & proxies
- Compound operations
- Internationalization
- Better suited for Internet (i.e., WAN)
- Migration and replication
- Extensible protocol

CSE506: Operating Systems

13

User Level NFS-Based File Servers

- Context switches
- extra communication
- Amd dead/hung?
- CFS: cryptographic file server
- FUSE like



CSE506: Operating Systems

14

Resources

- RFC 1094/1813
 - ◆ Usenix papers [Sandberg 84] and [Pawlowski 94]
- NFS V.2/3/4 specs and drafts
 - ◆ ietf.org

CSE506: Operating Systems

15