

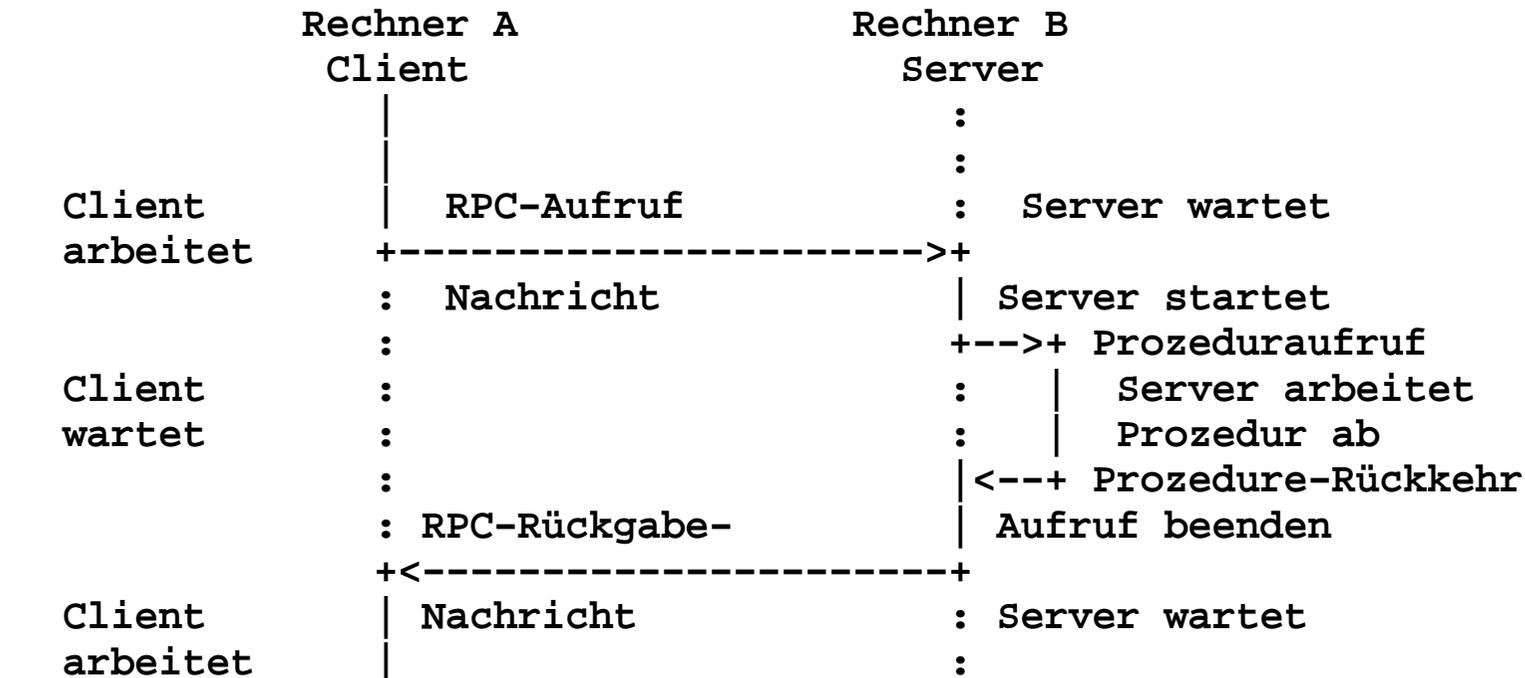
14. NFS - Network File System  
=====

## 1. RPC - Remote Procedure Call

Protokoll der Kommunikationssteuerungsschicht speziell für NFS entwickelt, wird aber auch von anderen Systemen benutzt, z.B. NIS.

Standard: RFC 1050

Funktionsprinzip:



Identifizierung:

Programmnummer

Versionsnummer

Transportprotokoll

Prozedurnummer

z.B.: NIS ypserv

100004, 1 oder 2, udp oder tcp

## 2. Portmapper

dient zur dynamischen Zuordnung von Programmnummern von RPC-Programmen und Ports, d.h. muß vor den Servern gestartet werden, die RPC benutzen (NIS,NFS)

Programme:

```
portmap - SUNOS, DEC-UNIX, HP-UX, LINUX,...
```

```
portmap [-d] [-v]
```

```
rpcbind - Solaris
```

```
rpcbind [-d] [-w]
```

Optionen:

```
-v verbose
```

```
-d debug
```

```
-w warm-start
```

Datenfile:

```
/etc/rpc
```

```
#
```

```
#      rpc
```

```
#
```

```
rpcbind      100000  portmap sunrpc rpcbind
```

```
rstatd      100001  rstat rup perfmeter
```

```
rusersd     100002  rusers
```

```
nfs         100003  nfsprog
```

```
ypserv      100004  ypprog
```

```
mountd      100005  mount showmount
```

```
ypbind      100007
```

```
walld       100008  rwall shutdown
```

```
yppasswdd   100009  yppasswd
```

```
etherstatd  100010  etherstat
```

```
rquotad     100011  rquotaprog quota rquota
```

```
sprayd      100012  spray
```

### 3. Sicherheit

#### 1. NIS

neuere NIS-Server haben eigenes Sicherheitsfile  
Informationen werden nur an Netze bzw. Hosts gegeben,  
die im Sicherheitsfile spezifiziert sind.

```
/var/yp/securenets  
  <Netzwerkmaske> <Netzwerkadresse>
```

z.B.:

```
255.255.0.0 141.20.0.0  
255.255.255.0 141.21.10.0  
255.255.255.255 141.11.5.6
```

#### 2. Sicherheitsportmapper

Alte Portmapper haben auf alle Anfragen reagiert.  
Dadurch leichte Einbrüche.

Neue Portmapper benutzen `tcpd_library`.

Zugriffsberechtigung wird in zwei Files beschrieben:  
erlaubte Zugriffe in:

```
/etc/hosts.allow  
  <daemon-name>:<host-list> | <network-list>
```

z.B.:

```
portmap: 141.20.20.0/255.255.255.0  
portmap: 141.20.21.0/255.255.255.0  
....  
portmap: 141.20.36.0/255.255.255.0  
portmap: 141.20.39.0/255.255.255.0  
ALL: LOCAL .informatik.hu-berlin.de
```

verbotene Zugriffe in:

/etc/hosts.deny

<daemon-name>: <host-liste> : <action>

z.B.:

```
portmap: ALL :( /usr/etc/safe_finger -l @%h | \
  /usr/ucb/mail -s "bellus-%d-%h-%a" bell ) &
```

```
ALL: ALL: (/usr/etc/safe_finger -l @%h | \
  /usr/ucb/mail -s "bellus-%d-%h-%a" bell) &
```

%d - Daemon-Name

%h(%H) - Client(Server)-Name(Adresse)

%a(%A) - Client(Server)-Adresse

**Achtung!!!!**

Portmapper niemals mit kill -9 beenden.

Vorher aktuellen Inhalt retten!!

BSD:

```
pmap_dump >ports.save
```

```
kill
```

```
portmap
```

```
pmap_set <ports.save
```

Solaris:

```
kill SIGHUP <rpcbind> # erzeugt /tmp/portmap.file, /tmp/rpcbind.file
```

```
rpcbind -w # restart
```

Abfragen des Portmappers: rpcinfo

```
rpcinfo      rpcinfo -p [ host ]          - aktuelle Programme
```

```
rpcinfo [ -n portnum ] -u host program [ version ]
  - Aufruf der Procedure 0 (UDP)
```

```
rpcinfo [ -n portnum ] -t host program [ version ]
  - Aufruf Procedure 0 (TCP)
```

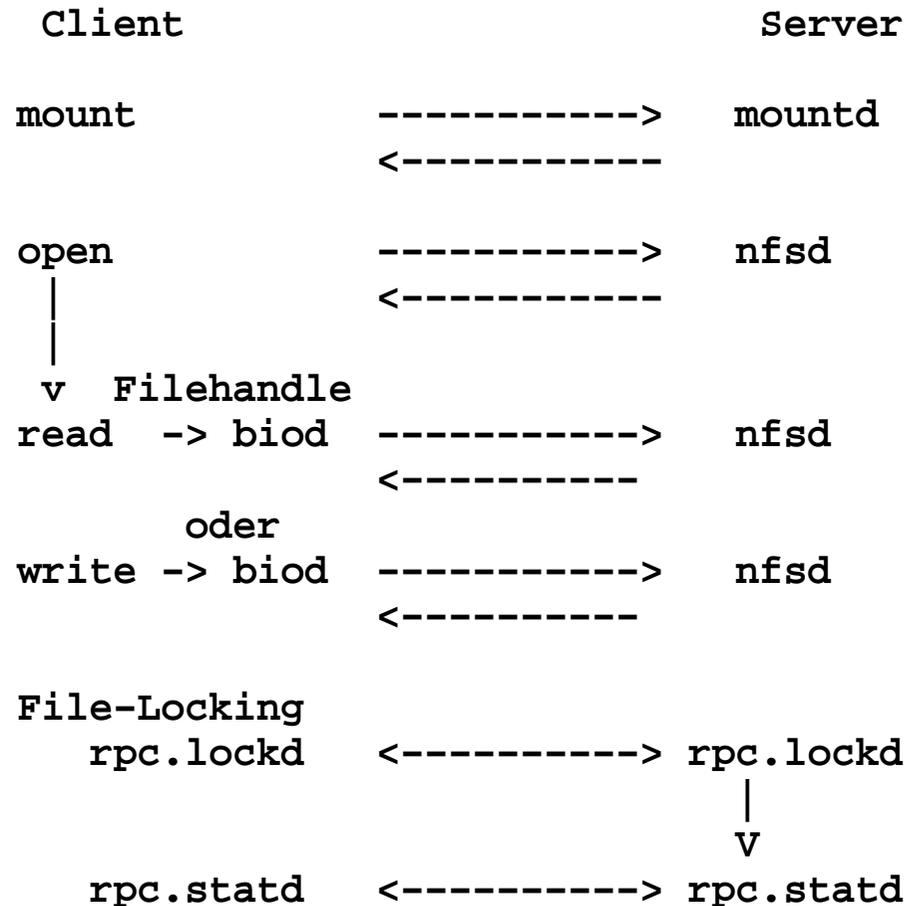
```
rpcinfo -b program version - broadcast für alle Hosts
```

```
rpcinfo -d program version - steichen eines Programms, lokal
```

## 4. NFS

Ziel: Der Nutzer soll keinen Unterschied zwischen lokalen und entfernten Filesystemen sehen.

Ablauf des Zugriffs (mount,read,write)



NFS: mountd: rpc: 10005 , Version: 1,2,3, Prot: UDP, TCP

nfsd: rpc: 10003, Version: 1,2,3,4 Prot: UDP, TCP

NFS Version 4 nur TCP und immer Port 2049 (Firewall fähig)

## 5. NFS-Server

### Mount-Daemon

Mount-Daemon ist ein RPC-Server, der NFS-Requests eines Clienten an einen NFS-Server bearbeitet. Er prüft die Zulässigkeit der Mount-Forderung an Hand entsprechender Files: /etc/exports oder /etc/dfs/sharetab  
Ist die Operation zulässig, wird eine entsprechende Verbindung hergestellt, sonst abgewiesen

Kommando: mountd [-v] [-r]  
-v - verbose  
-r - Mount-Daemon für weitere mounts sperren

### nfsd -Daemon

Dient zur Realisierung der verschiedenen Filezugriffe eines Clienten auf dem NFS-Server. Pro Zugriffsaktion wird eine Instanz von nfsd benötigt.

Kommando: nfsd <Zahl der Instanzen>

Die Zahl der Instanzen wird durch die Zahl der Interface und deren Geschwindigkeit bestimmt.

1 10 MBit/s-Interface - nfsd 8

1 100 MBit/s-Interface - nfsd 16

z.B.: fob (4 100 MBit-Interface) - nfsd 64

**Files:**

/etc/exports (BSD)

Eintrag: <directory> -<option>[,<option>] ...

option> - ro - nur zum Lesen exportieren

rw=host[:host].. - exportieren

zum Lesen und Schreiben an die  
angegebenen Hosts

anon=<UID> - unbekannte Nutzer werden  
als <UID> abgebildet

root=host[:host]... - root-Zugriff für  
angegebene Hosts

access=client[:client]... - Mount Zugriff  
für angegebene Clienten.

Client: Host oder Netzgruppe

default: alle

**Beispiele:**

```
/usr          -access=clients
               # export to my clients

/usr/local
               # export to the world

/usr2         -access=hermes:zip:tutorial
               # export to only these machines

/usr/sun      -root=hermes:zip
               # give root access only to these

/usr/new      -anon=0
               # give all machines root access

/usr/bin      -ro
               # export read-only to everyone

/usr/stuff    -access=zip,anon=-3,ro
               # several options on one line
```

Wird /etc/exports nach dem Start von mountd geändert, so muß das Kommando "exportfs -a" abgesetzt werden.  
 "exportfs" liefert eine Liste der momentan exportierten Direcories.

/etc/dfs/dfstab (Solaris) Quelle für /etc/dfs/sharetab

Umwandlung erfolgt durch das Kommando "shareall"

Das File enthält eine Reihe von "share"-Kommandos.

```
share [ -F <FSType> ] [ -o <option>[,<option>... ]
      [ -d <description> ] [ <pathname> ]
FSType  -  nfs
options -  rw - Lesen und Schreiben für alle
           rw=client[:client]....
           Lesen und Schreiben für Klienten,
           sonst niemand
           ro - Lesen für alle
           ro=client[:client]....
           Lesen für Klienten, sonst niemand
           root=client[:client]...
           Rootzugriff für Klienten
           anon=<UID>
           UID für Unbekannte
```

Beispiel:

```
share -F nfs -o ro /export/exec/Solaris_2.6_sparc.all/usr
share -F nfs -o ro=allfb /opt/SUNWspro
share -F nfs -o rw=allfb -d "" /usr/local
share -F nfs -o ro,anon=0 /opt/Install
share -F nfs -o ro,anon=0 /cdrom/patches_3_0_7_d1
share -F nfs -o rw=star,root=star /export/root/star
share -F nfs -o ro /usr
```

**Solaris: ZFS**

/etc/dfs/dfstab nicht mehr notwendig, aber möglich.

NFS-Export wird über Properties von ZFS für das jeweilige Filesystem aktiviert.

```
zfs set sharenfs= on|off|<Option-List> <Filesystem>
```

```
on    - export an alle
```

```
off   - keins export
```

```
<Option-List> := <Option> [,<Option-List> ]
```

```
<Option>      := <Recht>=<Hostspezifikation>
```

```
<Recht>       := rw | ro | root
```

```
<Hostspezifikation>:=<Hostname>[:<Hostspezifikation>] |  
                    <Netzgruppe>[:<Hostspezifikation>]
```

**Beispiele:**

```
zfs set sharenfs=rw=allfb DATEN/data-1st1
```

```
zfs set sharenfs=rw=allfb DATEN/data-1st1
```

```
zfs set sharenfs=root=bellus:master,rw=allfb DATEN/data-1st1
```

```
zfs set sharenfs=rw=hpetri:hpetriq:bellus:bellus1 DATEN/li
```

**Kommandos:**

```
zfs share -a|<Filesystem>
```

```
zfs unshare -a|<Filesystem>
```

## 6. NFS-Clients

Mittels normalem "mount" wird die Verbindung zum Server hergestellt. Spezielle Optionen für NFS!!!

```
mount -t nfs -o <option>[,<option>...] <host>:<r-dir> <l-dir>
```

Optionen für mount auf der Client-Seite:

- rw/ro - gewünschte Zugriffsrechte
- bg - mount-Operation im Hintergrund ausführen
- rsize=<nn> - Blocklänge lesen
- wsize=<nn> - Blocklänge schreiben
- timeo=<n> - Timeout in Sekunden
- retrans=<n>- Zahl der Wiederholungen - transport
- retry=<n> - Zahl der Wiederholungen
- port=<n> - Port-nummer
- vers=<n> - NFS-Version
- soft - Soft-Mounten
- hard - Hard-Mounten
- intr - Übertragung unterbrechbar

biod - Daemon für asynchrone Block I/O bei NFS.  
dient der Beschleunigung der NFS-Zugriffe

```
biod <Anzahl der Daemonen>
```

Anzahl der Daemonen - siehe nfsd (8..16)

## 7. Locking

Über NFS ist auch ein File-Locking möglich. Dazu werden auf der Serverseite und auf der Clientseite jeweils ein Lock-Daemon (rpc.lockd) und ein Status-Daemon (rpc.statd) benötigt.

Wirkungsweise:

```
lock (client) --> rpc.lockd (client)
rpc.lockd(client) --> rpc.lockd(server)
rpc.lockd(server) --> rpc.statd(server)
                        Eintrag in /etc/sm (host)
                        Eintrag in /etc/sm.bak (host+lock)
rpc.statd(server) <--> rpc.statd(client)
```

Solaris: /var/statmon/sm und /var/statmon/sm.bak

### Error-Recovery bei einem Crash

Server rpc.statd kontaktiert die rpc.statd der Klienten entsprechend den Eintragungen in /etc/sm.bak

Maßnahmen bei Fehler: rpc.statd cannot talk to statd at xxxx

```
ps -aux
kill -9 <rpc.statd>
rm /etc/sm.bak/*
rpc.statd
```

## 8. Automounter

Automounter dient zum automatischen mounten über NFS bei Bedarf. Das entlastet wesentlich das Netz und die Server, da nicht ständig alles gemountet sein muß.

Der Automounter wird über Maps gesteuert:

**indirekte Maps:** Alle gemounteten Filesysteme werden in einem Directory untergebracht. Die zu mountenden Filesysteme sind nicht sichtbar, wenn sie nicht gemountet sind

**Beispiel:**

|          |          |                     |
|----------|----------|---------------------|
| fob-vol1 | -rw,soft | fob:/export/volume1 |
| fob-vol2 | -rw,soft | fob:/export/volume2 |
| fob-vol3 | -rw,soft | fob:/export/volume3 |
| fob-vol4 | -rw,soft | fob:/export/volume4 |
| fob-vol5 | -rw,soft | fob:/export/volume5 |

**direkte Maps:** Die gemounteten Filesysteme werden einzeln in den Directory-Baum eingehängt. Sie sind immer sichtbar.

**Beispiel:**

|               |          |                      |
|---------------|----------|----------------------|
| /usr/local    | -rw,soft | knecht:/usr/local    |
| /opt/SUNWspro | -rw,soft | knecht:/opt/SUNWspro |
| /cdrom        | -ro,soft | knecht:/mnt/cdrom    |

Der Automounter kann über die Kommandozeile oder über die Mastermap gesteuert werden. Wird der Name einer Map ohne führenden "/" angegeben, wird die Map über das NIS/LDAP geladen.

**Master-Map:**

```
<Mount-directory> <indirect Map> <Mount-Optionen>
/-                <direct Map>  <Mount-Optionen>
```

```
automount [-mnTv] [-D name=value] [-f master-file]
          [-M mount-directory] [-tl duration] [-tm interval]
          [-tw interval] [directory map [-mount-options]] ...
```

**Defaults:**

```
/etc/auto.master
/etc/auto.home
```

**Beispiele:**

```
automount /vol auto.home      # indirekt
automount /- /etc/auto.home   # direkt
```

Um eine einfache Verwaltung der Maps zu erreichen, sollte auf jeden Fall NIS oder LDAP für die Verwaltung der Maps eingesetzt werden.

**Anzeigen der mit NIS verwalteteten Maps:**

```
ypcat -k auto.home
ypcat -k auto.master
ldapcat auto.home      # nur bei Informatik
ldapcat auto.master    #
```

## 9. Starten von NFS-Server, NFS-Clienten und Automounter

### Linux

#### NFS-Server:

- /etc/init.d/nfsserver
- /etc/sysconfig/autofs
- /etc/sysconfig/nfs
- /etc/exports
- /etc/autofs\_ldap\_aut.conf (wenn LDAP benutzt)

#### NFS-Client:

- /etc/init.d/nfs
- /etc/sysconfig/nfs

#### Automounter:

- /etc/init.d/autofs
- /etc/sysconfig/autofs
- /etc/autofs\_ldap\_aut.conf (wenn LDAP benutzt)
- /etc/auto.master
- /etc/auto.home

**Solaris bis 2.9**

```
/etc/init.d/nfs.server
  /etc/dfs/dfstab
/etc/init.d/nfs.client
/etc/init.d/autofs
  /etc/auto_master
  /etc/auto_home
```

**Solaris ab 2.10****UFS**

```
/etc/dfs/dfstab
/etc/default/nfs
/etc/auto_master
/etc/auto_home
```

```
svcadm enable svc:/network/nfs/server:default
svcadm enable svc:/network/nfs/client:default
svcadm enable svc:/system/filesystem/autofs:default
```

**ZFS**

```
zfs set sharenfs="rw=host1:host2:netzgruppe" DATEN/export
```

## 10. Fehlersuche

rpcinfo - siehe oben

showmount - Anzeigen von gemounteten Filesystemen

/usr/sbin/showmount [-adehv] [--all] [--directories]

                  [--exports] [--help] [--version] [host]

-a or --all                   Ausgabe von Host und Directoy

-d or --directories           Ausgabe von Directory

-e or --exports              Ausgabe der Export-Liste

-h or --help                 Hilfe

-v or --version              Version

nfsstat - Anzeigen des Zustands von NFS (Server und Client)

/usr/sbin/nfsstat [-anrcsz] [-o facility] ... (Linux)

-s           Server

-c           Client

-n           Statistik

-r           RPC-Statistik

-z           setze Kernelstatistik auf 0

-o facility   Ausgabe von speziellen Statistiken  
              nfs, rpc, net, fh, rc (nicht bei Solaris)

netstat - Anzeigen des Zustands der Interfacekarten

Optionen (Linux):

-i   Statistik der Interfacekarten

-r   Routen anzeigen

-n   keine Übersetzung der IP-Adressen in Namen

-a   alles

-t   aktive TCP-Verbindungen (Solaris: -P tcp)

-u   aktive UDP-Verbindungen (Solaris: -P udp)

## Firewall

Die notwendigen Ports können durch Firewall gesperrt sein. Bis NFS 3 werden die Ports dynamisch verteilt, d.h. sie stehen erst nach dem Starten des NFS-Servers fest und müssen dann im Firewall für die entsprechenden Clients frei geschaltet werden. Ab NFS 4 erfolgt der Austausch nicht mehr über RPC sondern über den festen Port 2049 (nfsd)!!

### Solaris:

```
rpcinfo -p
# interessante Ports: rpcbind, status, nlockmgr rstatd,
#                       rquotad, nfs, mountd
ipfstat -i | egrep "rpc|nfs"
```

### Linux:

```
rpcinfo -p
# interessante Ports: portmapper, mountd, status, nfs, nlockmgr
iptables -L -n | egrep "111|2049"
```