



NETWORK FILE SYSTEM (NFS)

By:

Amruta Udamale

Sequence

- Introduction to NFS
- Version
- NFS basic and protocol- State ,Architecture
- Component
- Different NFS processes
- Working
- Read operation while accessing file from server
- Steps to mount directory from NFS server

Introduction to NFS

- Developed by Sun Microsystems
- Use Distributed File Protocol
- Mostly used with UNIX OS
- Implement for other OS like
 - Mac OS, Microsoft Windows, Novell NetWare, and IBM AS/400
- Allows users to 'Share' a directory
- RPC-based (uses XDR)
- NFS is perfect for a NAS
- Opposed to SAMBA
- Currently based on VFS
- Major Goals
 - simple crash recovery
 - reasonable performance :80% of a local drive

Sun Protocols in the ISO OSI Model

7	Application	Mail	RCP	Rlogin	RSH
		FTP	NFS	YP	Telnet
6	Presentation	XDR			
5	Session	RPC			
4	Transport	TCP		UDP	
3	Network	IP (Internetwork)			
2	Data Link	Ethernet	Point-to Point	IEEE 802.2	
1	Physical	Ethernet	Point-to Point	IEEE 802.3	

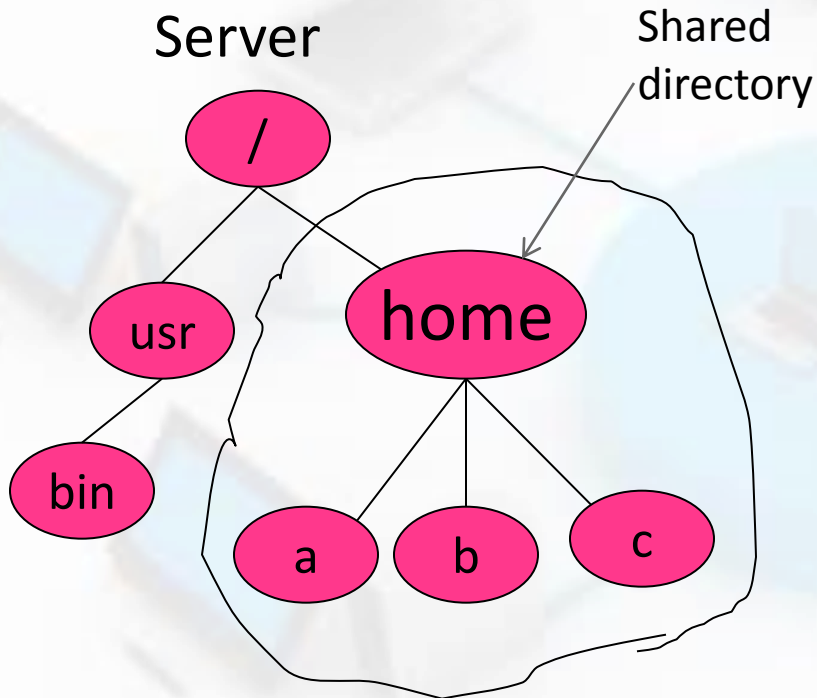


Sun's Native Architecture

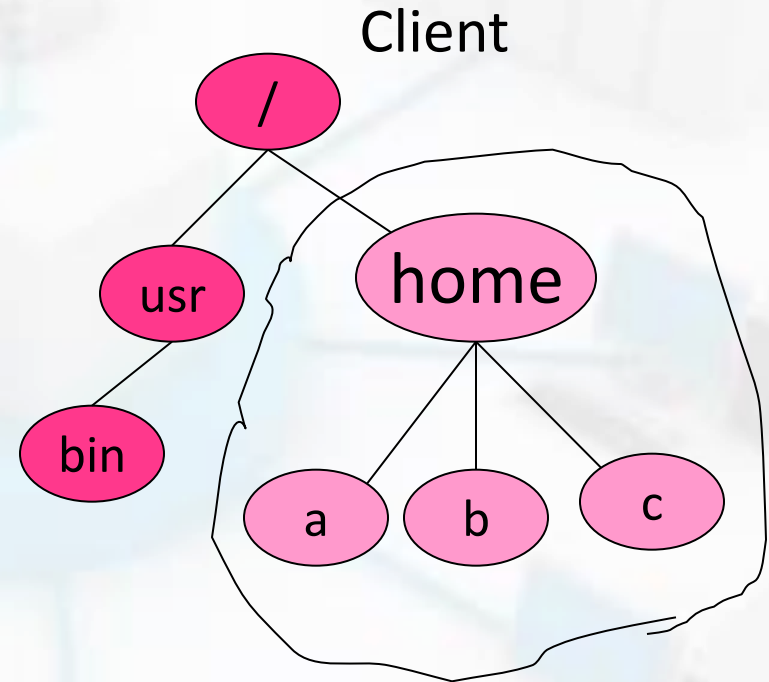


Future Additions

NFS Overview



NFS: Export Subtree



NFS: Mount a remote file system

The clients 'mount' the shared directory, it becomes part of their own directory structure.

NFS Version

- NFSv1 (1984)
 - develop in-house experimental purpose
- NFSv2 (March 1989)
 - Release for commercial use
 - UDP, Stateless, 32 bit
- NFSv3 (June 1995)
 - Overcome to v2
 - UDP and TCP, stateless, 64bit
- NFSv4 (April 2003)
 - Overcome to v2, v3
 - Statefull Server
 - Use tcp
 - Focus on Performance, Accessibility, Scalability, Strong security, Internationalization, Extensibility

System Structure: Server Type

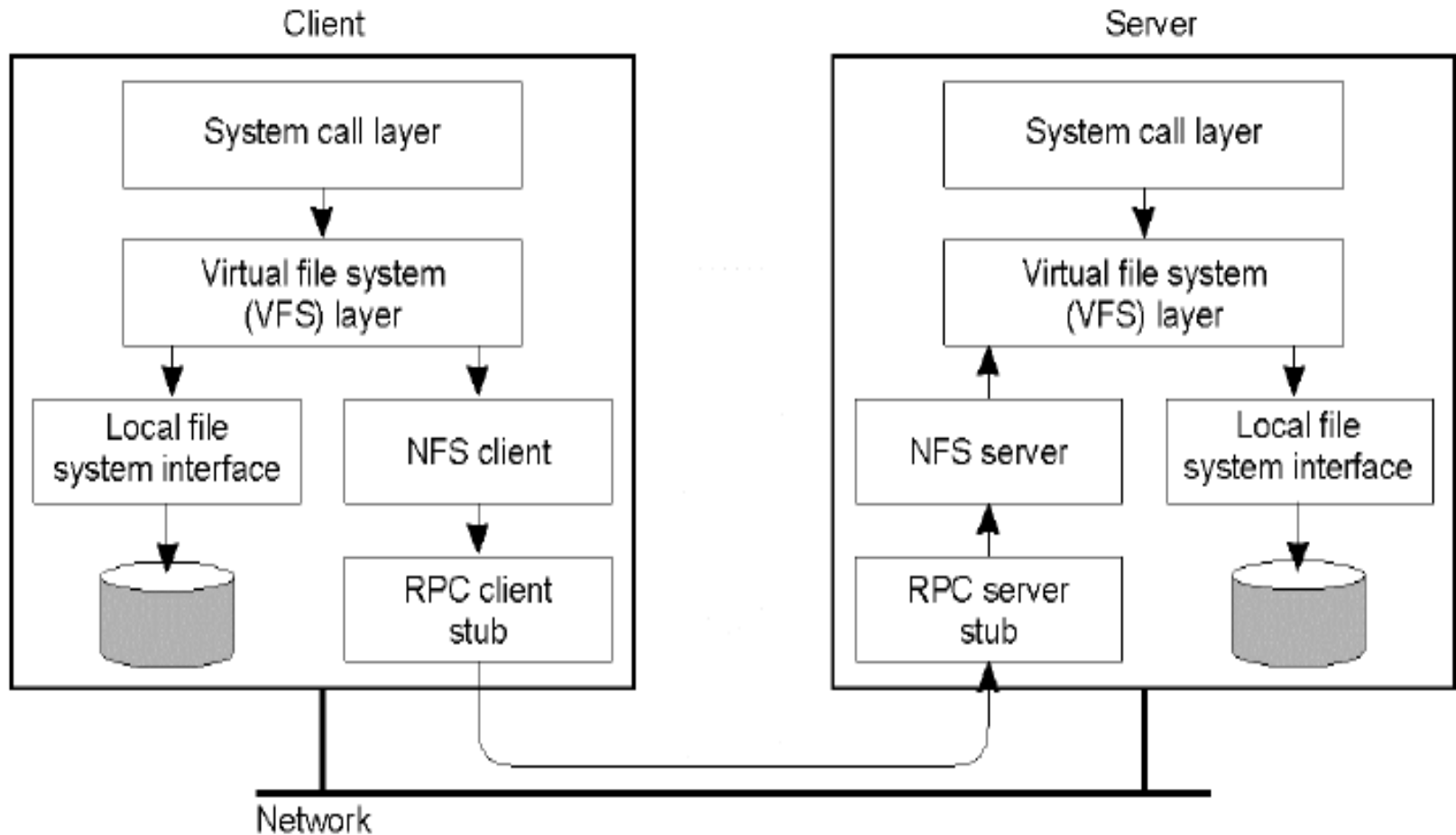
- Stateless server

- No information kept for client requests
- Information needed to be provided by the client
- More tolerant to server crashes

- Statefull server

- maintins information about client Access
- Shorted request messages
- Better performance
- Idempotency easier
- Consistency is easier to achieve

NFS Architecture



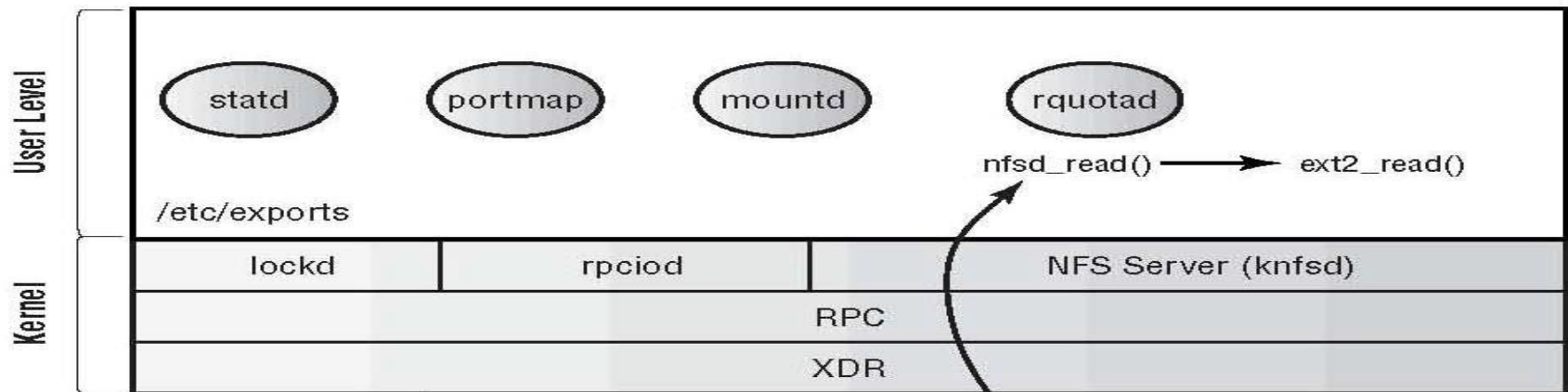
- VFS to handle local and remote files common interface
- RPC for data transport

Sun NFS Protocol

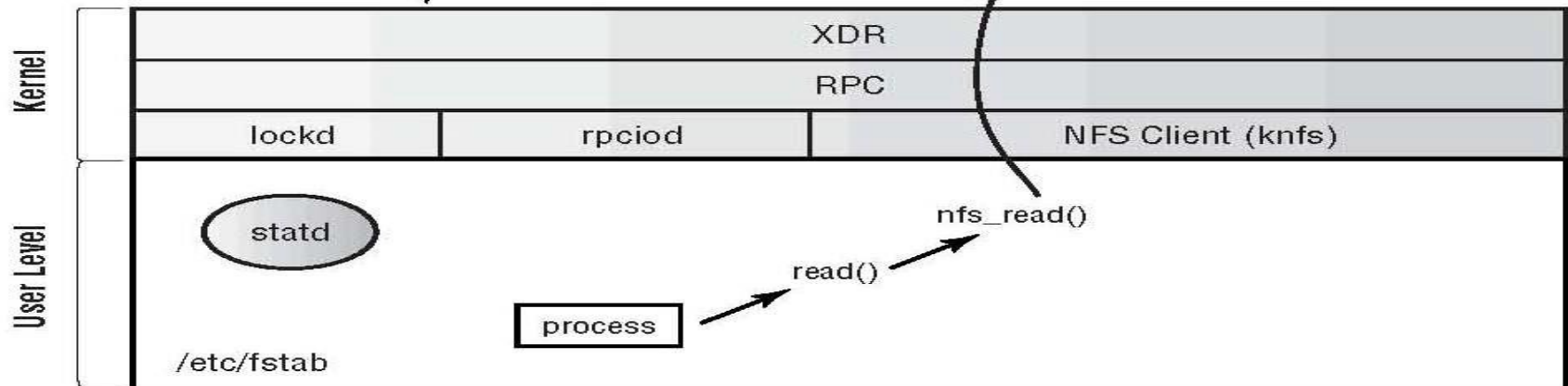
Operation	v3	v4	Description
Create	Yes	No	Create a regular file
Create	No	Yes	Create a nonregular file
Link	Yes	Yes	Create a hard link to a file
Symlink	Yes	No	Create a symbolic link to a file
Mkdir	Yes	No	Create a subdirectory in a given directory
Mknod	Yes	No	Create a special file
Rename	Yes	Yes	Change the name of a file
Rmdir	Yes	No	Remove an empty subdirectory from a directory
Open	No	Yes	Open a file
Close	No	Yes	Close a file
Lookup	Yes	Yes	Look up a file by means of a file name
Readdir	Yes	Yes	Read the entries in a directory
Readlink	Yes	Yes	Read the path name stored in a symbolic link
Getattr	Yes	Yes	Read the attribute values for a file
Setattr	Yes	Yes	Set one or more attribute values for a file
Read	Yes	Yes	Read the data contained in a file
Write	Yes	Yes	Write data to a file

Component

NFS Server



Network

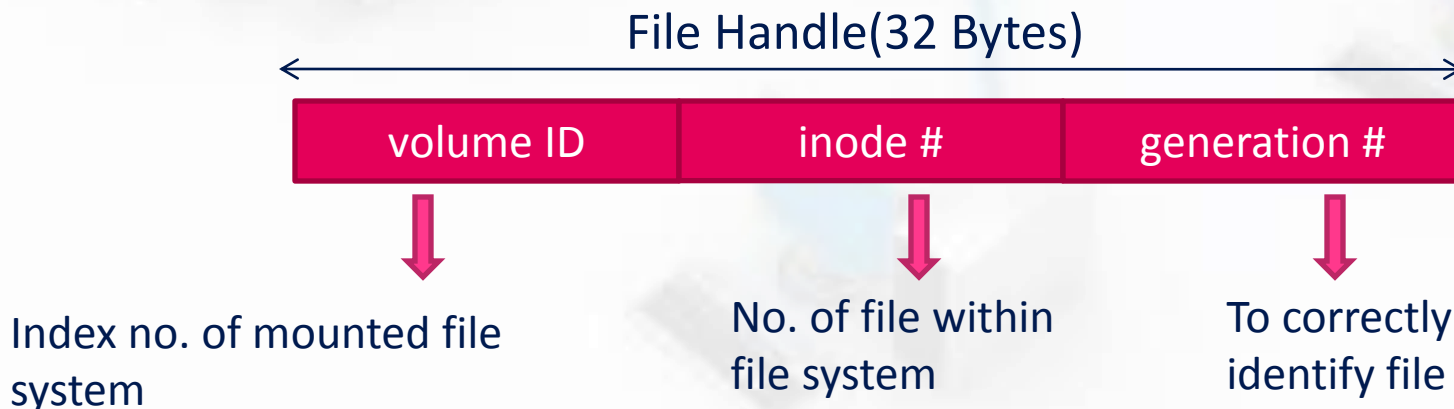


NFS Client

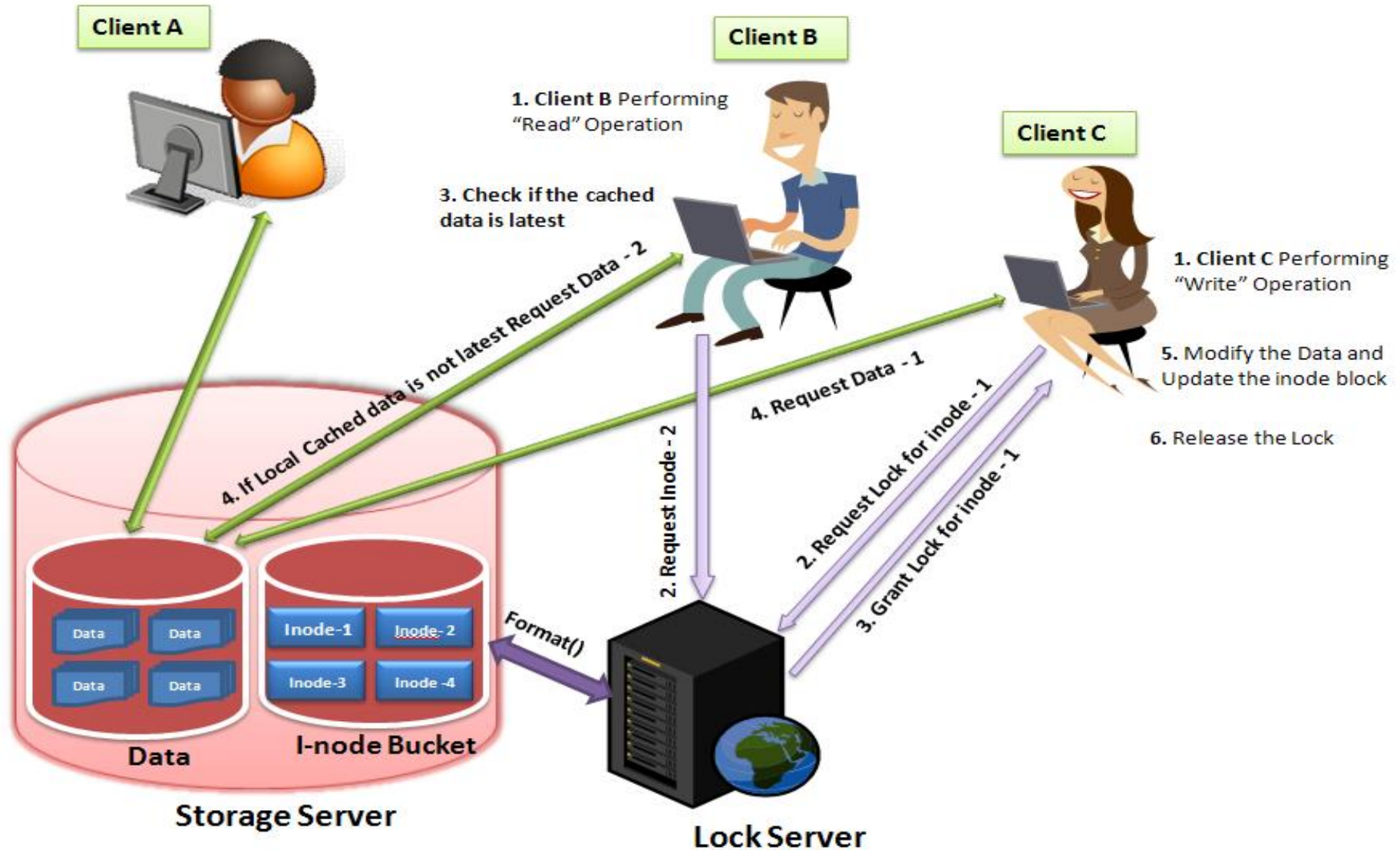
1. File Handles

❖ How does the client tell the server which file or directory the operation applies to? Similarly, how does the server return the result of a lookup?

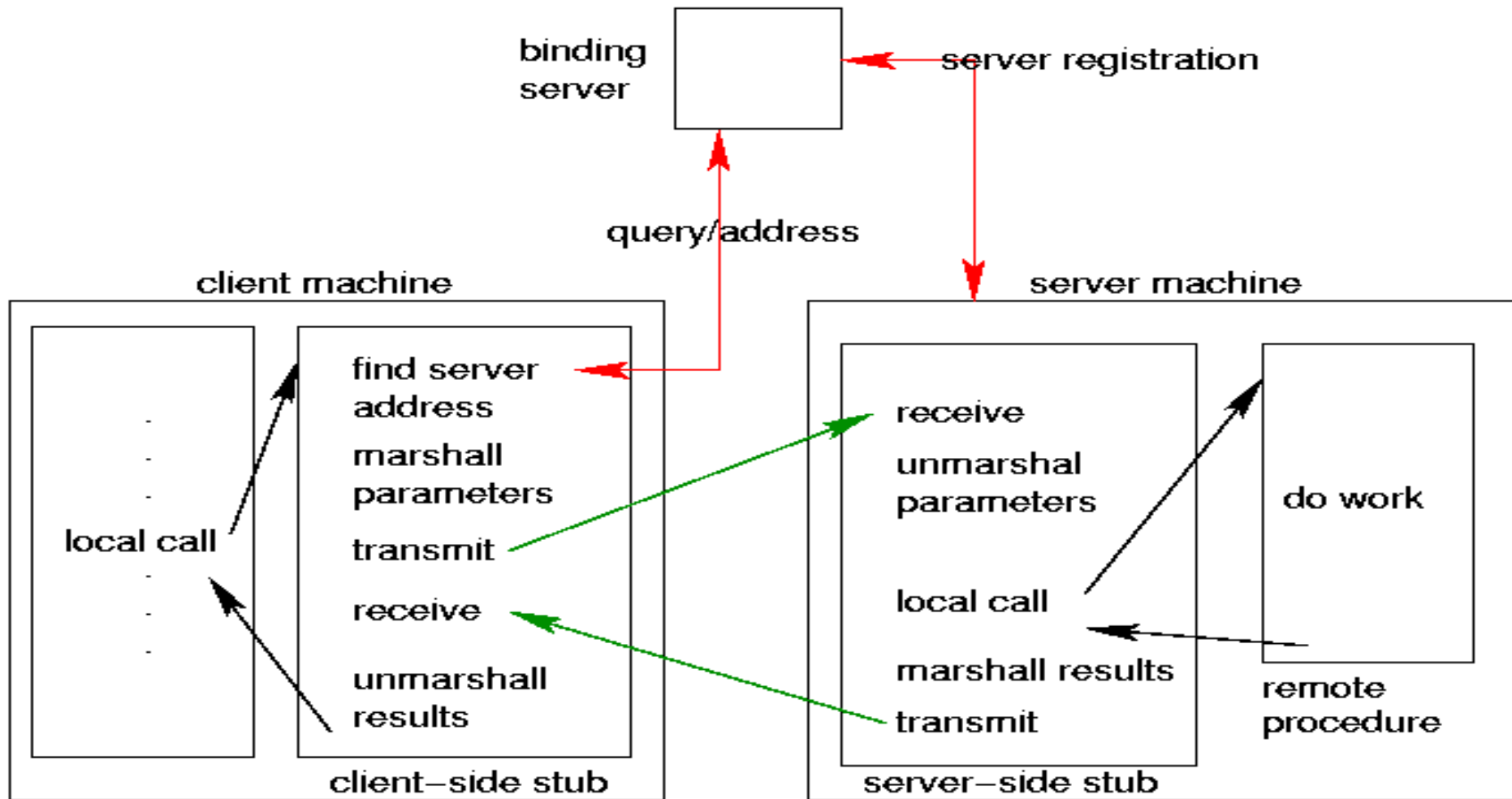
- File handle or fhandle
- Determined by the server
- Includes all information needed to identify the file/object on the server, and find it quickly.
- Client shouldn't modify it
- Stale File Handle



Use of File handle

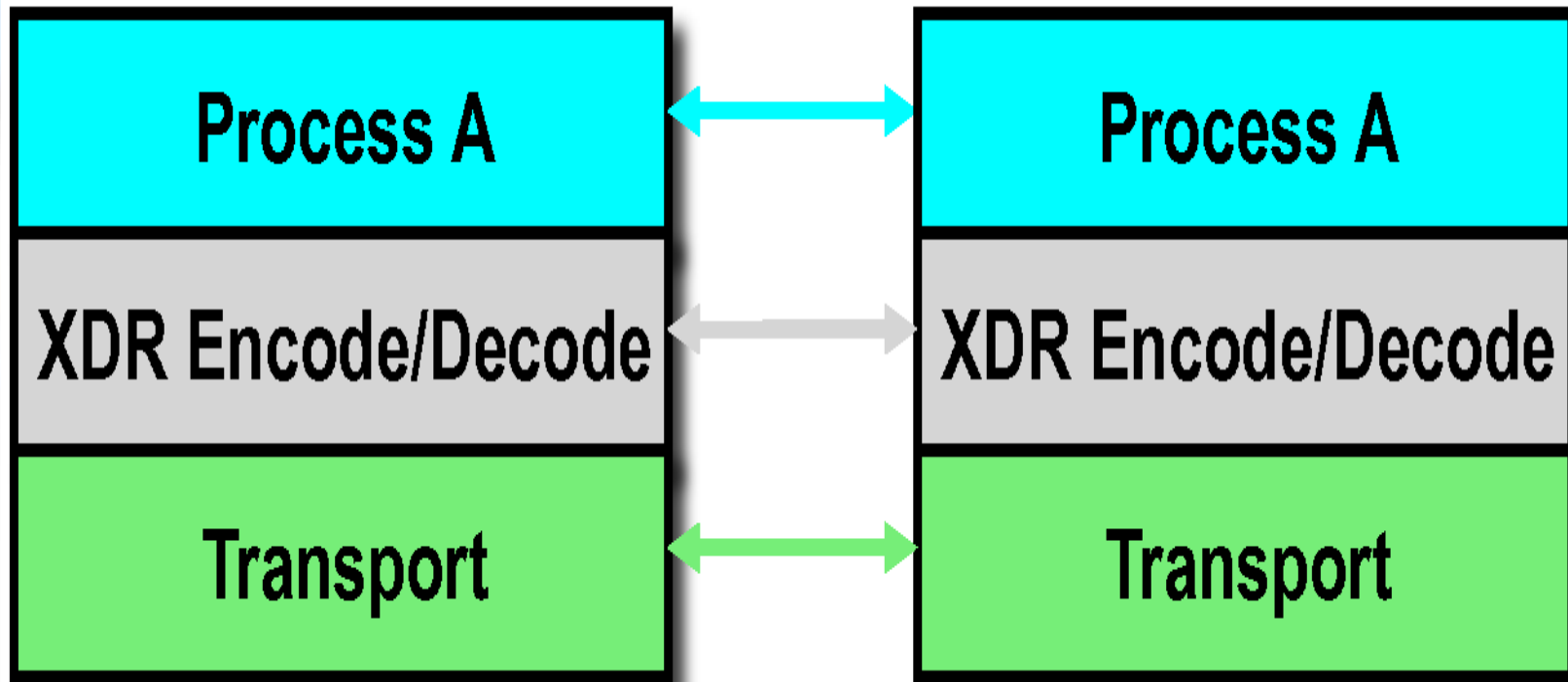


2. Remote Procedure Call (RPC)

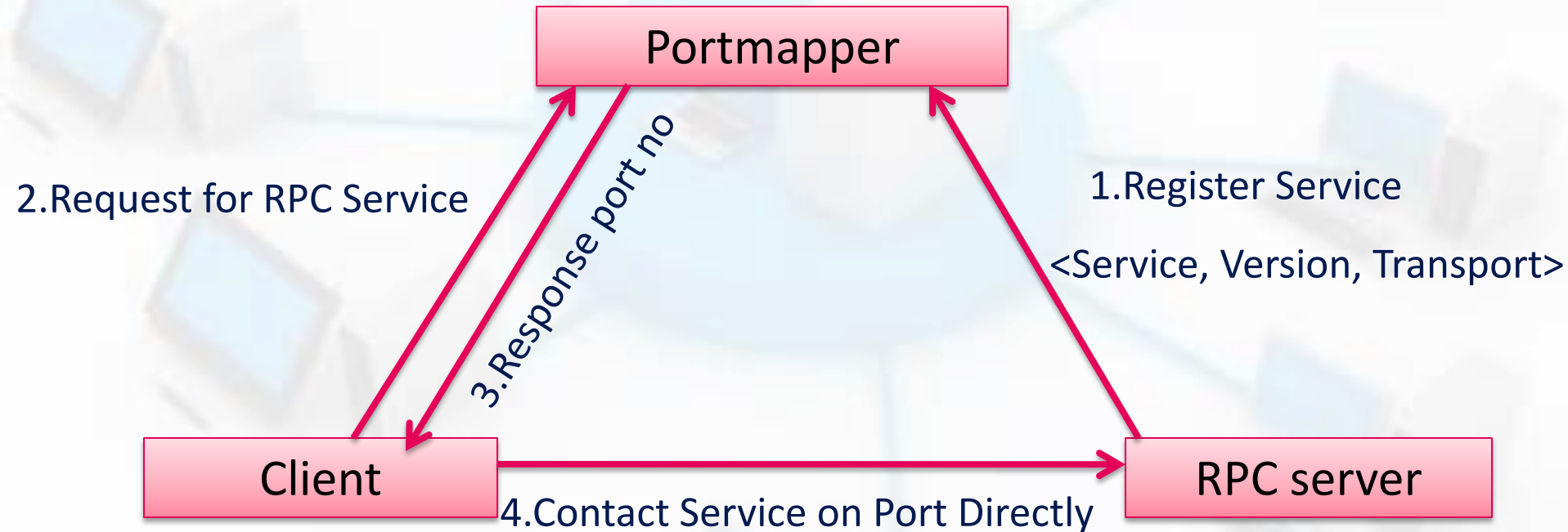


- Initially implementations of RPC was used with the UDP protocol
- Today both UDP and TCP are used

3. External Data Representation (XDR)



4. Portmapper



5. NFS Daemon Processes

- a. **nfsd** daemon
- b. **biod** daemon
- c. **rpc.mount** daemon
- d. **rpc.lockd** daemon
- e. **rpc.statd** daemon

nfsd Daemon

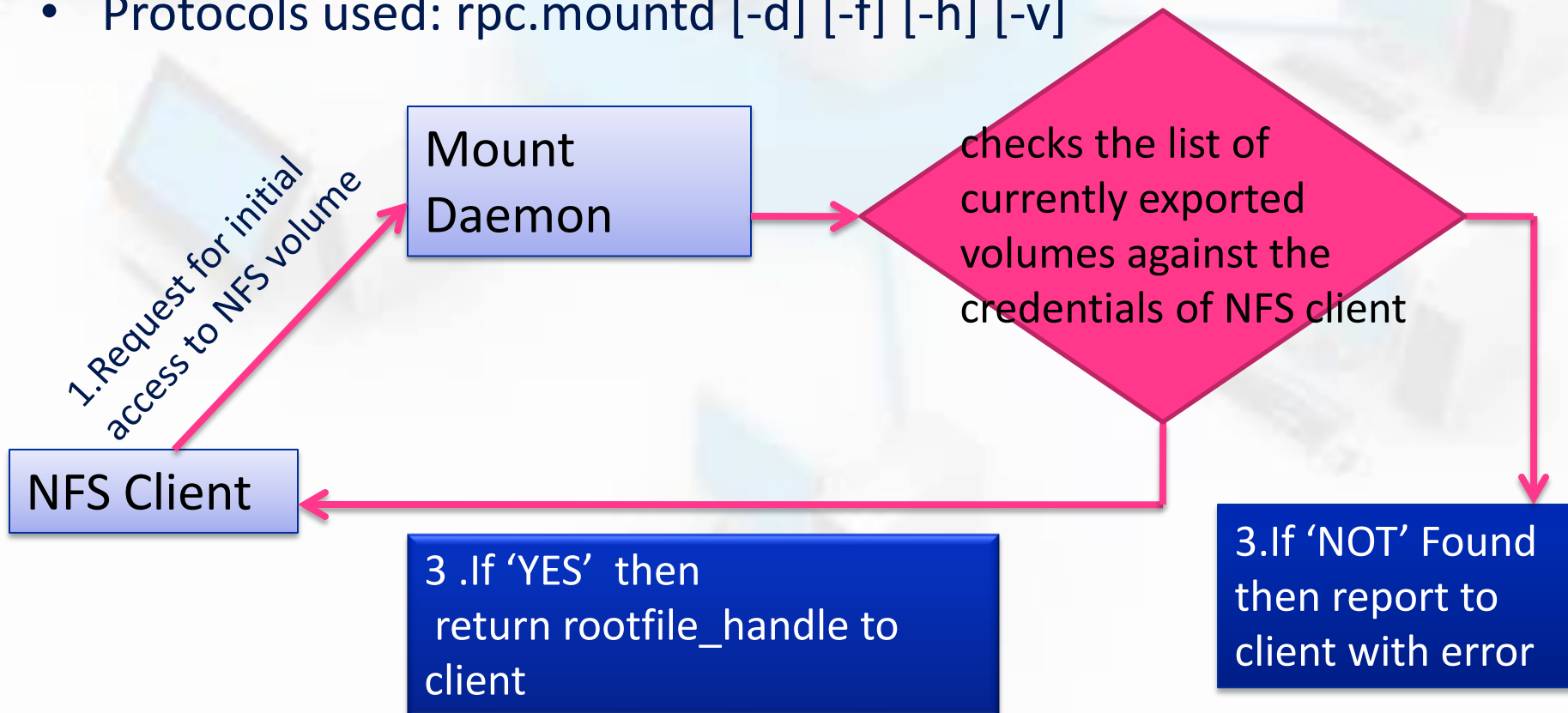
- Handle incoming RPC requests
- May have multiple **nfsd** daemons per site
- A nfsd daemon makes kernel calls to do the real work
- Allows multiple threads

biod Daemon

- Does readahead for clients
 - To make use of kernel file buffer cache
- Only improves performance
 - NFS works correctly without biod daemon
- Also flushes buffered writes for clients

rpc.mount Daemon

- server implements the MOUNT protocol
- Provides initial file handle for a remote volume
- Also checks that incoming requests are from privileged ports (in UDP/IP packet source address)
- Protocols used: `rpc.mountd [-d] [-f] [-h] [-v]`



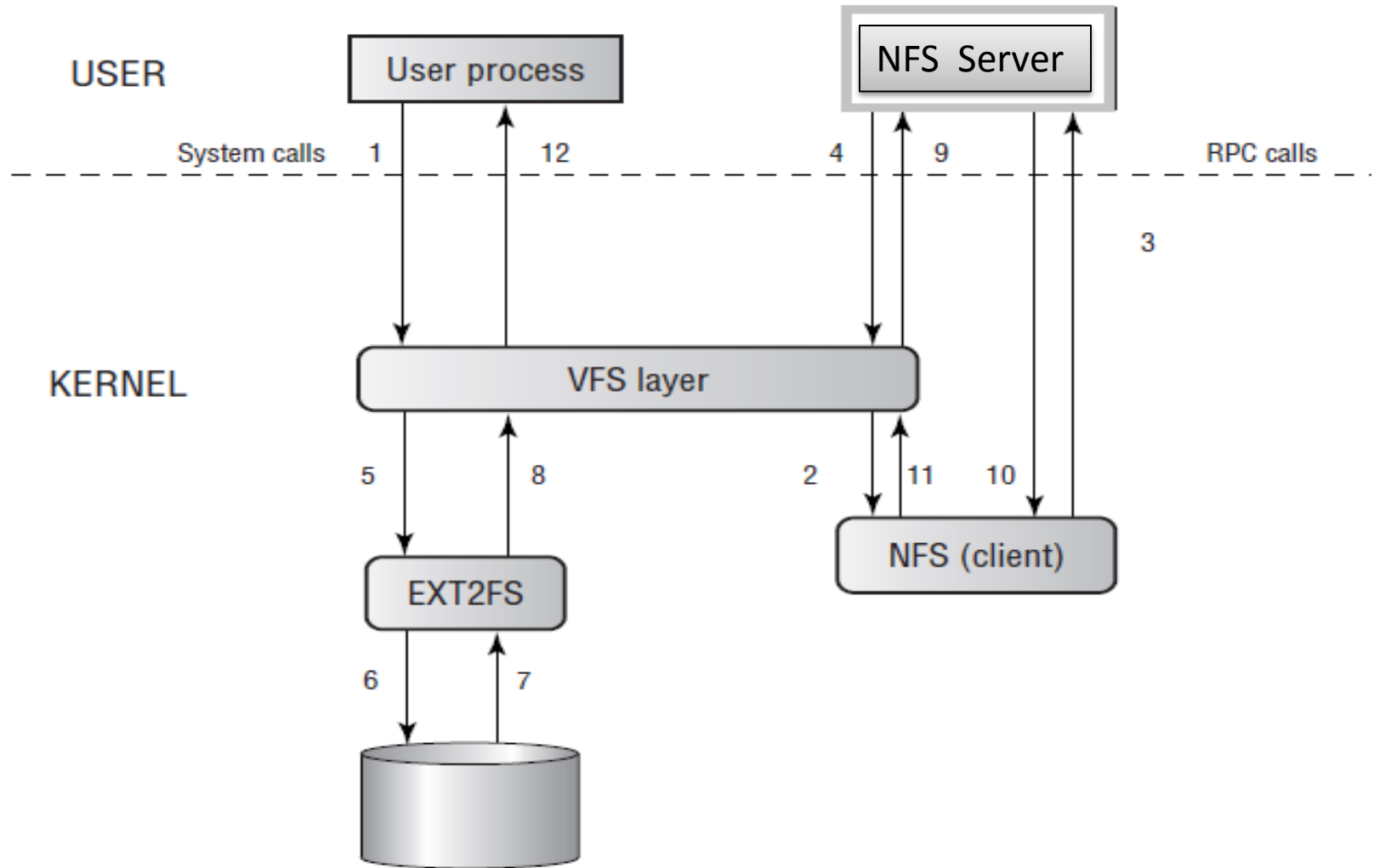
rpc.lockd Daemon

- NFS server is stateless, so it does not handle file locking
- rpc.lockd provides locking
- server crashes and recovers : rpc.lockd reestablish locks
- Use NFS Lock Manager (NLM) Protocol
- Procedure used:
NLM_NULL, NLM_TEST, NLM_LOCK,
NLM_GRANTED NLM_UNLOCK, NLM_FREE

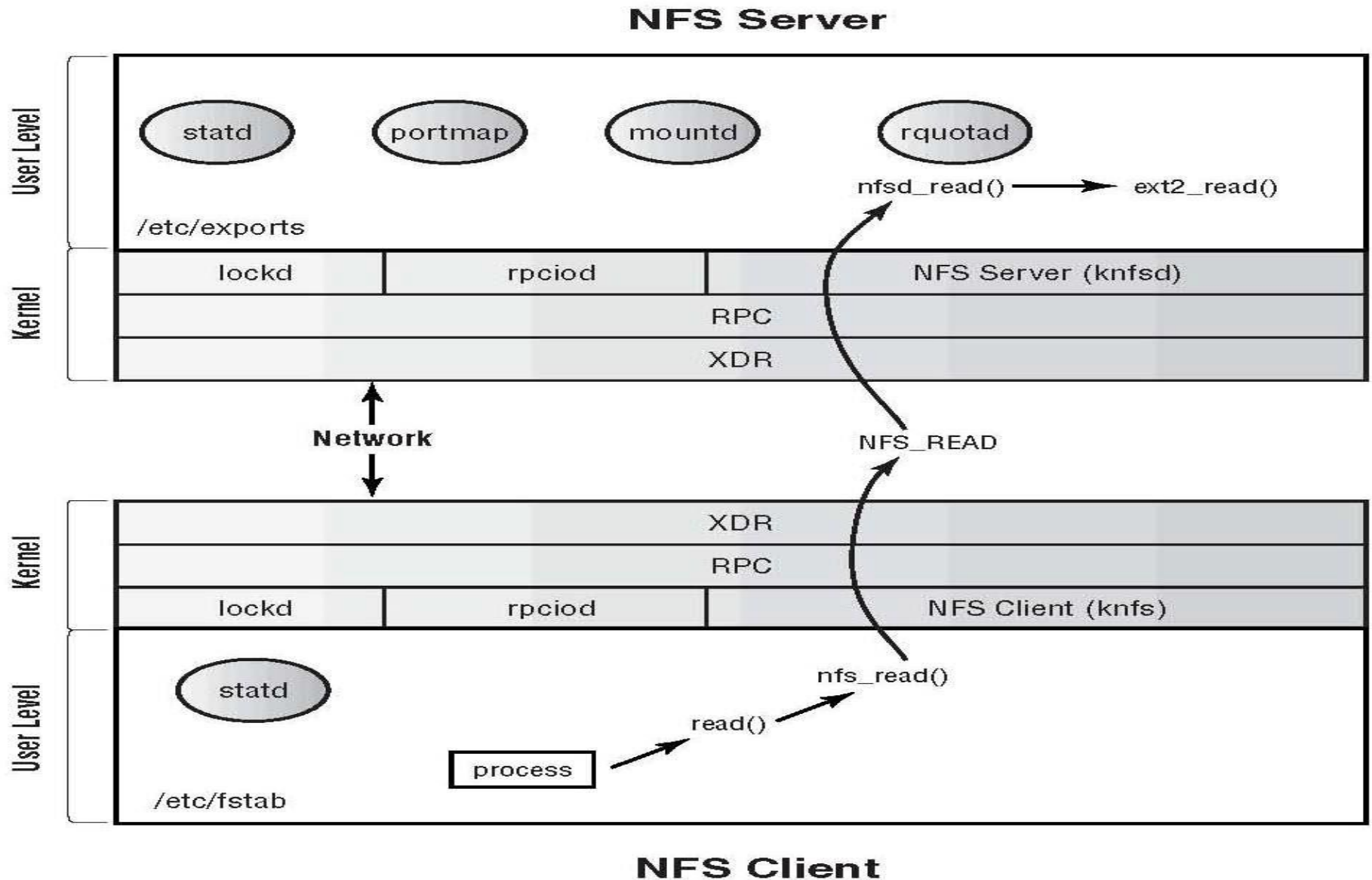
rpc.statd Daemon

- at client and server
- Used to check status of a machine
- Server's rpc.lockd asks rpc.statd to store lock info (in file system)
 - And to monitor status of locking machine
- If client crashes, clear its locks from server
- client crashes: rpc.statd contacts client when it becomes available again
- Use Network Status Monitor (NSM) Protocol
- Procedure used:
SM_NULL,SM_STAT,SM_MON,SM_NOTIFY

Working



Read Operation From Server



Steps to mount directory from NFS server

root@ubuntu: /home/amruta

4:38 AM amruta

```
root@ubuntu:/home/amruta# apt-get install nfs-kernel-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-kernel-server is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:/home/amruta#
```

Server Installation

amruta@ubuntu: ~

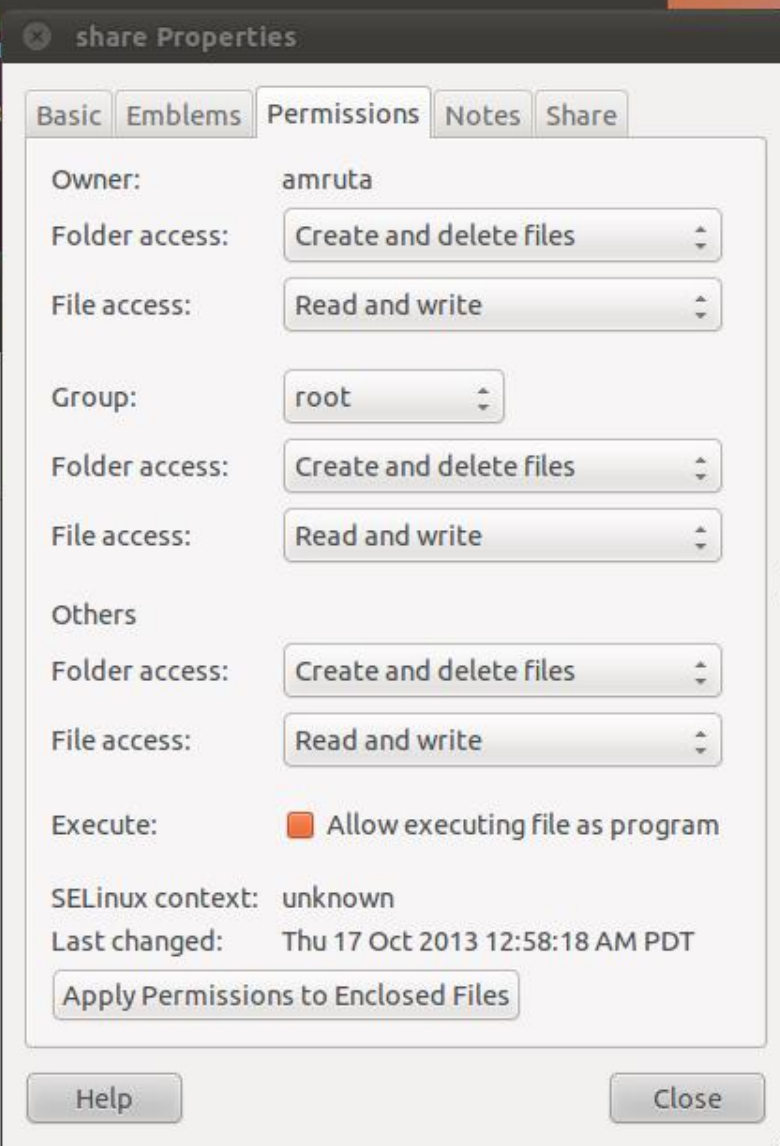
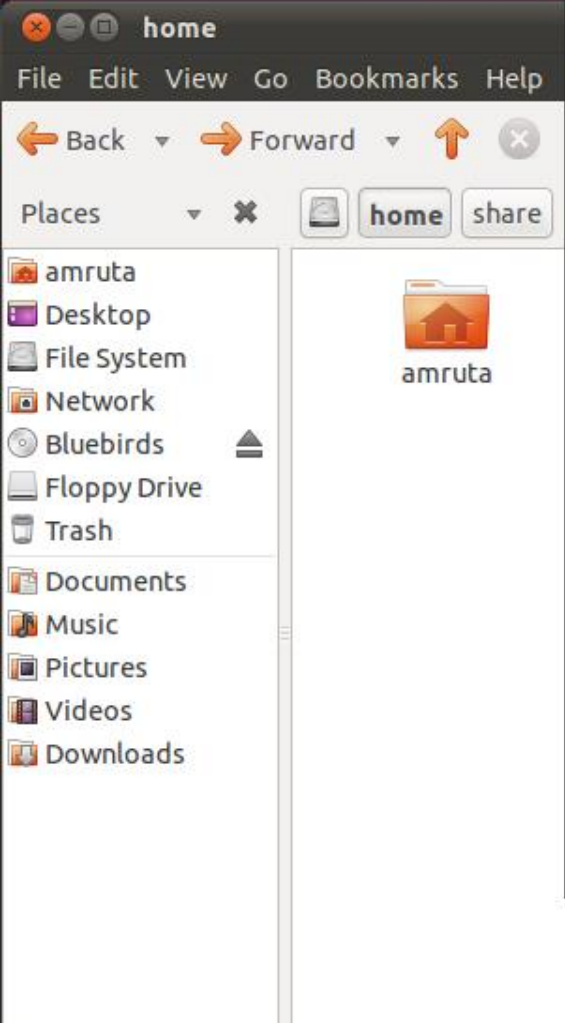
File Edit View Search Terminal Help

```
amruta@ubuntu:~$ sudo mkdir /home/share  
[sudo] password for amruta:  
amruta@ubuntu:~$ sudo chown amruta /home/share  
amruta@ubuntu:~$
```

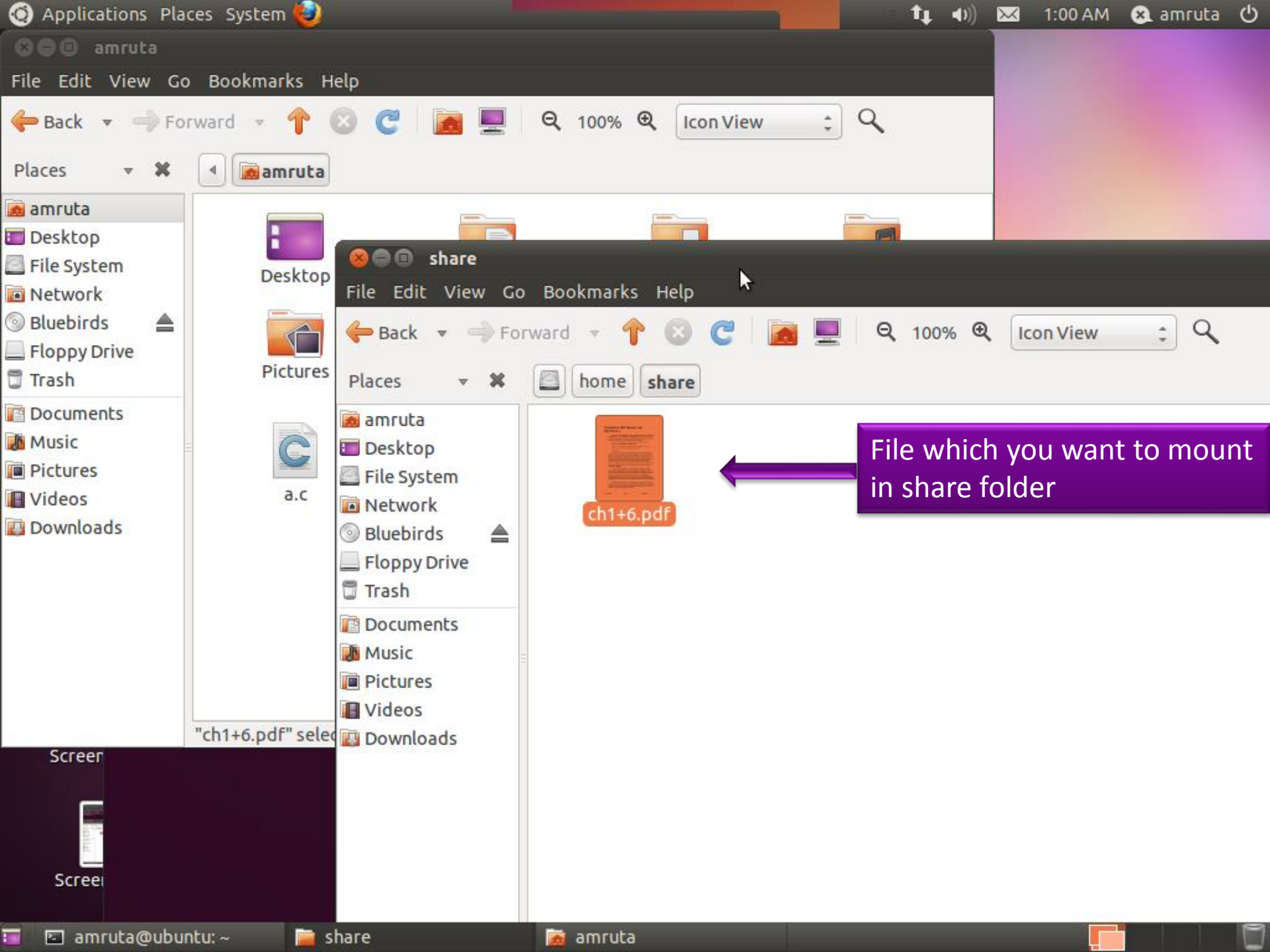
Creating Share Folder in
Home directory



```
amruta@ubuntu: ~  
File Edit View Search Terminal Help  
amruta@ubuntu:~$ sudo mkdir /home/share  
[sudo] password for amruta:  
amruta@ubuntu:~$ sudo chown amruta /home/share  
amruta@ubuntu:~$
```



Setting Folder's Properties



File which you want to mount
in share folder



pictures

```
amruta@ubuntu: ~  
amruta@ubuntu:~$ sudo mkdir /home/share  
[sudo] password for amruta:  
mkdir: cannot create directory `/home/share': File exists  
amruta@ubuntu:~$ sudo chown amruta /home/share/  
amruta@ubuntu:~$ sudo chown amruta /home/share1/  
chown: cannot access `/home/share1/': No such file or directory  
amruta@ubuntu:~$ sudo mkdir /home/share1  
amruta@ubuntu:~$ sudo chown amruta /home/share1/  
amruta@ubuntu:~$ sudo gedit /etc/exports
```

exports (/etc) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

exports ✕

```
# /etc/exports: the access control list for filesystems which may be  
exported  
#  
# to NFS clients. See exports(5).  
#  
# Example for NFSv2 and NFSv3:  
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2  
(ro,sync,no_subtree_check)  
#  
# Example for NFSv4:  
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)  
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)  
#
```




*exports x

```
# /etc/exports: table of access control list for filesystems which may be
exported
#
#           to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes          hostname1(rw,sync,no_subtree_check) hostname2
(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4           gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes     gss/krb5i(rw,sync,no_subtree_check)
#
/home/share 192.168.0.0/255.255.255.0(ro,sync)
```

Edit File with IP address



Reload



Mark All Upgrades



Apply



Properties

Quick filter

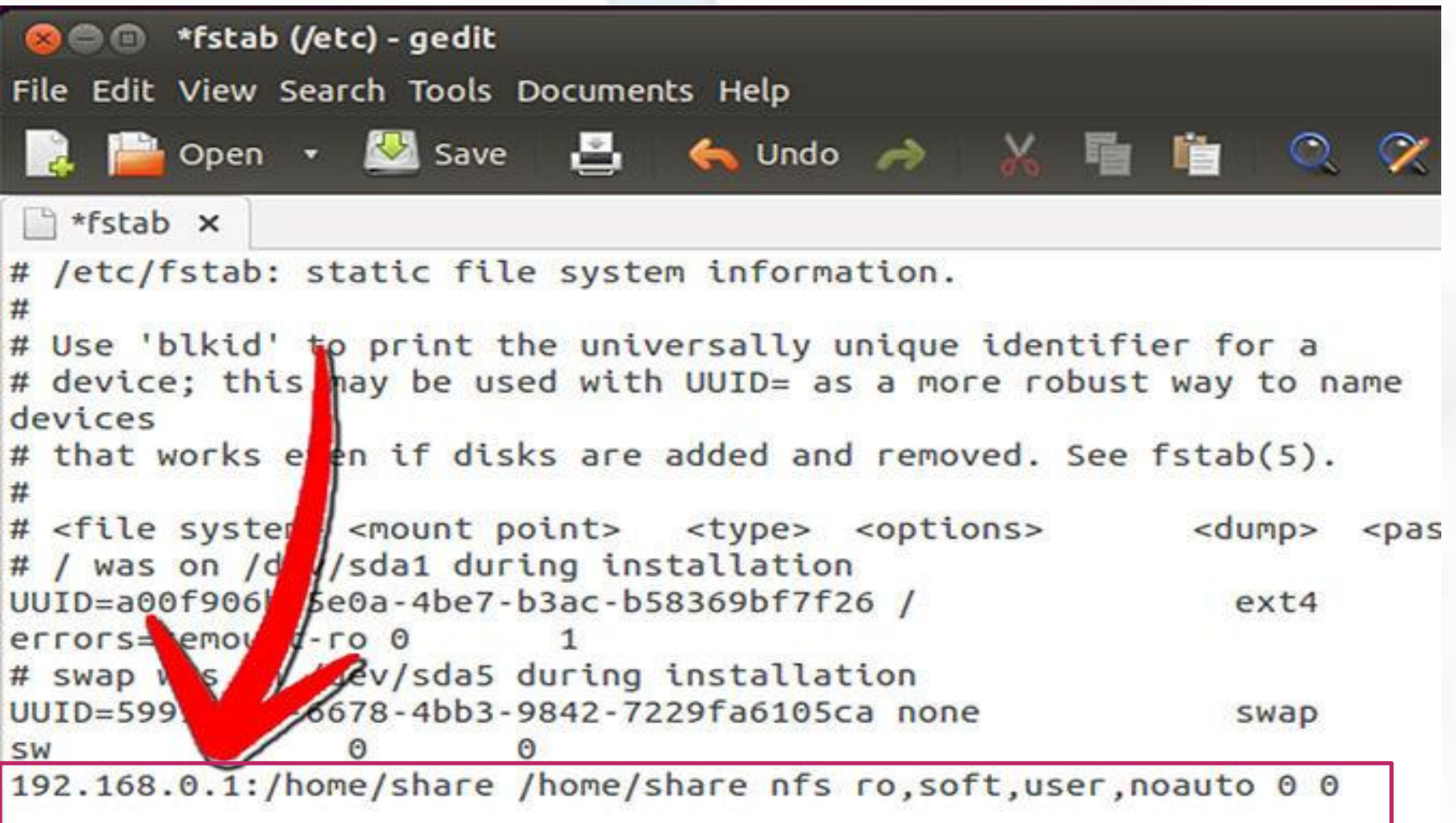
nfs-co

Select Packages For
Mounting Purpose

S	Package	Installed Version
<input type="checkbox"/>	nfswatch	
<input checked="" type="checkbox"/>	nfs-kernel-server	1:1.2.6-3ubuntu2
<input checked="" type="checkbox"/>	nfs-common	1:1.2.6-3ubuntu2
<input type="checkbox"/>	jftp	
<input type="checkbox"/>	python-flufl.lock-doc	
<input type="checkbox"/>	ruby-lockfile	
<input type="checkbox"/>	libfile-nfslock-perl	
<input type="checkbox"/>	sbrshd	
<input type="checkbox"/>	kdenetwork-filesharing	

NFS support files common to client and server

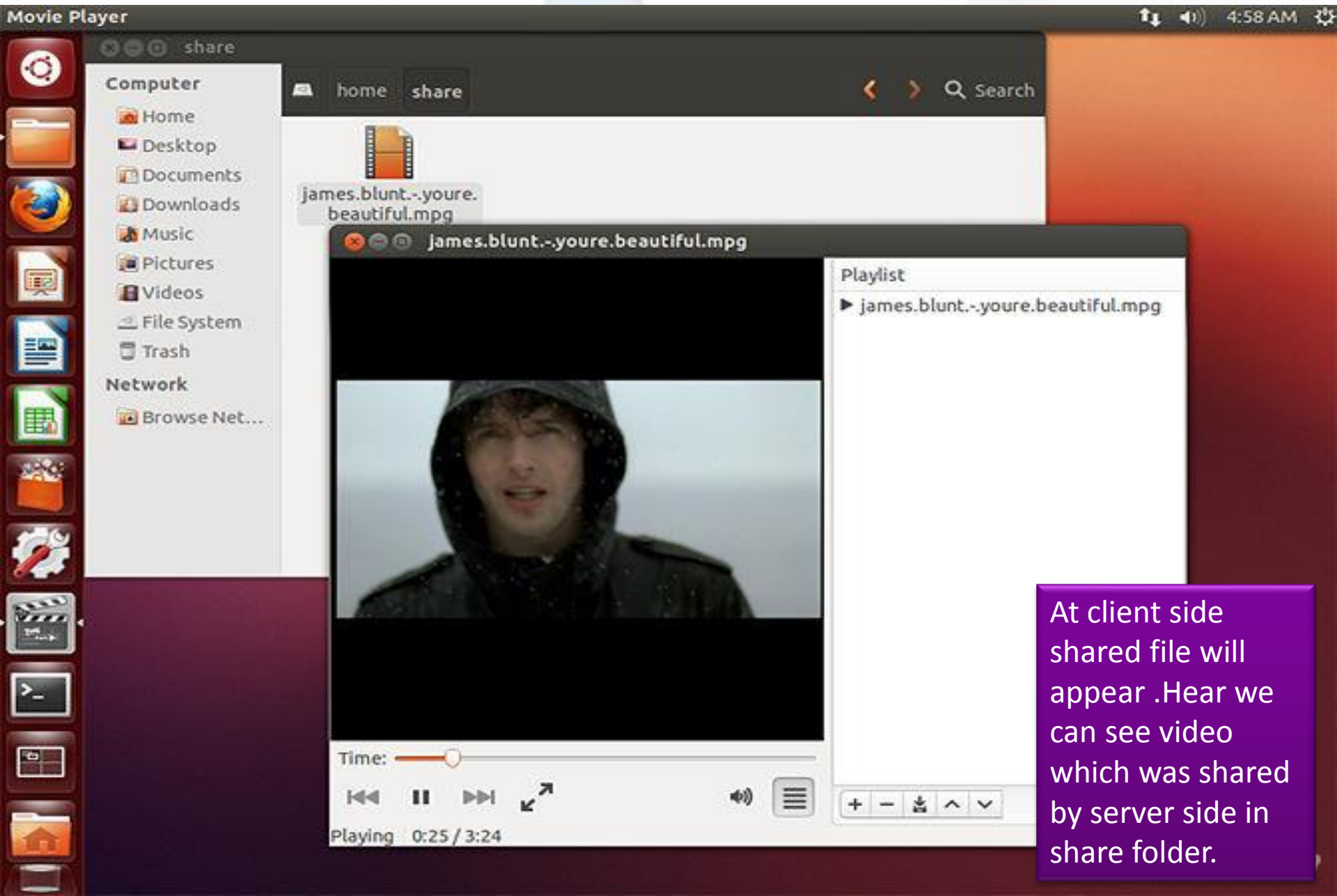
After wards do on cmd :sudo getit /etc/fstab



```
*fstab (/etc) - gedit
File Edit View Search Tools Documents Help

*fstab x
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name
# devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=a00f9061-5e0a-4be7-b3ac-b58369bf7f26 / ext4
errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=5991-6678-4bb3-9842-7229fa6105ca none swap
sw 0 0
192.168.0.1:/home/share /home/share nfs ro,soft,user,noauto 0 0
```

For output :mount /home/share1



At client side shared file will appear .Hear we can see video which was shared by server side in share folder.

Conclusions

- It is ubiquitous.
- It is easy to setup and administer.
- It provides a better solution than the alternative of not sharing files.
- NFS v4 improves security.
- Alternatives of NFS :
 - Server Message Block (SMB, also known as CIFS) protocol
 - Apple Filing Protocol (AFP)
 - NetWare Core Protocol (NCP)
 - OS/400 File Server file system (QFileSvr.400)

References

- http://en.wikipedia.org/wiki/Network_File_System
- <http://www.cs.sunysb.edu/~ezk/cse595-s13/nfs/ch1+6.pdf>
- <http://lass.cs.umass.edu/~shenoy/courses/spring07/lectures/Lec20.pdf>
- <http://www.tldp.org/HOWTO/pdf/NFS-HOWTO.pdf>

A network diagram with a central blue globe connected to various computer icons. The globe is in the center, and several lines radiate from it to different computer setups, including desktop monitors, laptops, and server racks, arranged in a circular pattern around the globe.

Thank You !!