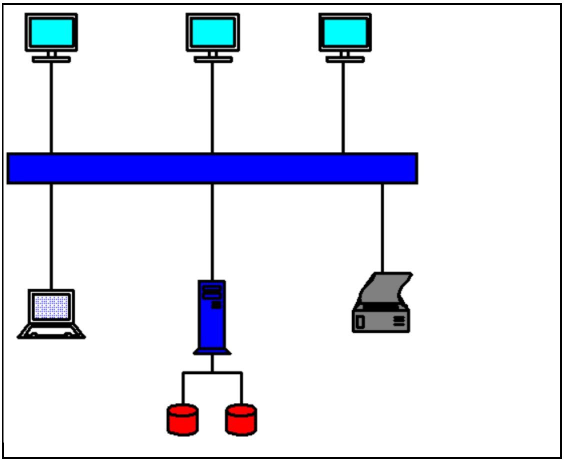
Network File System

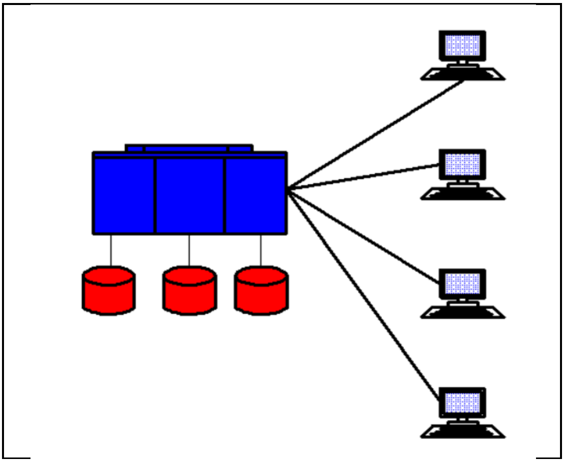
Network service that allows transparent sharing of file systems or directories between nodes on a network

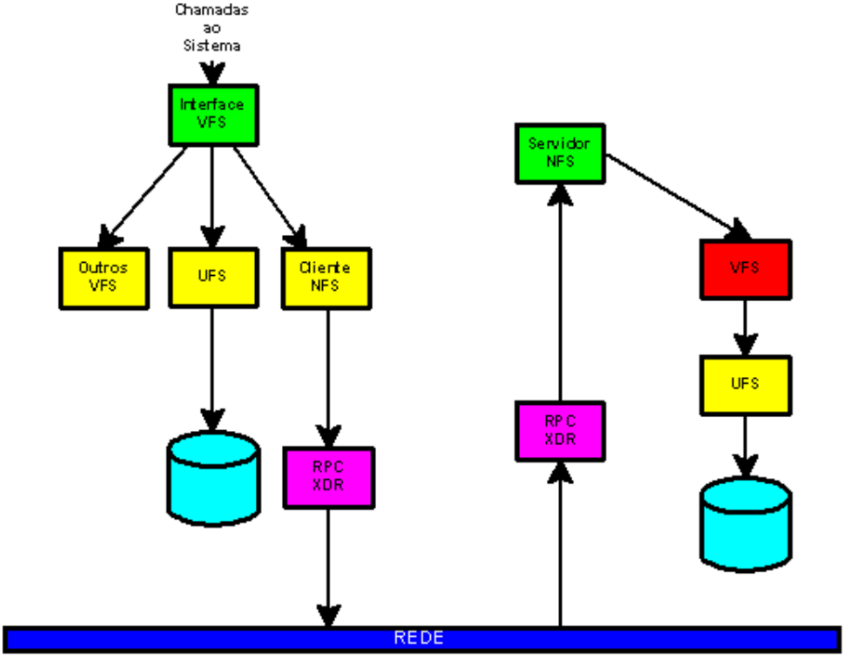
Implemented using RPC (Remote Procedure Call), whose protocols are described using XDR (eXternal Data Representation)

Centralized Environment Distributed Environment



NFS – Implementation



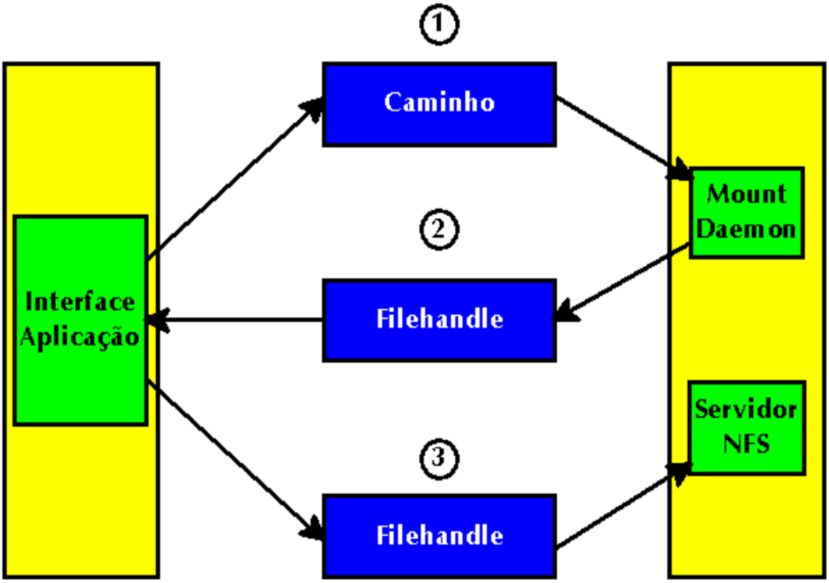


Features

The NFS implementation offers five types of transparency:

- Types of file systems
- Location of file systems
- operating system type
- machine type
- network type

Mount Protocol



1. Client sends the path of a file or directory to the NFS server

2. The server responds by returning the corresponding filehandle

3. Client sends filehandle to NFS server

NFS server

- Export file systems to make their use transparent to NFS client applications
- Reads or writes files in response to requests from NFS clients
- Does not keep information related to files opened by customers
- Can serve NFS clients or other servers
- Does not cache write requests from NFS clients

NFS Client

- Mount shared file systems exported by NFS server
- Files are read or written via requests to the NFS server

- Keep all information about your files open

- You can use the services of multiple NFS servers

- Can communicate with NFS servers through nodes on Ethernet

- Cache for recording

Client/Server Interaction

- **/etc/exports** determines the file systems to be exported

- The **exportfs** command makes file systems available

- The **/etc/xtab** file contains the list of exported file systems

- Server starts NFS daemon (**nfsd**) and daemon to mount (**mountd**)

- The server daemon **rpc.mountd** returns a pointer *(file handle)* to the requested directory or file system

- The client's *file handle* is placed in the kernel's mount table

- All further references are passed to the NFS daemon running on the server using the client's file handle

- NFS read ahead/write behind is managed by block i/o daemon (**biod**)

Export from File Systems

Export options can be applied to file systems or directories

- **ro** - export file system as read-only
- **rw=host[:host2]** - export file system as read-only for most and as read-write for specified hosts
- **access=host1:[host2]** - allows specified hosts mount access to the file system
- **root=host1[:host2]** - gives the listed hosts root access to the file system
- **anon=userid** assigns this userid to any request from an unknown user that is not rejected

Export from File Systems

`/etc/exports`

```
/usr
/usr/kvm
/home/obelix               -access=falbala:asterix
/export/root/falbala       -access=root,root=falbala
/export/swap/falbala       -access=falbala,root=falbala
```

A directory will not be exported if:

- is a directory above or below an already exported one

- It is within the same file system (on the same partition)

/usr/etc/exportfs

```
# exportfs -a
```

exports all file systems listed in the **/etc/exports** file

```
# exportfs -u /home/obelix
```

removes the file **/home/obelix** from the list of exported files

```
# exportfs -o access=falbala /home/src
```

allow the falbala machine access to the **/home/src** directory

```
# exportfs


/usr/lib
/home/src
/export/exec/
/export/root/falbala         -root=falbala,access=falbala
/export/swap/falbala         -root=falbala,access=falbala
```

Informational Commands

```
# showmount falbala

      obelix
      panoramix
      abracurcix

# showmount -e obelix

      export list for obelix
      /usr                    (everyone)
      /export/root/falbala    falbala
      /home                   falbala
```

Importing File Systems

`/usr/etc/mount`

Mount options can be applied to file systems or directories

- **rw|ro** - Read/Write (default) or read-only
- **hard** - Try mount until server responds
- **soft** - Returns an error if the server does not respond
- **bg** - If the first attempt fails, try again in the background
- **timeo=n** - Set nfs timeout to n tenths of a second (default=7)
- **intr** - Allow mount interrupts via keyboard
- **secure** - Uses a more secure protocol for NFS transactions
- **rsize=n** - Sets a size equal to n bytes for the read buffer (default=8192)
- **wsize=n** - Sets a size equal to n bytes for the write buffer (default=8192)

/etc/fstab

The /etc/fstab file contains a list of file systems to be automatically mounted at boot

	mount	mount	mount	dump	fsck
device	point	type	options	freq	order
/dev/sd0a	/	4.2	rw	1	1
/dev/sd0g	/usr	4.2	rw	1	2
obelix:/usr	/mnt	nfs	ro,bg	0	0

/usr/etc/mount

mount request via nfs

```
      # mount -tvo nfs ro,soft,bg obelix:/usr /mnt
```

List mounted file systems

```
     # mount
```

List mounted file systems in /etc/fstab file format

```
      mount -p
```

Save current configuration in /etc/fstab file

```
      # mount -p >> /etc/fstab
```

Remount all file systems

```
      # mount -a
```

Unmount a filesystem

```
    # umount /usr/share/man
```

NOTAS

The /usr/etc/mount command appends a particular file system to the directory structure. The directory specified in the mount command must previously exist. If the directory has any content before the mount, it will remain hidden until the filesystem is unmounted again.

In order for an nfs client to effectively mount a file system, the nfs server will need to be exporting it. If the server specifies who can import or mount the requested file system, the nfs client must be included in this list. The nfs server must also have the rpc.mountd and nfsd daemons running; the nfs client needs to have the biod daemon running. Additional mount options:

-t Specifies the file system type.

-v Verbose. Display message for each mounted file system
-o Specifies file system options.
options They are a list of words separated by commas.