NFS Client
- Application
- Client Shell

NFS Server
- Port Mapper
- Mountd Daemon
- NFSD Daemon

This diagram was generated with EventStudio System Designer (http://www.EventHelix.com/EventStudio)

Copyright (c) 2021 EventHelix.com Inc. All Rights Reserved.

This sequence diagram describes mounting, opening and reading of a file via the NFS (Network File System).
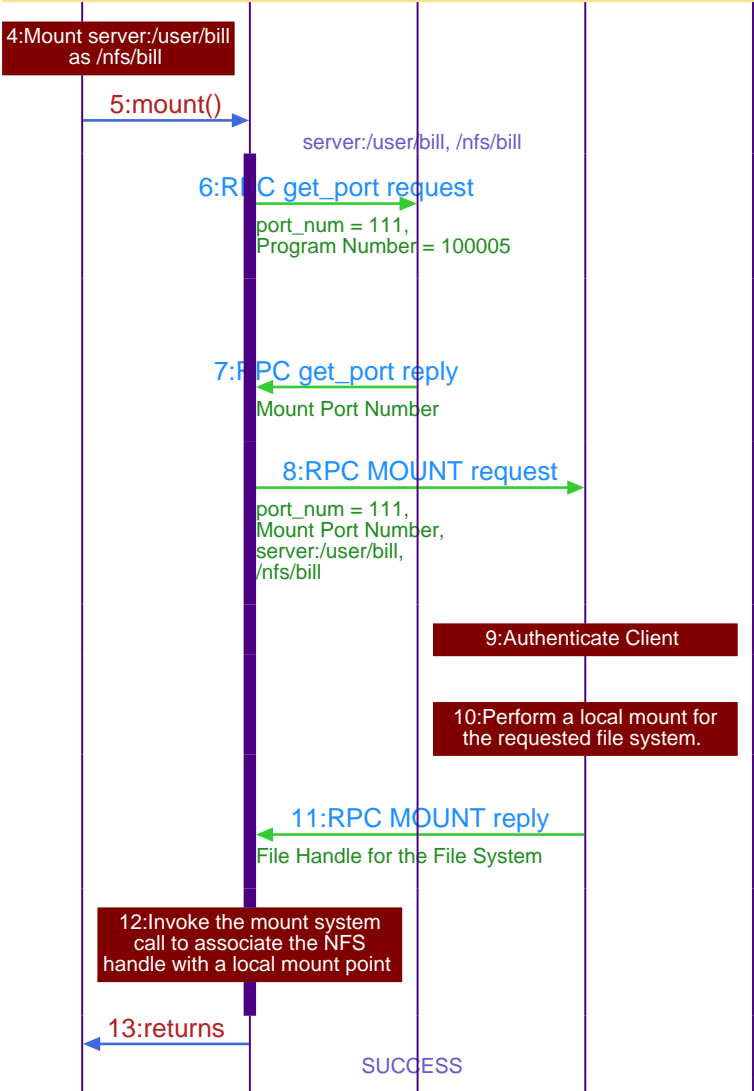
Server Startup

1:Open Port 111 for UDP and TCP
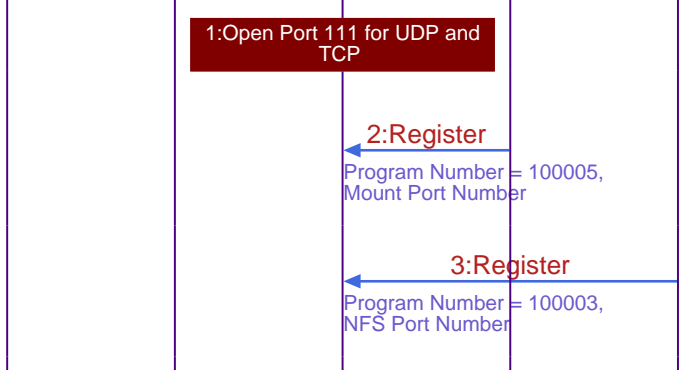
The port mapper starts listening on UDP and TCP port 111 for program number to port mapping requests from NFS clients.

2:Register
Program Number = 100005, Mount Port Number

Mountd, the daemon handling NFS mounts, registers its port number for receiving mount requests (Program number 1000005).

3:Register
Program Number = 100003, NFS Port Number

The NFS server registers its port number for NFS (Program number 1000003).

NFS Mount

4:Mount server:/user/bill as /nfs/bill

Application initiates mounting of a file system.

5:mount()
server:/user/bill, /nfs/bill

The mount API for the OS is invoked.

6:RPC get_port request
port_num = 111, Program Number = 100005

The mount applications port number on the server is determined from the port mapper. The request includes the program number for mount. Remote Procedure Call (RPC) is used as the underlying protocol for this access. All message interactions in NFS are performed as remote procedure calls.

7:RPC get_port reply
Mount Port Number

8:RPC MOUNT request
port_num = 111, Mount Port Number, server:/user/bill, /nfs/bill

Request the NFS server to mount the file system.

9:Authenticate Client

Mountd uses the client IP address to authenticate the client.
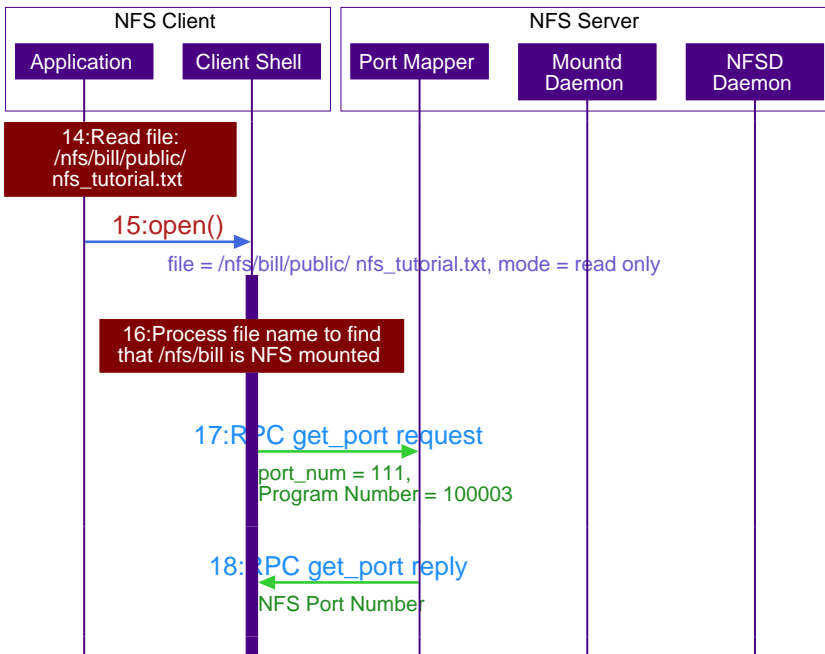
10:Perform a local mount for the requested file system.

The server performs a local mount. The results of this mount command will be passed back to the NFS client.

11:RPC MOUNT reply
File Handle for the File System

Communicate the mount results to the NFS client.

12:Invoke the mount system call to associate the NFS handle with a local mount point

The remote mount information is saved along with the local mount point.

13:returns
SUCCESS

Report a successful mount to the application.

Opening a File

NFS Client

| Application | Client Shell |

NFS Server

| Port Mapper | Mountd Daemon | NFSD Daemon |

14:Read file: /nfs/bill/public/ nfs_tutorial.txt

15:open()
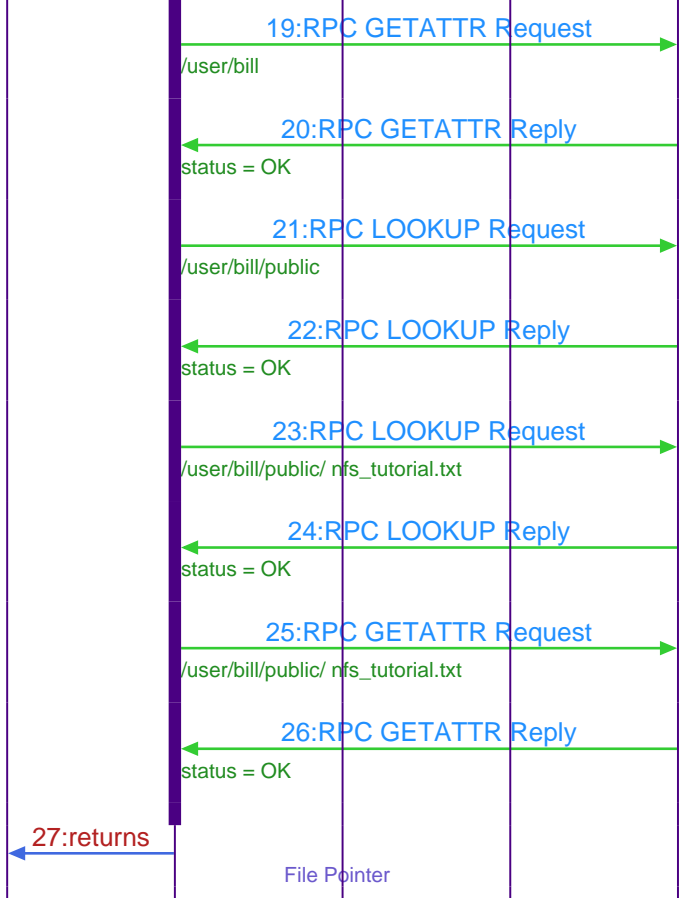
file = /nfs/bill/public/ nfs_tutorial.txt, mode = read only

Application uses the OS API to open a file. The application is not aware that the requested file is remotely located and needs to accessed via NFS.

16:Process file name to find that /nfs/bill is NFS mounted

The OS parses the file to find that the file /nfs/bill is a NFS mounted file system. This initiates the NFS file open processing.

17:RPC get_port request

port_num = 111, Program Number = 100003

Since the file is located on an NFS Server, the client initiates access by requesting the port number for the NFS service.

18:RPC get_port reply

NFS Port Number

The client hierarchically parses the path and obtains a file handle. NFS commands GETATTR and LOOKUP are used to perform these operations. GETATTR returns the attributes of a file, for example permissions, file size, file owner. LOOKUP looks up the file and returns a file handle to the file.

19:RPC GETATTR Request

/user/bill

Get the file attributes for the /user/bill directory.

20:RPC GETATTR Reply

status = OK

21:RPC LOOKUP Request

/user/bill/public

Check if the public directory is available.

22:RPC LOOKUP Reply

status = OK

23:RPC LOOKUP Request

/user/bill/public/ nfs_tutorial.txt

Check if the requested file exists.

24:RPC LOOKUP Reply

status = OK

25:RPC GETATTR Request

/user/bill/public/ nfs_tutorial.txt

Obtain the attributes about the requested file.

26:RPC GETATTR Reply

status = OK

27:returns

File Pointer

Reading File Contents

28:read()

The application initiates a read for the file.

This file is on a NFS mounted file system, so the read command is performed via NFS READ RPC calls. The READ calls specify the file name, the starting offset in each request. The NFS server reads the standard block size and returns to the client. Note that the NFS server itself is stateless and handles each request independently. The client maintains the application and performs multiple reads to complete the file read.

The following interaction shows how the client performs multiple reads to complete the file reading operation.

NFS Client

Application | Client Shell

NFS Server

Port Mapper | Mountd Daemon | NFSD Daemon

29:RPC READ Request

/user/bill/public/ nfs_tutorial.txt,
start_offset = 0

Read from the file start (offset 0)

30:RPC READ Reply

status = OK,
read_bytes = 1024

31:RPC READ Request

/user/bill/public/ nfs_tutorial.txt,
start_offset = 1024

Initiate the read from the offset 1024 (0 to 1023 byte offsets have already been read).

32:RPC READ Reply

status = OK,
read_bytes = 2048

33:RPC READ Request

/user/bill/public/ nfs_tutorial.txt,
start_offset = 2048

Read the remaining part of the file.

34:RPC READ Reply

status = OK,
read_bytes = 20

35:returns