

MIT OpenCourseWare
<http://ocw.mit.edu>

6.033 Computer System Engineering
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

Preparation for Recitation 12

Read textbook section 4.5, *Case Study: The Network File System (NFS)*.

NFS provides a filesystem interface to files stored on a remote machine. You are familiar with another network file system, AFS, which provides the same functionality to you when you login to an Athena workstation and access your home directory.

NFS raises a number of interesting naming questions. NFS goes to great lengths to make sure file handles are unique; it adds generation numbers for inodes and a file system ID to the filehandle in addition to the inode number. What is it about NFS's design that makes unique filehandles so important? What could go wrong if two files had the same filehandle?

The architects of NFS modified the UNIX kernel significantly to add the Virtual File System interface; modern variants of UNIX retain the VFS even when they don't use NFS. VFS provides a layer of indirection between applications and the underlying file system. Why was this indirection so important to NFS?

Some NFS operations are idempotent. Idempotent operations can be safely executed more than once. Compare the NFS write (see table 4-1) to the standard UNIX write system call. Why is the UNIX syscall not idempotent and why is an NFS write idempotent? Can the rename NFS operation be considered idempotent? What about if there are multiple clients?