File Sharing

- Sharing of files on multi-user systems is desirable
- Sharing may be done through a **protection** scheme
- On distributed systems, files may be shared across a network
- Network File System (NFS) is a common distributed file-sharing method
- If multi-user system
  - **User IDs** identify users, allowing permissions and protections to be per-user
    Group IDs allow users to be in groups, permitting group access rights
  - Owner of a file / directory
  - Group of a file / directory

File Sharing – Remote File Systems

- Uses networking to allow file system access between systems
  - Manually via programs like FTP
  - Automatically, seamlessly using **distributed file systems**
  - Semi automatically via the **world wide web**
- **Client-server** model allows clients to mount remote file systems from servers
  - Server can serve multiple clients
  - Client and user-on-client identification is insecure or complicated
  - **NFS** is standard UNIX client-server file sharing protocol
  - **CIFS** is standard Windows protocol
  - Standard operating system file calls are translated into remote calls
- Distributed Information Systems (**distributed naming services**) such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote computing

File Sharing – Failure Modes

- All file systems have failure modes
  - For example corruption of directory structures or other non-user data, called **metadata**
- Remote file systems add new failure modes, due to network failure, server failure
- Recovery from failure can involve **state information** about status of each remote request
- **Stateless** protocols such as NFS v3 include all information in each request, allowing easy recovery but less security
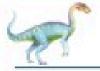
File Sharing – Consistency Semantics

- Specify how multiple users are to access a shared file simultaneously
    - Similar to Ch 5 process synchronization algorithms
        - Tend to be less complex due to disk I/O and network latency (for remote file systems
    - Andrew File System (AFS) implemented complex remote file sharing semantics
    - Unix file system (UFS) implements:
        - Writes to an open file visible immediately to other users of the same open file
        - Sharing file pointer to allow multiple users to read and write concurrently
    - AFS has session semantics
        - Writes only visible to sessions starting after the file is closed

Protection

- File owner/creator should be able to control:
    - what can be done
    - by whom
- Types of access
    - **Read**
    - **Write**
    - **Execute**
    - **Append**
    - **Delete**
    - **List**