Lecture Three

File System Hierarchy in Network Operating Systems

Dr. Tarfa Yaseen Hamed
Department of Networks
College of Computer Science and Mathematics
University of Mosul
2025

File System Hierarchy

- **Definition**: The file system hierarchy refers to the organization and structure of files and directories in an operating system. In a Network Operating System (NOS), this hierarchy is extended to manage files across multiple connected systems.
- **Purpose**: It provides a logical and standardized way to store, retrieve, and manage files, ensuring efficient access and security in a networked environment.

File System Hierarchy (cont.)

- **Key Concepts**:
  - **Files**: Units of data storage (e.g., documents, executables).
  - **Directories**: Containers for organizing files and subdirectories.
  - **Mount Points**: Locations in the hierarchy where external file systems (e.g., network shares) are attached.

File System Hierarchy in a NOS

- In a NOS, the file system hierarchy is designed to support:
- **Centralized Management**: Files are stored on servers and accessed by clients.
- **Distributed Access**: Users across the network can access shared resources.
- **Scalability**: The hierarchy must accommodate growing numbers of users and files.
- **Security**: Access controls are implemented to protect sensitive data.

Key Components of File System Hierarchy

- **Root Directory (/)**
  - The top-level directory in the hierarchy.
  - All other directories and files are organized under the root.
  - In a NOS, the root directory may represent a shared network resource.

Common Directories in a NOS

- **/home**: Contains user directories and personal files.
  - Example: /home/username for individual user storage.
- **/var**: Stores variable data like logs, databases, and emails.
- **/etc**: Holds configuration files for the system and applications.
- **/bin** and **/sbin**: Contain essential binary executables for system operations.
- **/usr**: Stores user-installed software and libraries.
- **/tmp**: Temporary files that are deleted upon reboot.
- **/mnt** and **/media**: Mount points for external file systems (e.g., network shares, USB drives).
- **/net**: A directory for accessing network file systems (e.g., NFS, SMB).

Network-Specific Directories

- **/net**: Used in some NOS to mount remote file systems.
- **/share**: A common directory for shared resources accessible by multiple users.
- **/backup**: Stores backups of critical data, often accessed over the network.

File System Types in a NOS

- **Local File Systems**: Used on individual machines (e.g., NTFS, ext4).
- **Network File Systems**: Enable file sharing across the network.
  - **NFS (Network File System)**: Common in Unix/Linux environments.
  - **SMB/CIFS (Server Message Block/Common Internet File System)**: Used in Windows environments.
  - **AFS (Andrew File System)**: Designed for scalability and security in distributed systems.

File System Hierarchy and Permissions

- **Access Control**: In a NOS, permissions are critical to ensure only authorized users can access files.
  - **Read (r)**: Allows viewing of file contents.
  - **Write (w)**: Allows modifying or deleting files.
  - **Execute (x)**: Allows running executable files or scripts.
- **Ownership**: Files and directories are owned by users and groups, with specific permissions assigned.
- **ACLs (Access Control Lists)**: Advanced permission systems for fine-grained control.

Design Principles

- **Simplicity**: The hierarchy should be easy to navigate and understand.
- **Consistency**: Standardized directory structures across systems.
- **Scalability**: Ability to handle growing amounts of data and users.
- **Security**: Implement robust access controls and encryption.
- **Redundancy**: Use RAID or backup systems to prevent data loss.

Challenges in File System Hierarchy for NOS

- **Data Synchronization**: Ensuring consistency across distributed systems.
- **Latency**: Network delays can affect file access speeds.
- **Security Risks**: Unauthorized access or data breaches.
- **Storage Management**: Efficiently managing limited storage resources.

Case Study: Linux File System Hierarchy in a NOS

- **Root (/)**: Central point for all directories.
- **/home**: User directories stored on a central server.
- **/var/log**: Centralized logging for network activities.
- **/mnt/nfs**: Mount point for NFS shares.
- **/etc/exports**: Configuration file for NFS exports.

Best Practices for Managing File System Hierarchy in a NOS

- **Regular Backups**: Ensure data is backed up frequently.
- **Permission Audits**: Regularly review and update access controls.
- **Monitoring**: Use tools to monitor file system usage and performance.
- **Documentation**: Maintain clear documentation of the hierarchy and access policies.

Case Study - Windows Server 2019 File System Hierarchy

- **Introduction to Windows Server 2019**
  - **Windows Server 2019** is a widely used NOS (Network Operating System) designed for enterprise environments.
  - It provides robust file system management, including support for shared directories, access controls, and distributed file systems.
  - The file system hierarchy in Windows Server 2019 is based on the **NTFS (New Technology File System)** and supports **SMB/CIFS** for network file sharing.

Key Directories in Windows Server 2019

- **System Drive (C:\)**
  - The root directory of the system drive, typically C:\.
  - Contains critical system files and directories.
- **Windows Directory (C:\Windows)**
  - Stores the operating system files, including system libraries, executables, and configuration files.
  - Example: C:\Windows\System32 contains essential system binaries.

Key Directories in Windows Server 2019 (cont.)

- **Program Files** (C:\Program Files and C:\Program Files (x86))
  - Contains installed applications and software.
  - C:\Program Files is for 64-bit applications.
  - C:\Program Files (x86) is for 32-bit applications.

Key Directories in Windows Server 2019 (cont.)

- **Users Directory (C:\Users)**
  - Contains user profiles and personal files.
  - Example: C:\Users\Username for individual user directories.

Key Directories in Windows Server 2019 (cont.)

- **Shared Directories (C:\Shares)**
  - A common location for shared folders accessible over the network.
  - Example: C:\Shares\Finance for financial documents.

Key Directories in Windows Server 2019 (cont.)

- **System Volume Information (C:\System Volume Information)**
  - Stores system restore points and other system-related data.
  - Accessible only by the system and administrators.

Key Directories in Windows Server 2019 (cont.)

- **Network Shares (\\ServerName\ShareName)**
  - Network shares are accessed using the \\ServerName\ShareName syntax.
  - Example: \\FileServer\Public for a shared public folder.

File System Features in Windows Server 2019

- **NTFS (New Technology File System)**
  - Supports advanced features like:
    - **File Permissions**: Granular access control using ACLs (Access Control Lists).
    - **Encryption**: Encrypting File System (EFS) for securing sensitive data.
    - **Quotas**: Disk quotas to limit user storage usage.
    - **Compression**: File and folder compression to save disk space.

File System Features in Windows Server 2019

- **SMB/CIFS (Server Message Block/Common Internet File System)**
- Enables file sharing across the network.
- Supports features like:
  - **SMB Direct**: For high-performance file transfers over RDMA (Remote Direct Memory Access).
  - **SMB Multichannel**: For increased throughput and fault tolerance.

SMB/CIFS (Server Message Block/Common Internet File System)

- **SMB (Server Message Block)**: A network file-sharing protocol that allows systems to access files, printers, and other resources on a network.
- **CIFS (Common Internet File System)**: A dialect of SMB, originally developed by Microsoft, and widely used in Windows environments.
- **Purpose**: Enables file and resource sharing between systems in a networked environment, regardless of the operating system.

How SMB/CIFS Works

- **Client-Server Model**:
  - The **client** requests access to files or resources.
  - The **server** hosts the shared resources and responds to client requests.
- **Communication**:
  - SMB operates over TCP/IP (port 445) or NetBIOS (ports 137-139).
  - Clients and servers negotiate the SMB dialect (e.g., SMB 1.0, SMB 2.0, SMB 3.0) during the connection setup.
- **File Access**:
  - Clients can open, read, write, and delete files on the server.
  - Supports file locking to prevent conflicts during simultaneous access.

Challenges in Windows Server 2019 File System Management

- **Security**: Ensuring proper permissions and encryption to prevent unauthorized access.
- **Scalability**: Managing large numbers of files and users.
- **Backup and Recovery**: Implementing regular backups to prevent data loss.
- **Compatibility**: Ensuring compatibility with older SMB versions for legacy systems.