

1.

In a modern web server environment, the file system plays a critical role in managing and organizing data. The server hosts multiple websites, each stored in its own directory with subdirectories for HTML files, CSS stylesheets, images, and scripts. When a client requests a webpage, the server accesses the file system to retrieve and deliver the required resources. For example, a request for an image file triggers the server to look inside the website's 'images' folder and fetch the correct file path. The file system ensures that permissions are properly managed so only authorized users or processes can modify files. Logs and configuration files are also stored in well-structured locations to maintain reliability and order. Without an efficient file system, data could become disorganized, making the server unstable or insecure. This structure supports performance, security, and smooth operations.

2.

Journaling is a technique used by file systems to enhance reliability and protect against data corruption. It works by keeping a log, or journal, of changes before they are permanently written to the main file system. If the system crashes during a write operation, the journal helps restore consistency by replaying or discarding incomplete transactions. This prevents issues such as broken files or corrupted metadata, which can occur during sudden power loss or hardware failures. Journaling is especially important in environments like servers or databases, where data integrity is critical. By reducing the need for lengthy file system checks after crashes, it saves time and improves recovery speed. File systems like ext3, ext4, and NTFS use journaling for better reliability and stability. Overall, journaling ensures a more robust and fault-tolerant file system operation.

3.

For practice, I designed a simple directory structure for a web application project. At the root, there is a main folder called 'WebApp.' Inside it, subfolders are created: 'HTML' for web pages, 'CSS' for styling files, 'JS' for scripts, 'Images' for media, and 'Logs' for server activity. Within 'HTML,' each page like 'index.html' and 'about.html' is neatly stored. The 'CSS' folder contains

'style.css,' while 'JS' has 'app.js.' This structure ensures files are easy to find and maintain. Reflecting on this exercise, it reinforced how critical organization is in development. A logical directory structure speeds up troubleshooting, supports teamwork, and reduces clutter. It also mirrors professional project standards, which is important when collaborating with others. Overall, it was a simple yet effective practice to appreciate how structured storage makes a significant difference in real projects.

4.

Using Canva, I created a simple file system diagram to represent the directory structure of a web server. At the top, the root directory 'WebApp' branches into subfolders: 'HTML,' 'CSS,' 'JS,' 'Images,' and 'Logs.' The 'HTML' folder includes 'index.html' and 'about.html,' while the 'CSS' folder contains 'style.css.' Similarly, 'JS' holds 'app.js,' and 'Images' includes placeholders for media files like 'logo.png.' The 'Logs' folder is shown as storing server log files. This diagram visually demonstrates how files are organized and accessed by the system. Explaining it is easy: the hierarchy shows parent-child relationships between folders and files, making it clear where everything belongs. Visualizing the file system in this way also helps understand its structure better than just text. Such diagrams are practical for planning, teaching, or collaborative work.

