Forensic Analysis of the LittleFS Filesystem

Where is LittleFS Used?

LittleFS is a filesystem designed with embedded systems and microcontrollers in mind, specifically those with limited storage and RAM/CPU resources. It's often found in IoT devices, wearables, sensors, and consumer electronics. Platforms like STM32, ESP32 and nRF52 often use LittleFS in their firmware, especially when they need persistent storage with flash memory. It is popular because it works very well with NOR flash, handles power failures gracefully and requires minimal system resources.

Main Characteristics

LittleFS was built to handle three common problems in embedded environments:

1. **Power-loss resilience**: Data operations needs to be atomic. It uses a copy-on-write strategy, which means that it never overwrites existing data until the new data is safely stored.
2. **Wear leveling**: Flash memory has a limited number of write cycles. LittleFS spreads these writes across the memory to extend the device's lifespan.
3. **Low memory usage**: It can run with just a few kilobytes of RAM and code space, making it ideal for smaller microcontrollers.

It also supports:

- Static block sizes (typically 512B to 4KB)
- Nested directories
- Configurable block and cache settings

The design is also log-structured, which means that updates are written sequentially to new locations instead of modifying blocks in-place. This simplifies recovery and ensures integrity.

What Happens When a File is Deleted?

When a file is deleted in LittleFS the operation affects both the directory's structure and the metadata of the underlying block:

1. The directory's entry pointing to the file is removed or marked as invalid.
2. The file data itself is not erased immediately. Instead it becomes unreferenced in the filesystem.
3. Over time, the garbage collection system will reclaim and reuse those blocks. Until then, the raw contents often still exist on the flash.
4. Metadata structures (such as file headers and block pointers) may also remain in the storage image until overwritten.

This behavior means that, from a forensic standpoint, deleted files may still be recoverable from the raw storage — at least temporarily. Unlike secure filesystems that zero out deleted content, LittleFS prioritizes performance and wear reduction over secure deletion.