

Chapter 41: Locality and The Fast File System (FFS)

1. The Problem with the Old Unix File System

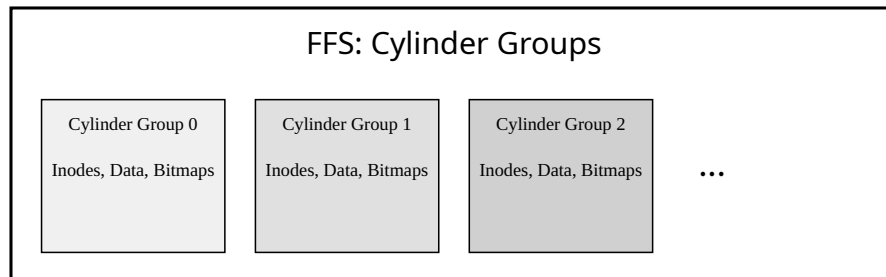
- The original Unix file system was simple but suffered from poor performance.
- **Problem:** It was not “disk-aware.” It treated the disk as a random-access memory, leading to excessive seek times.
- **Consequences:**
 - Files and their metadata were scattered across the disk.
 - Free space was not managed intelligently, leading to fragmentation.

2. The Solution: The Fast File System (FFS)

- FFS was designed to be “disk-aware” and improve performance by optimizing for disk characteristics.
- **Key Idea:** Keep related data together to minimize seek times.

3. Organizing Structure: Cylinder Groups

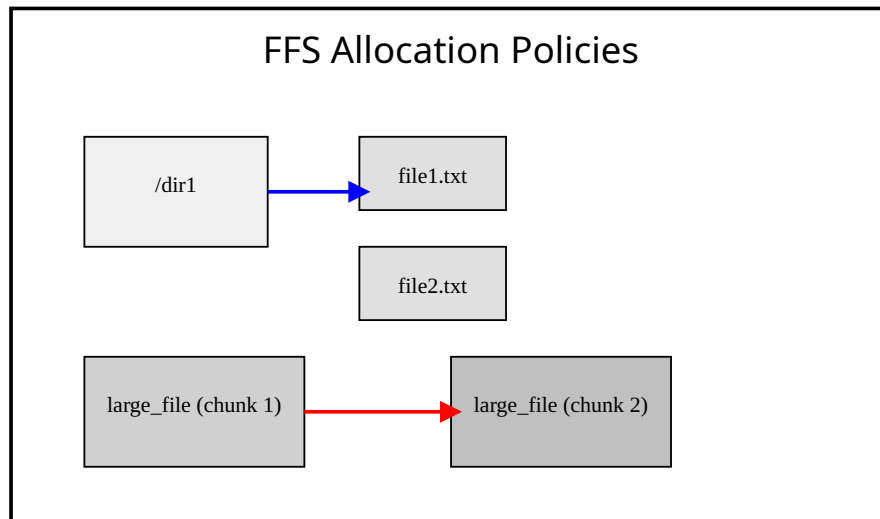
- FFS divides the disk into **cylinder groups** (or block groups).
- Each cylinder group is a self-contained unit with its own:
 - Superblock (a copy)
 - Inodes
 - Data blocks
 - Bitmaps for tracking free inodes and data blocks
- This allows for localization of data and metadata.



4. Allocation Policies

- FFS uses intelligent allocation policies to improve locality:
 - **Directories:** New directories are placed in the cylinder group with the lowest number of directories and the highest number of free inodes.
 - **Files:** Files are placed in the same cylinder group as their parent directory.

- **Large Files:** Large files are split into chunks, and each chunk is placed in a different cylinder group. This prevents a single large file from dominating a cylinder group.



Allocation Code Example

```

// Simplified FFS allocation logic
if (is_directory(new_file)) {
    // Find group with fewest directories and most free inodes
    group = find_best_directory_group();
} else {
    // Place file in the same group as its parent directory
    group = parent_directory->group;
}

if (is_large(file)) {
    // Spread chunks across different groups
    allocate_chunks_across_groups(file);
}
  
```

5. Other FFS Innovations

- **Sub-blocks:** To reduce internal fragmentation, FFS allowed for the division of blocks into smaller sub-blocks (or fragments).
- **Long File Names:** FFS increased the maximum file name length from 14 to 255 characters.
- **Symbolic Links:** Introduced the concept of symbolic links.

- **Atomic `rename()`:** Provided an atomic `rename()` operation, which is crucial for file system consistency.

6. FFS Summary

- FFS was a major step forward in file system design.
- By being disk-aware and using intelligent allocation policies, it significantly improved performance over the original Unix file system.
- Many of the concepts introduced in FFS are still used in modern file systems today.