Log-Structured File System (LFS)

COMS W4118

References: Operating Systems Concepts (9e), Linux Kernel Development, previous W4118s

Log-Structured File System

- Motivation
  - Faster CPUs: I/O becomes more and more of a bottleneck
  - More memory: file cache is effective for reads
  - Implication: writes compose most of disk traffic
- Problems with previous FS
  - Perform many small writes
    - Good performance on large, sequential writes, but many writes are still small, random
  - Synchronous operations to avoid data loss
    - i.e., journaling
  - Depends upon knowledge of disk geometry
    - i.e., cylinder groups

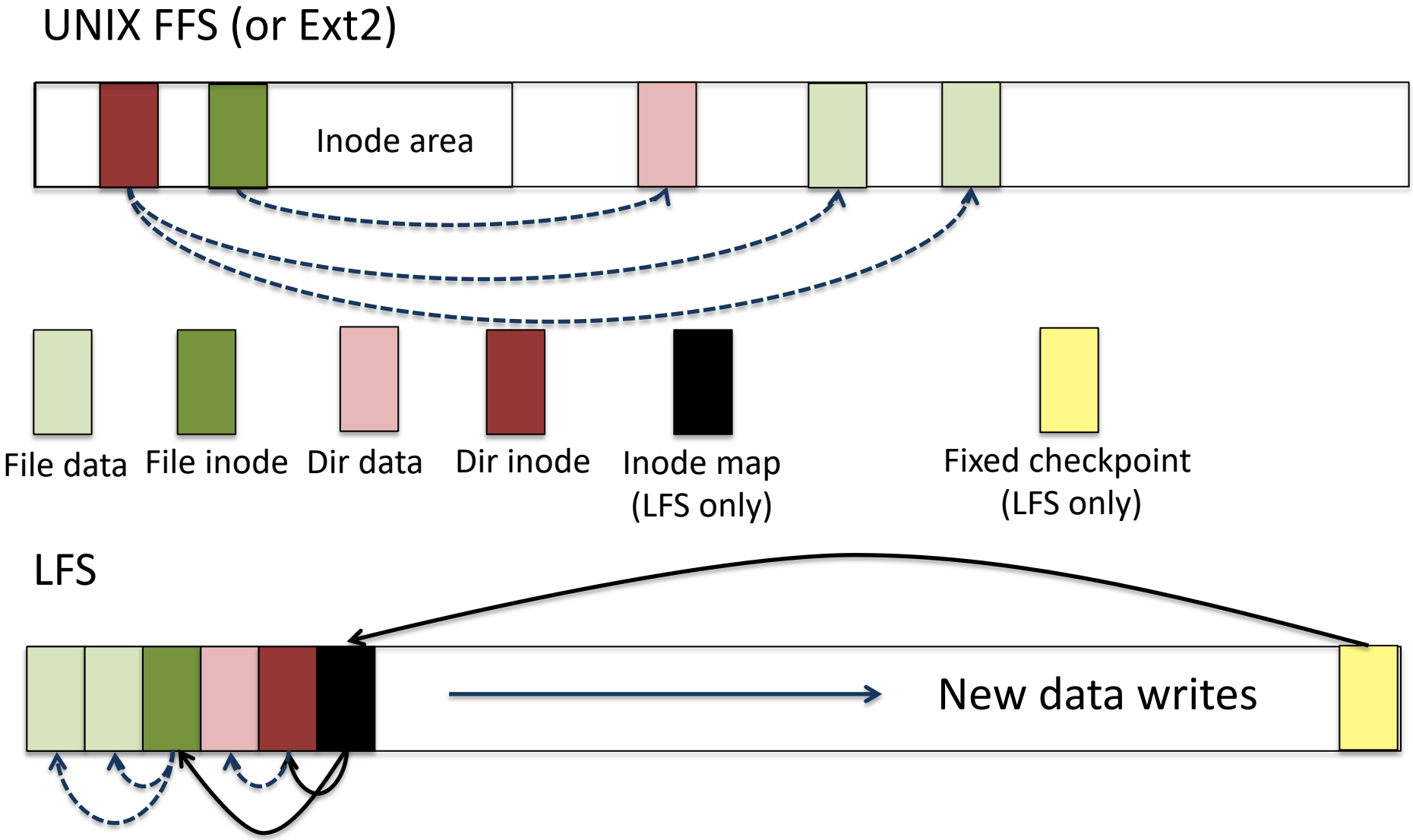
LFS Big Ideas

- Insight: treat disk like a tape-drive
  - Disk performs best for sequential access
  - Essentially, extreme journaling
- Write data to disk in a sequential log
  - Delay all write operations
    - Prefer writing one huge “segment” over a bunch of small writes
  - Write metadata and data for all files intermixed in one operation
    - How to find data/metadata if not centralized?
  - Do not overwrite old data on disk
    - When do you clean up old data?

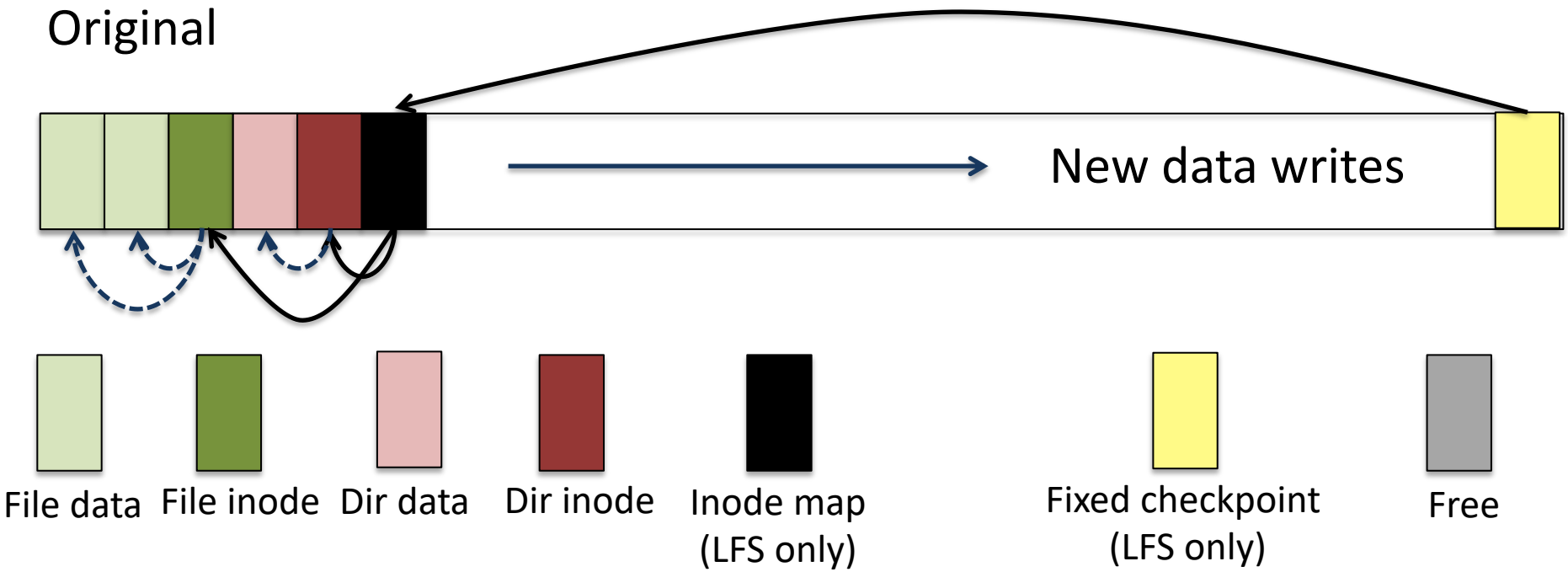
LFS Data Structures

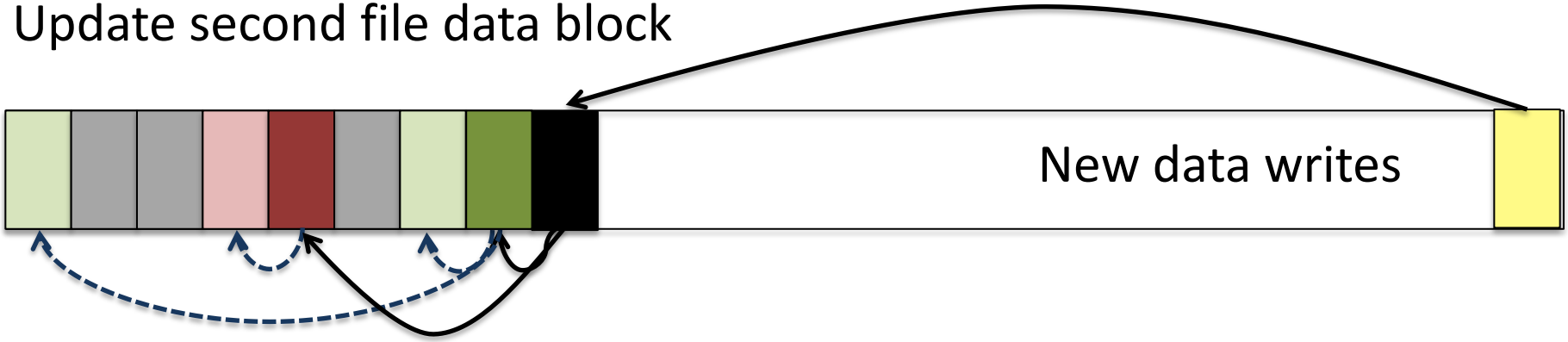
- Same basic structures as Unix
  - Directories, inodes, indirect blocks, data blocks
  - Reading data block implies finding the file's inode
    - Unix: inodes kept in array
    - LFS: inodes **move around** on disk
- Solution: **inode map** indicates where each inode is stored
  - Small enough to keep in memory
  - inode map written to log with everything else
  - Periodically written to known checkpoint location on disk for crash recovery

Efficient Reads: Indexing the Log



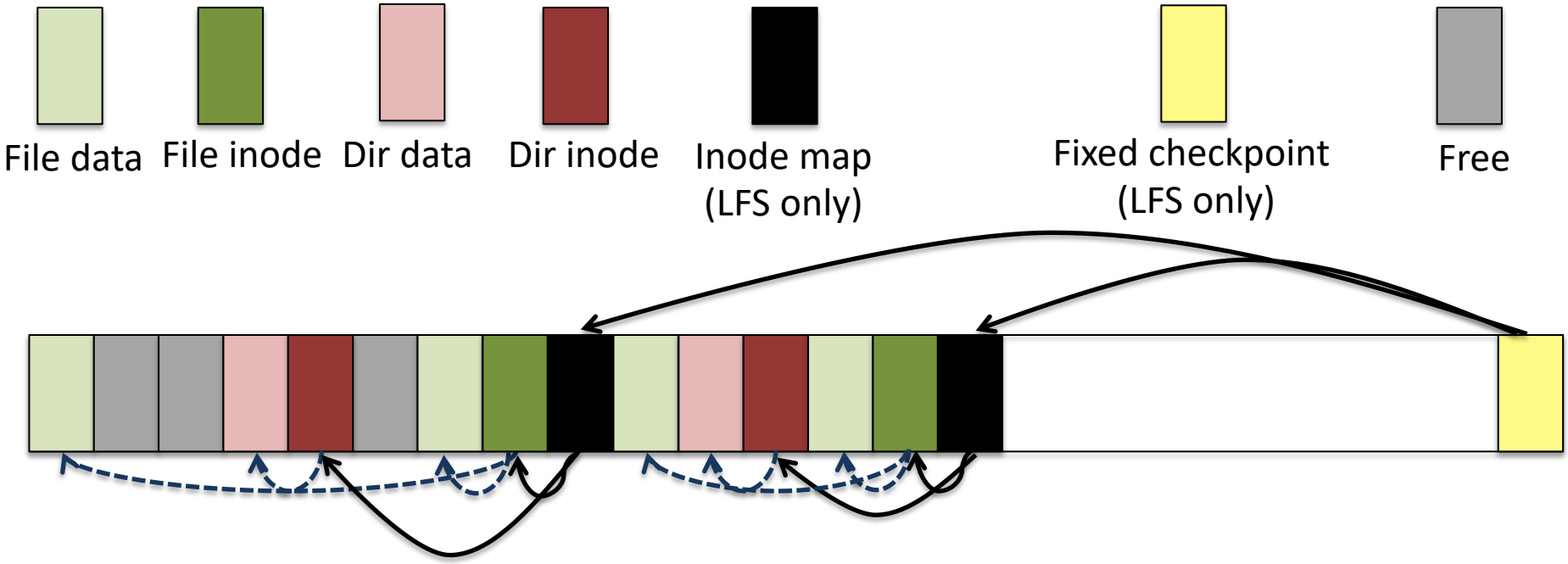
Writes: Copy on Write





Disk Cleaning

- When disk runs low on free space
  - Run a disk cleaning process
  - Compacts live information to contiguous blocks of disk



In reality, too expensive to clean contiguously.
FS is split into moderately large segments (e.g., 1MB or more).

Disk Cleaning

- When disk runs low on free space
  - Run a disk cleaning process
  - Compacts live information to contiguous blocks of disk
- Problem: long-lived data repeatedly copied over time
  - Solution: Group older files into same segment
  - Old segments won't have many changes. Skip.
- LFS: neat idea, influential
  - Paper on LFS one of the most widely cited OS paper
  - Many real file systems based on the idea
    - Relevant for SSD-conscious designs