Fast File System (FFS)

Jin-Soo Kim
(*jinsoo.kim@snu.ac.kr*)

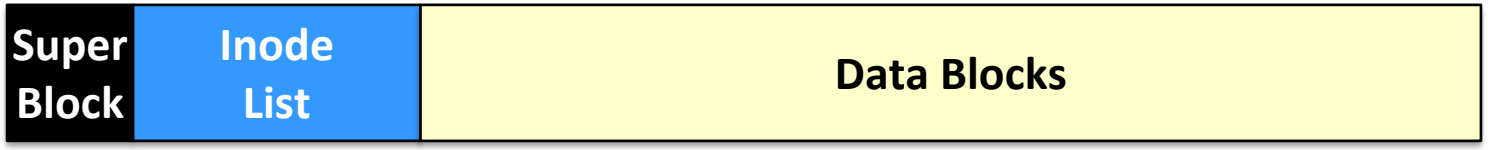
Systems Software &
Architecture Lab.

Seoul National University

Spring 2024

The Original Unix FS

- First Unix file system developed by Ken Thompson

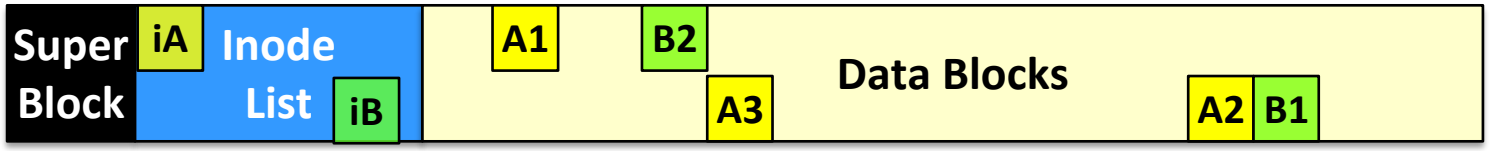


- Super block
  - Basic information of the file system
  - Head of freelists of inodes and data blocks
- Inode list
  - Referenced by index into the inode list
  - All inodes are the same size
- Data blocks
  - A data block belongs to only one file

FFS

- The original Unix file system (70's) was very simple and straightforwardly implemented
  - But, achieved only 2% of the maximum disk bandwidth
- BSD Unix folks redesigned file system called FFS
  - McKusick, Joy, Leffler, and Fabry (80's)
  - Keep the same interface, but change the internal implementation
- The basic idea is disk-awareness
  - Place related things on nearby cylinders to reduce seeks
  - Improved disk utilization, decreased response time

Unix FS: Problems



- Files are fragmented as the file system "_____"
  - Blocks are allocated randomly over the disk
- Inodes are allocated far from blocks
  - Traversing pathnames or manipulating files and directories requires long seeks between inodes and data blocks
- Files in a directory are typically not allocated in consecutive inode slots
- The small block size: 512 bytes

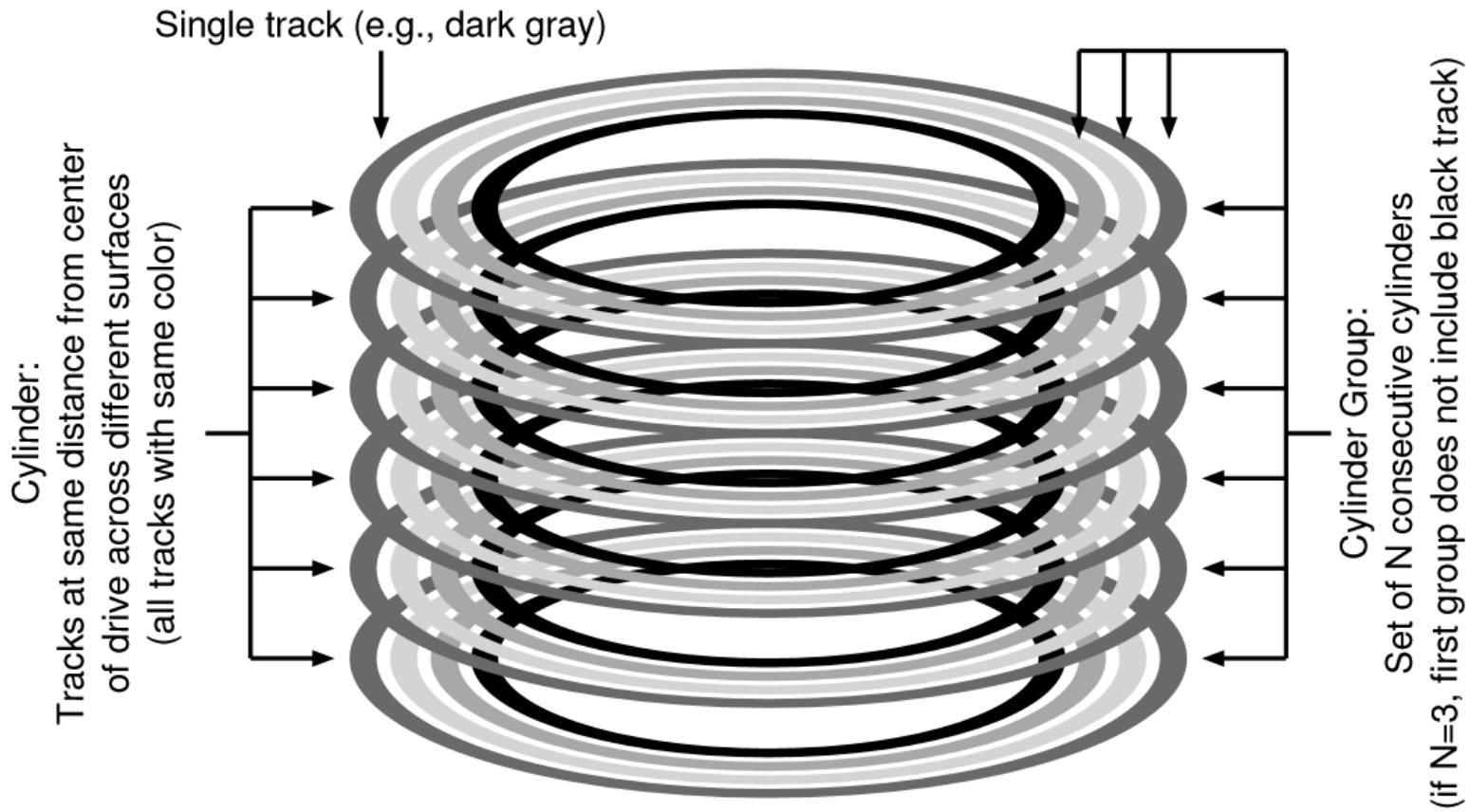
Bitmaps

- Use bitmaps instead of free lists

- Each bit represents whether the corresponding inode (or data block) is free or in use
- Provides better speed, with more global view
- Faster to find contiguous free blocks
- Helps to reduce file fragmentation

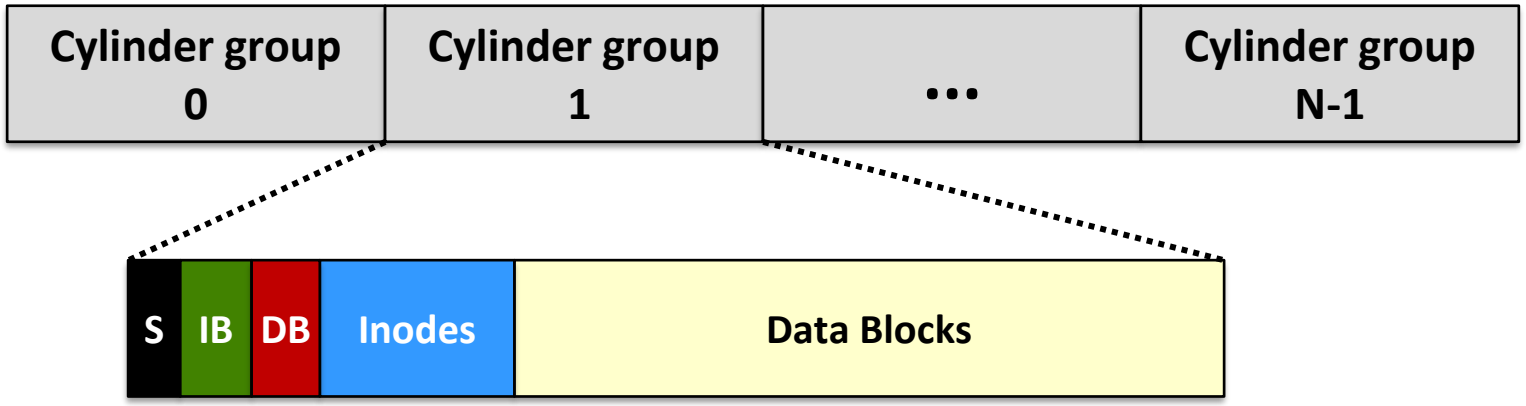
Cylinder Groups

- Divides the disk into a number of cylinder groups



On-Disk Layout

- Put all the structures within each cylinder group
  - Modern drives do not export disk geometry information
  - Modern file systems organize the drive into "block groups" (e.g., Linux Ext2/3/4)
  - Block size is increased to 4KB to improve throughput
  - Super block (S) is replicated for reliability reasons

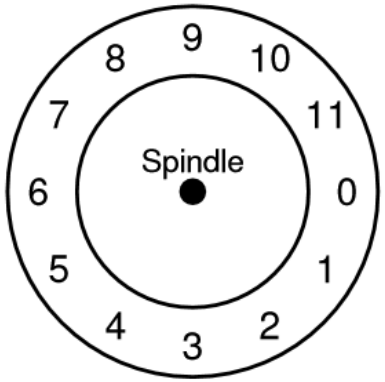


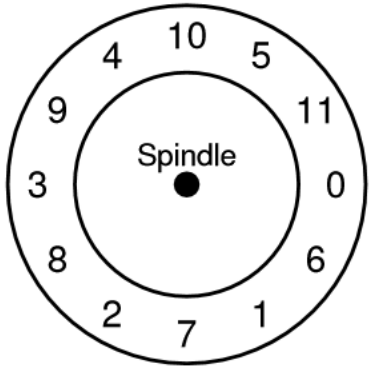
Allocation Policies

- Keep related stuff together
- Balance directories across groups
  - Allocate directory blocks and its inode in the cylinder group with a low number of allocated directories and a high number of free inodes
- Files in a directory are often accessed together
  - Place all files that are in the same directory in the cylinder group of the directory
  - Allocate data blocks of a file in the same group as its inode
  - Data blocks of a large file are partitioned into chunks and distributed over multiple cylinder groups

Other Features

- Fragments to reduce internal fragmentation
  - Each block can be broken optionally into 2, 4, or 8 fragments
  - The block map manages the space at the fragment level
- File system parameterization
  - Make the next block come into position under the disk head by skipping some blocks
- Free space reserve
- Long file names
- Atomic rename
- Symbolic links





Summary

- First disk-aware file system
  - Cylinder groups
  - Bitmaps
  - Replicated superblocks
  - Large blocks
  - Smart allocation policies
- FFS achieves 14% ~ 47% of the disk bandwidth
  - Original Unix FS: 3% ~ 5% of the disk bandwidth
  - The throughput deteriorates to about half when the file system is full
- FFS inspired modern file systems including Ext2/3/4