



# Operating Systems W13L2 - File Systems II (ctd)

▼ Class	Operating Systems
🕒 Created	@Nov 25, 2020 2:15 PM
🔗 Materials	
☑ Reviewed	<input type="checkbox"/>
▼ Type	Lecture

## Special File Systems (ctd)

Starting this stuff last lecture, reviewing it today and adding on to log structured system.

### Log Structured

- For log-structured file systems, disk caches can satisfy most requirements
- Writes to disk are traditionally slow
- When cleaning logs, the file becomes *fragmented* and thus wasting space →  
Compressing logs happens fairly frequently, but there is a cost to that
- Writing is fast here

### Journaling File System

- Logs operations prior to actually doing them, so if a crash occurs, the last thing is what was done
- Slower writing, but advantages in terms of storing crucial information

- Different systems have different use cases in order to fulfill different criteria

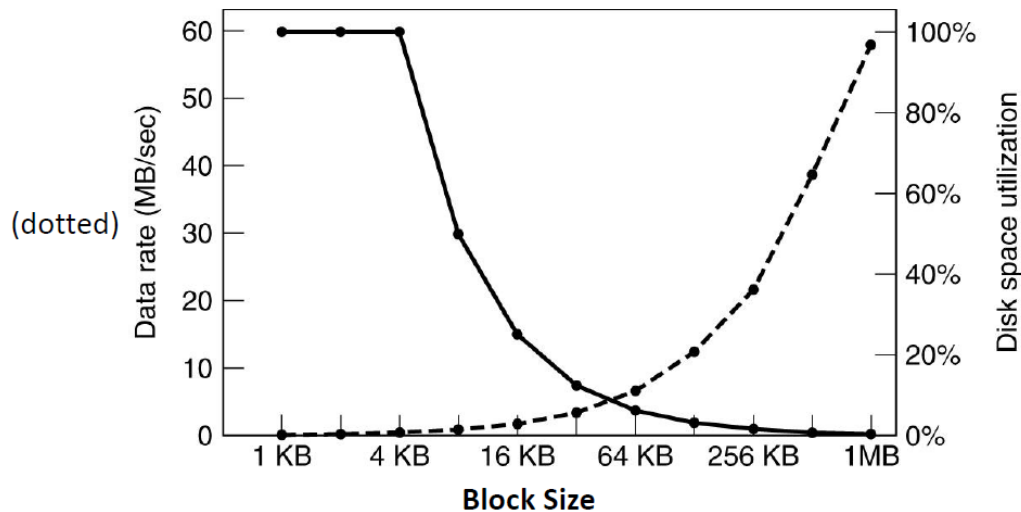
## Disk Space Management

- All file systems broken up into fix sized blocks
  - If block size is too large, then we waste space
  - If block size is too small, then we waste time

### ▼ Block Size vs. Data Rate / Disk Space Utilization

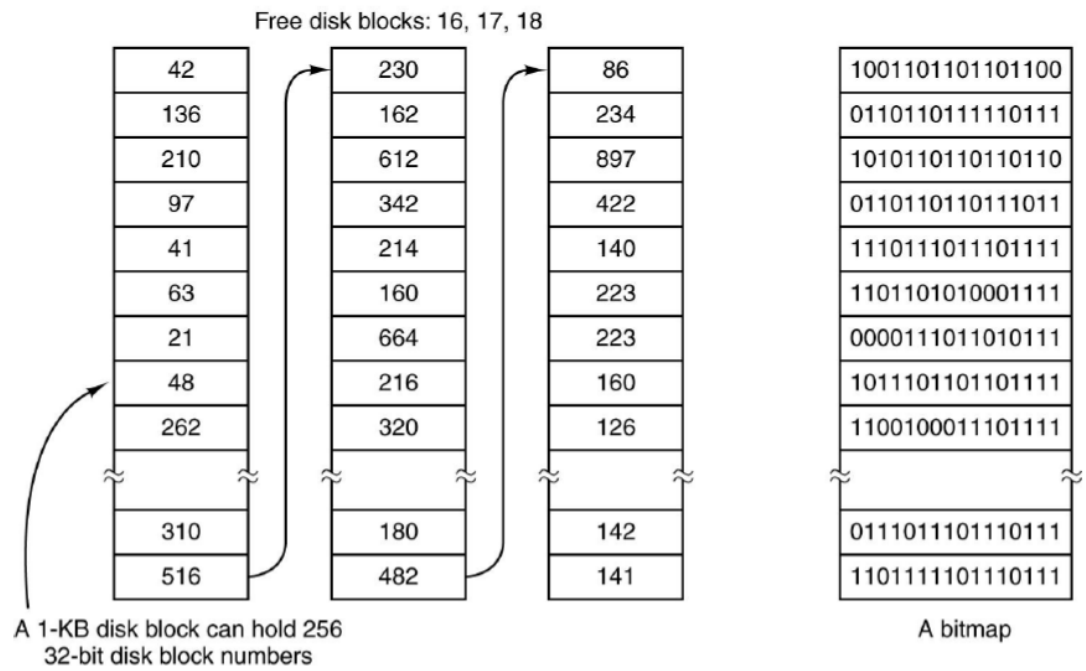
Access time for a block is completely dominated by the seek time and rotational delay (assume traditional hard-disk).

So ... The more data are fetched the better.



- Two methods to keep track of free blocks
  1. LinkedList of disk blocks with each block holding as many free (disk) block numbers as possible
  2. Using a bitmap — One bit corresponds to every block

### ▼ Diagram of Both Methods

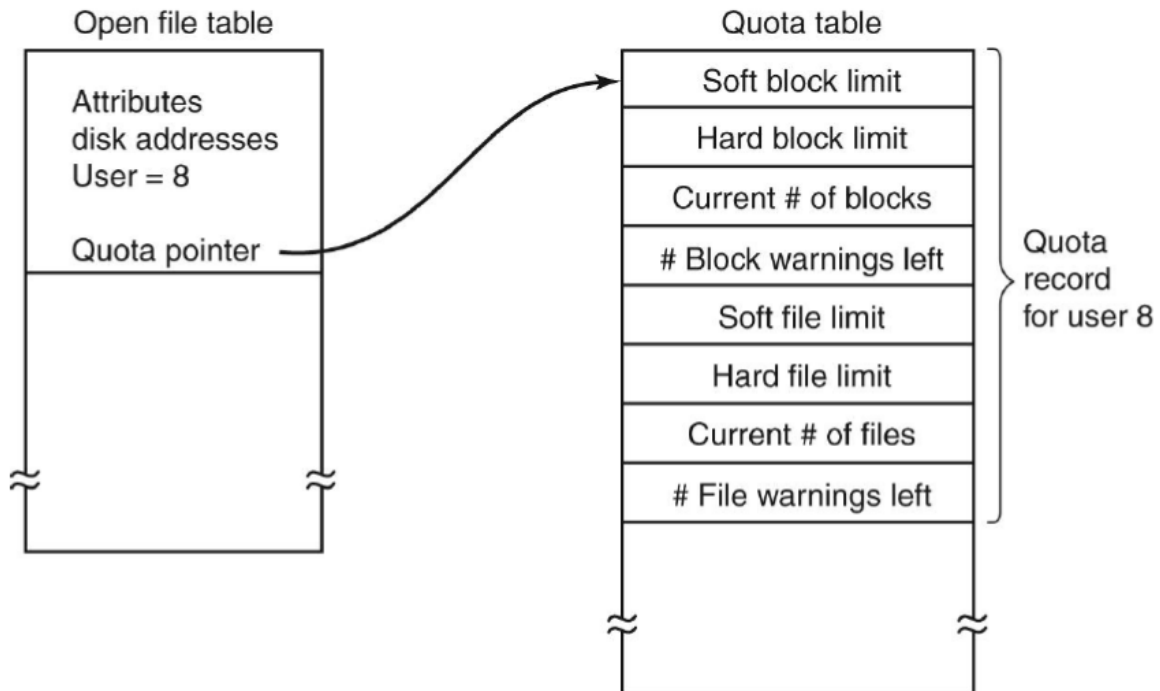


**Free blocks are holding the free list.**

- Also specify all instructions in steps and use terms that can be measured when you justify your design decisions

## Disk Quotas

### ▼ Diagram



- Each user is given space in memory table

## Reliability

Consists of backups and consistency

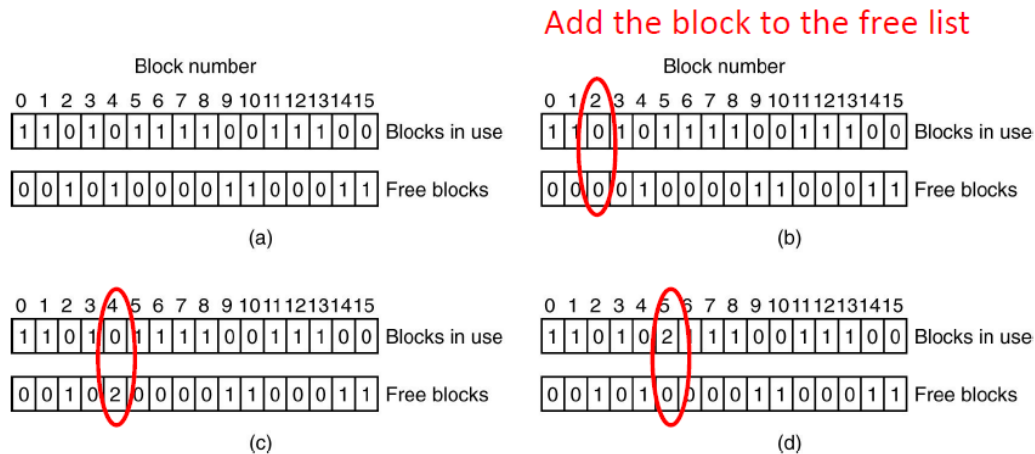
### Backups

- Usually desirable to backup only specific directories rather than entire system → *Incremental dump*
- With immense data, compression may be necessary (but also will be more expensive)
- **Incremental Dump:** Backup only the files that have been modified from last full backup
- ▼ Physical Dump

- Starts at block 0 of the disk
- Writes all the disk blocks onto the output tape (or any other type of storage) in order.
- Stops when it has copied the last one.
- + Simplicity and great speed
- Inability to skip selected directories and restore individual files.

### Consistency

- Two levels of consistency
  - ▼ Block level
    - Build two tables, each one contains a counter for each block, initially 0
    - Table 1: How many times each block is present in a file
    - Table 2: How many times a block is present in the free list
    - A consistent file system: each block has 1 either in the first or second table



Add the block to the free list

Rebuild the free list

Allocate a free block, make a copy of that block and give it to the other file.

#### ▼ File level

- Table of counters; a counter per file
- Counts the number of that file's usage count.
- Compares these numbers in the table with the counts in the i-node of the file itself.
- Both counts must agree.

- Two inconsistencies:
  - count of i-node > count in table
  - count of i-node < count in table
- Fix: set the count in i-node to the correct value

## Performance

- Caching — Block cache
- Block Read Ahead — Put blocks in cache before needed
- Reducing Disk Arm Motion — Putting blocks that might be used together close together on a cylinder
- Degrammentation — Putting blocks next to each other contiguously

## Conclusions

- Files and file system are major parts of an OS.
- There are different ways of organizing files, directories, and their attributes.
- Files and File system are the OS way of abstracting storage.