What is a File System?

15-213 / 15-503: Introduction to Computer Systems
Other Lecture, July 16, 2025

Instructors:

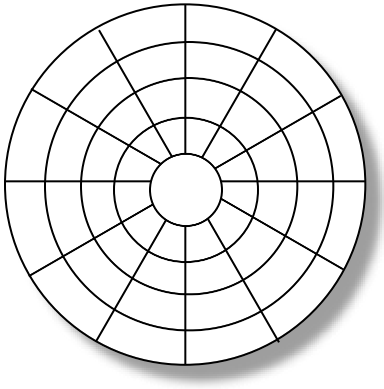
Brian Railing

Today

- What is a File System?
- Managing a file system
- Common operations

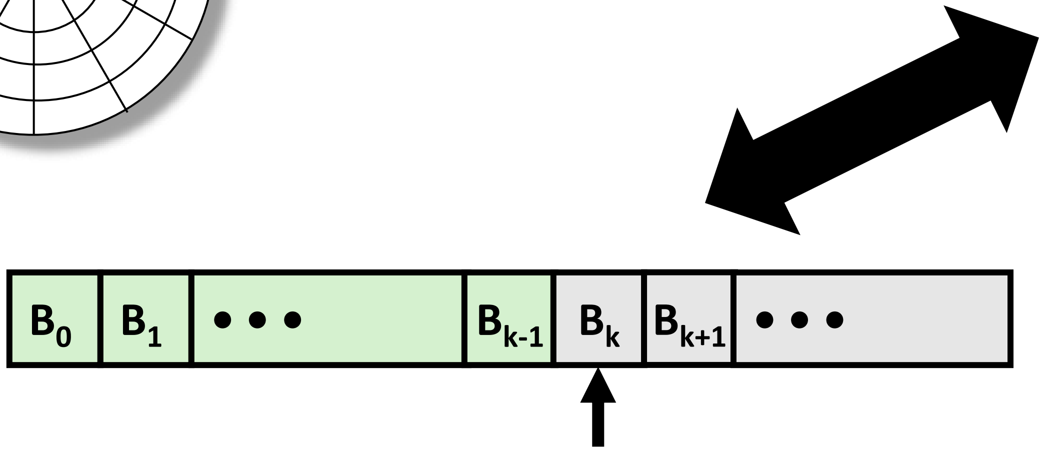
File System

- Manages disk blocks to provide a file abstraction



Surface* organized into tracks

Tracks divided into sectors



Current file position = k

*Durable storage has many architectures, but ultimately they expose "blocks"

Making a File System

- **File systems start by formatting raw disk blocks**
  - Designate one (or more) blocks as “super”
  - Record the rest of the blocks as free

Managing a File System

- **“Super” block is the master block with information**
  - Type information
  - Size
  - Root directory
  - Free blocks

- **SFS has a flat directory structure, so the root directory is part of the superblock**

Finding a File

- **A directory is a special file**
  - Maps strings to files
  - Those files could also be directories

Index in directory

Max files in directory

Allocated?

```
for (fileEntry = 0; (unsigned long)fileEntry < FILE_COUNT_LIMIT;
         fileEntry++) {
        if (superBlock->files[fileEntry].first_block != 0 &&
            strcmp(superBlock->files[fileEntry].name, fileName) == 0) {
            return addOpenFileEntry(fileEntry);
```

Check name

Opening a File

- **Find the file**
- **Create the three table entries**
  - Find an available file descriptor
  - Allocate an open file table entry
    - Pos, permissions, etc
  - Load file info into memory
    - *SFS is always in-memory, so this is implicit

Reading a File

- **The file system will map file pos to disk blocks**
- **Lots of ways to map**
  - Contiguous
  - Linked / FAT ← **SFS**
  - Indexed

Writing a File

- **Like reading, but the file could grow**
  - SFS preallocates space
  - Interesting synchronization

Deleting a File

- **Like free(), but …**
  - Can open files be deleted?

- **Two steps:**
  - Removing the mapping
  - Putting the blocks into the free list

SFS Specific Notes

- **“Shark” File System**
  - Uses mmap to bring the entire “disk” file into memory
  - Treats the disk as an array of 512-byte blocks
  - Block 0 is the superblock, other references to 0 are NULLs
  - Flat directory structure

Further Notes

- https://tcpp.cs.gsu.edu/curriculum/sites/default/files/Edupar115.pdf

- **Scope of assignment:**
  - Average of 200 lines of additional code
  - 13 hours to complete (9 days of assigned work)

- **The tricky part is identifying critical sections**
  - Critical sections are defined by the shared variable / resource
  - That can be two (or more) threads calling the same or different functions

Part of sfs_open

```
    sfs_filesystem_t *superBlock = accessSuperBlock();
    int fileEntry;
    int emptyEntry = -1;
    for (fileEntry = 0; (unsigned long)fileEntry <
FILE_COUNT_LIMIT;
         fileEntry++)
    {
        if (superBlock->files[fileEntry].first_block != 0 &&
            strcmp(superBlock->files[fileEntry].name, fileName)
== 0)
        {
            return addOpenFileEntry(fileEntry);
        }
        else if (emptyEntry == -1 &&
                 superBlock->files[fileEntry].first_block == 0)
        {
            emptyEntry = fileEntry;
        }
    }
```