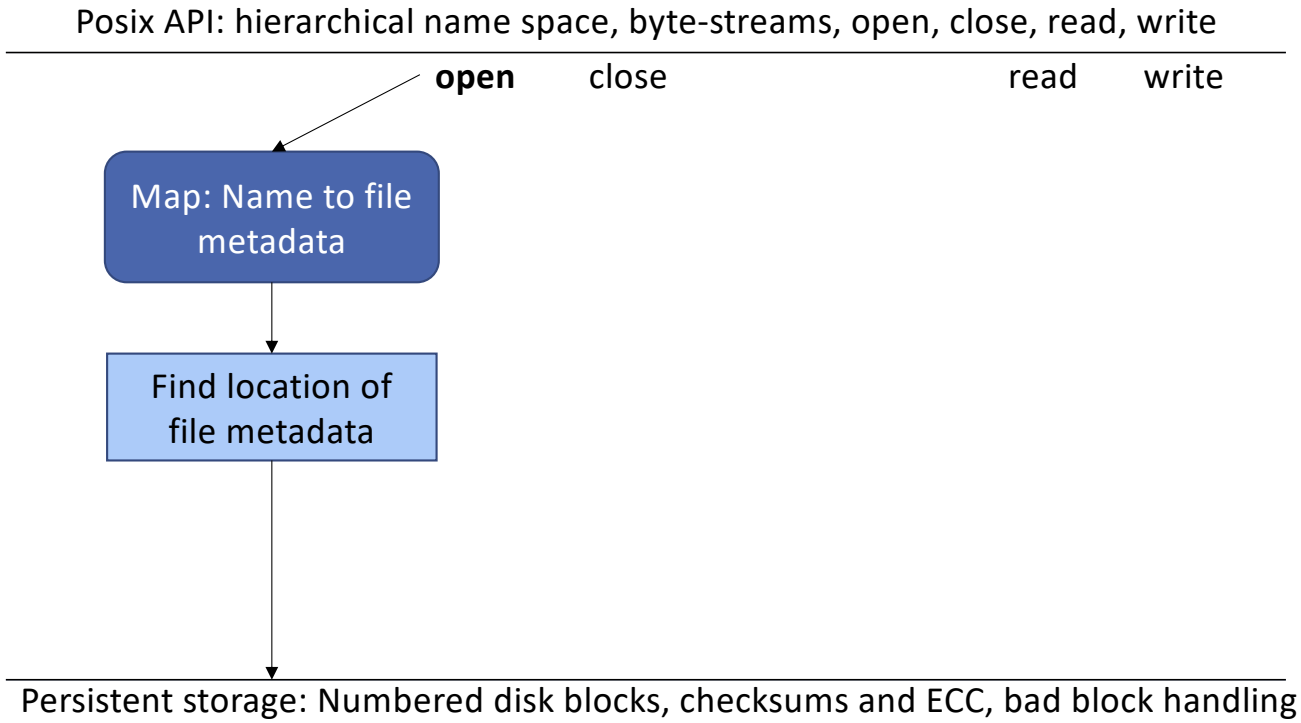
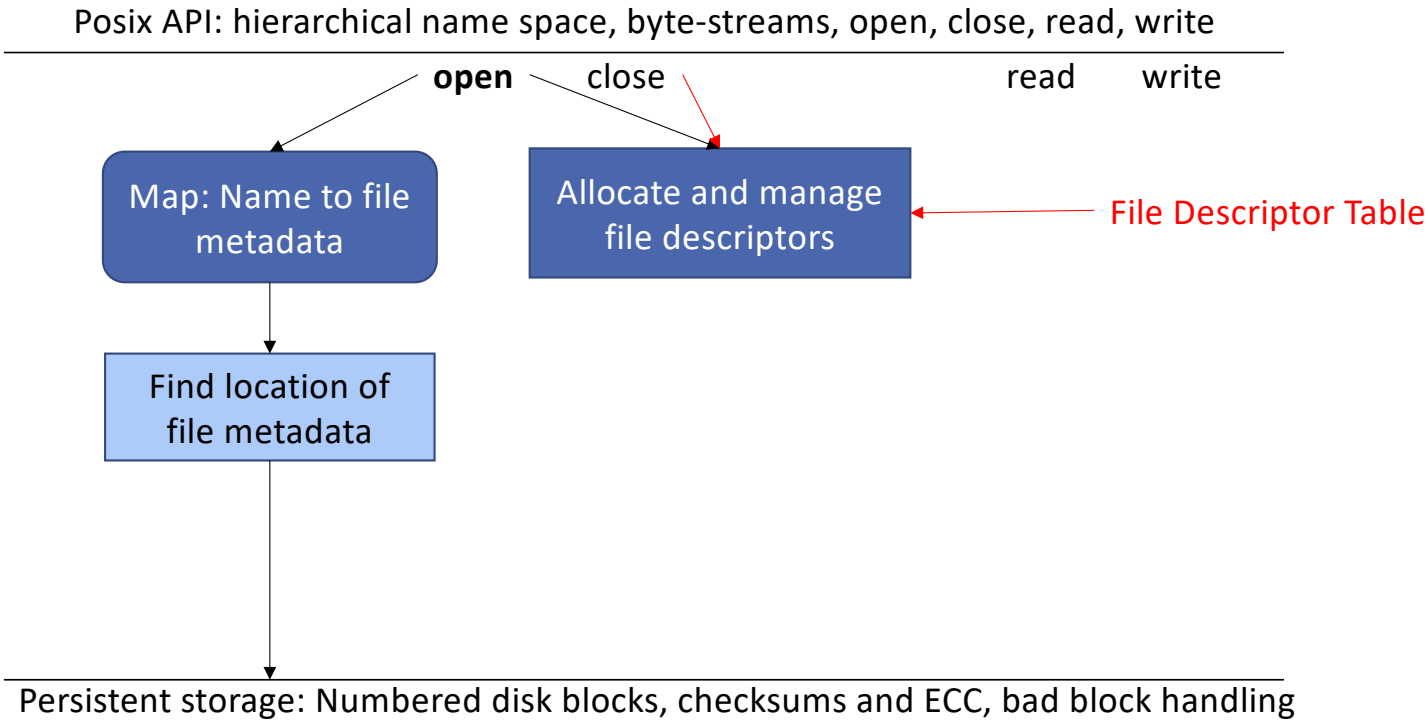
Today

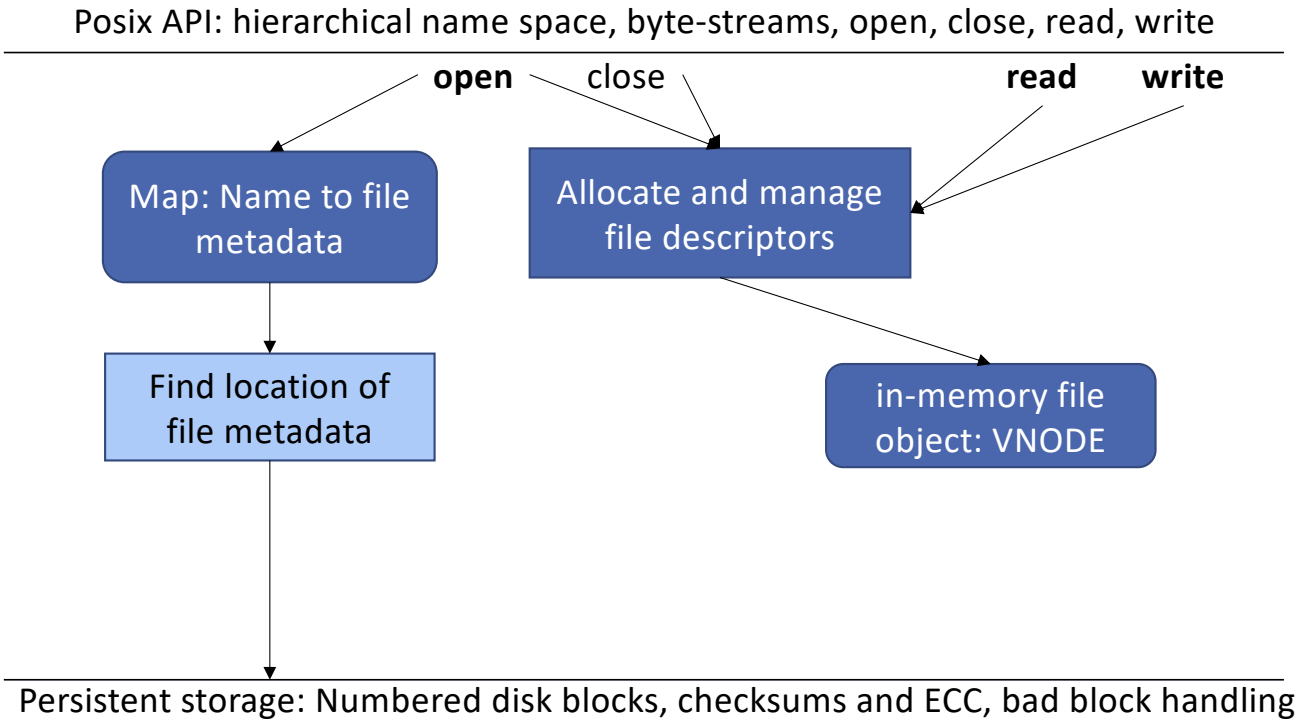
- Learning Outcomes
  - Identify the different things that a file system has to do to bridge between programmatic file system APIs and a storage device.
  - Define (in the context of file systems):
    - Metadata
    - File Descriptor Table
    - Vnode
- How we'll get there:
  - Given the file system system call API and how we have to talk to storage devices, we'll derive the kinds of functionality that need to fill that gap.
- Reading:
  - Not covered in the book.

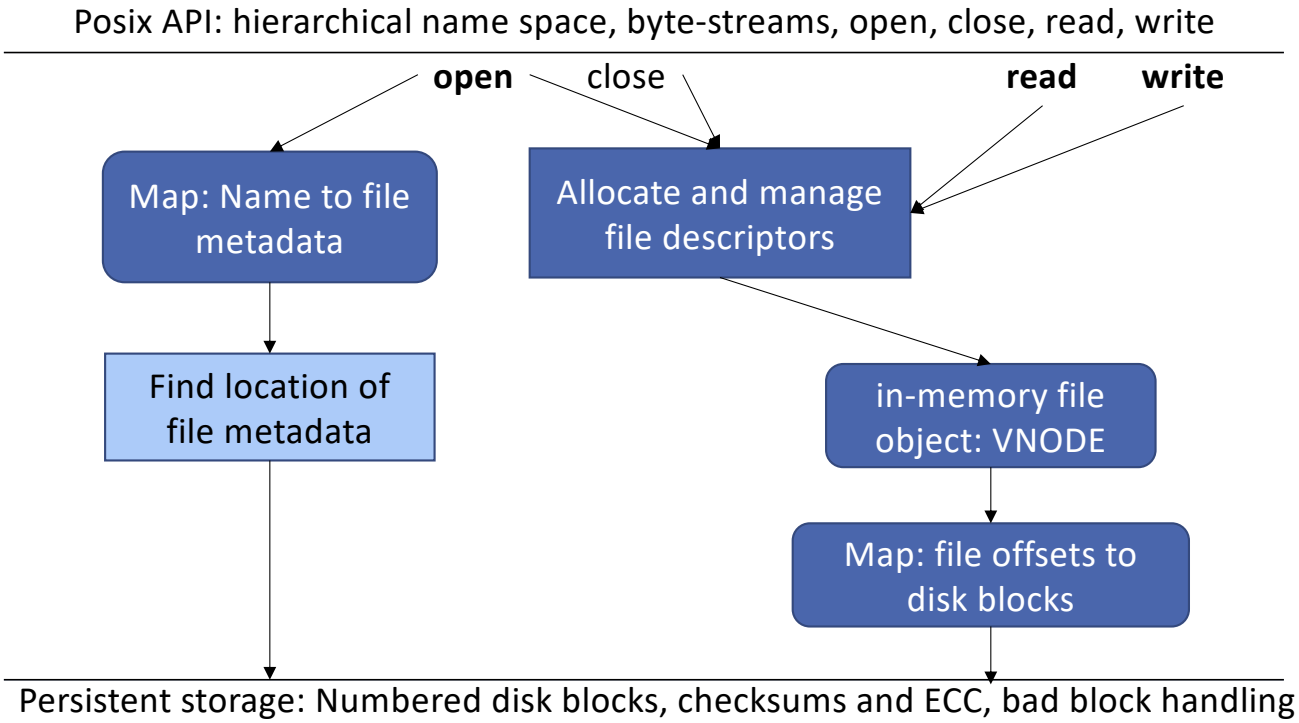
Posix API: hierarchical name space, byte-streams, open, close, read, write

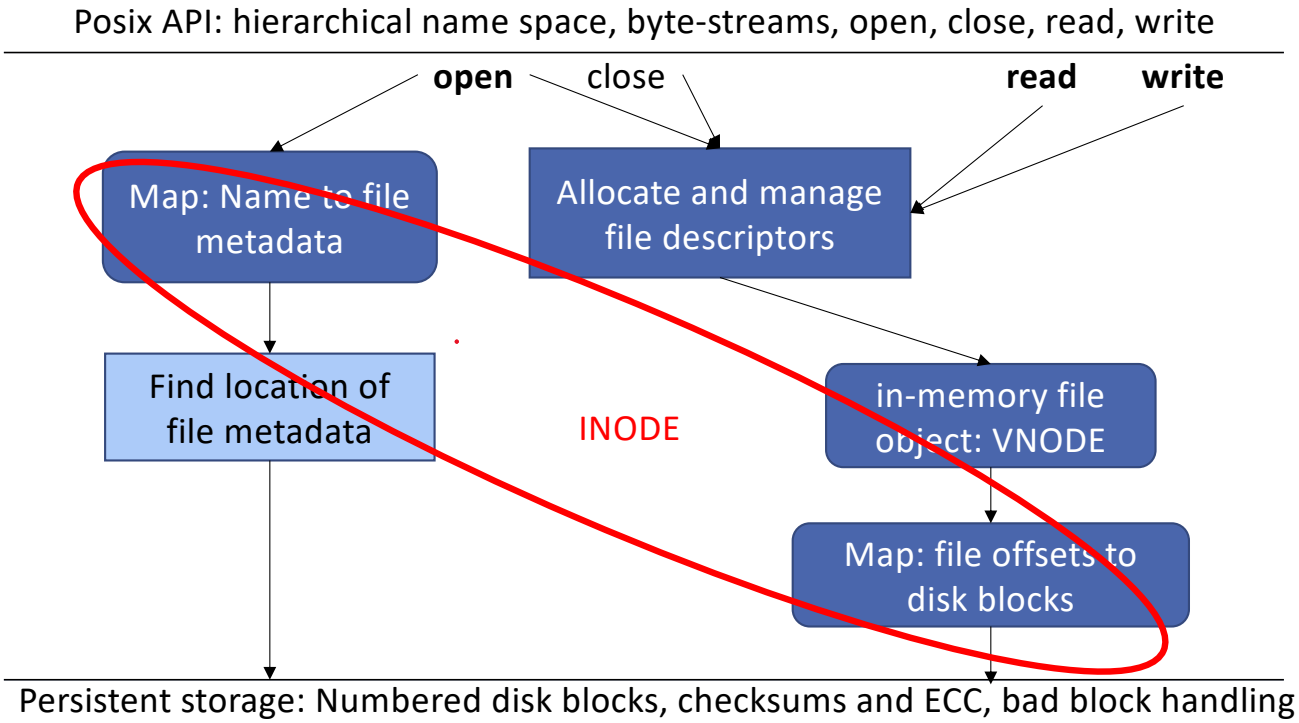
Persistent storage: Numbered disk blocks, checksums and ECC, bad block handling


Posix API: hierarchical name space, byte-streams, open, close, read, write
open
close
read
write
Map: Name to file metadata
Find location of file metadata
Persistent storage: Numbered disk blocks, checksums and ECC, bad block handling


Posix API: hierarchical name space, byte-streams, open, close, read, write
open
close
read
write
Map: Name to file metadata
Allocate and manage file descriptors
File Descriptor Table
Find location of file metadata
Persistent storage: Numbered disk blocks, checksums and ECC, bad block handling


Posix API: hierarchical name space, byte-streams, open, close, read, write
open
close
read
write
Map: Name to file metadata
Allocate and manage file descriptors
Find location of file metadata
in-memory file object: VNODE
Persistent storage: Numbered disk blocks, checksums and ECC, bad block handling


Posix API: hierarchical name space, byte-streams, open, close, read, write
open
close
read
write
Map: Name to file metadata
Allocate and manage file descriptors
Find location of file metadata
in-memory file object: VNODE
Map: file offsets to disk blocks
Persistent storage: Numbered disk blocks, checksums and ECC, bad block handling


Posix API: hierarchical name space, byte-streams, open, close, read, write
open
close
read
write
Map: Name to file metadata
Allocate and manage file descriptors
Find location of file metadata
INODE
in-memory file object: VNODE
Map: file offsets to disk blocks
Persistent storage: Numbered disk blocks, checksums and ECC, bad block handling

The Parts of a File System

- Metadata:
  - We have information about the entire file system.
  - We have information about each file (inode).
- Managing file descriptors:
  - Allocating/freeing file descriptors
  - Mapping file descriptors to in-memory objects representing files or directories (folders)
- Naming: Mapping a symbolic name to a particular file or directory (folder):
  - In-memory representation of the file (or directory/folder) that name represents.
  - Persistent structures that associate files with directories (folders)
- Storing files on disk: Mapping a file to its collection of blocks
  - Maintaining metadata for each file (or directory)
  - Deciding where to place files on disk

The Rest of this Unit

- File descriptor management and file sharing
- Representing files
- Implementing naming
- Putting it all together (case studies)