23 File Locking

When files are shared among processes, concurrent accesses can lead to race conditions, similar to those that can occur in shared memory. To allow cooperating processes to coordinate access to shared files, filesystem provides lock primitives. Dimensions in file locking:

- granularity
  - file lock: can lock entire file only
  - record lock: can lock arbitrary region of file (locks held on non-overlapping regions of a file do not conflict)
- semantics
  - shared lock: multiple processes can acquire shared locks for the same (region of) a file
  - exclusive lock: if a process hold an exclusive lock on a (region of) a file, no other process can hold a shared or exclusive lock on the same (region of a) file.
- enforcement
  - mandatory lock: OS prohibits access to files by processes that do not hold the appropriate lock.
  - advisory locks: OS does not restrict access to files based on lock ownership. Processes can chose to ignore locking.

24 Filesystem Reliability

Loss of data in a filesystem can have catastrophic effect (much worse than failure of hardware.) Need to ensure reasonable level of safety against data loss in the event of system failures. Three threats:

- accidental or malicious deletion of data by users
- media (disk) failure
- system crash during filesystem modifications, leaving data on disk in an inconsistent state

Backup: copy entire filesystem onto low-cost media (tape) at regular intervals (e.g., once a day). In the event of a failure of the primary media (disk), can replace media and restore data from backup media. Amount of data loss is limited to modifications that occured since the last backup.

Mirrored disks: multiple copies of the filesystem are maintained on independent disks. Disk writes update all the redundant disks in parallel. Used in applications that cannot tolerate any data loss (e.g., banking).

RAID disks: use multiple parallel disk drives for higher throughput and increased reliability. For example: each bit of a data byte is stored on one of eight disks. A ninth disk stores a *parity bit* for each data byte. Parity bit is the result of an XOR among the eight data bits of the corresponding data byte. As a result, can still recover the data if one of the nine disks fails.

Versioning filesystem: filesystem created a new version every time a file is modified. Old versions are kept until explicitly deleted. Can use copy-on-write to reduce storage requirements.

After a system crash in the middle of a filesystem operation, filesystem metadata may be in an inconsistent state. (Invariants of the on-disk data structures may not hold.) Example: a file was deleted, but its disk blocks have not yet been added to the freelist.
Solution 1: run a program during system startup that examines the entire filesystem, detects inconsistencies, and restores the invariants. In Unix, that programs is called fsck. Some of the conditions fsck checks and repairs:

- correct i-node reference counts
- missing blocks in the freelist
- blocks that are both in the freelist and part of a file
- incorrect information in the superblock
- out-of-range block and i-node numbers
- disconnected (unreachable) files and directories

Solution 2: keep a separate log

- write log entry describing operations about to be performed

- perform the operation on the filesystem
- delete the log entry
- after a crash:
- check the log
- if non-empty, perform operations described in the log