File System Reliability

CS439: Principles of Computer Systems

Bringing It Together

- Virtual address spaces are mapped onto physical address spaces in physical memory
  - Pages are mapped to frames!
  - Lots of policies to consider: demand paging, page replacement, working set, …
- The heap is part of a process's virtual address space (and is thus paged!)
  - Can be explicitly or automatically managed
- I/O devices are important for system usability.
  - Each one consists of the device, the controller, a bus, and a port
  - One of the most important I/O devices is the disk, which provides stable storage.
- File systems make the disk usable
  - Goals include: efficiency, usability, reliability
  - File systems demonstrate mechanical sympathy, including through reserved space, block groups, first-free allocation…)
  - Users organize files using directories
    - Directories are just special files that contain name/file number mappings
  - Programmers access the file system using the API
  - For reliability, we look for consistent metadata structures
  - In today's filesystems, we get consistency through journaling.

Vocabulary

data bitmap
inconsistency
update order
journal
write-back
UNIX
consistency
lost+found
asynchronous
atomicity
ad hoc
redundancy
journaling file systems
mirror
synchronous
roll back
write-through
buffer cache
parity
directory data
inode bitmap
throughput
serializable
write-behind updates
metadata
transactions
RAID
stripe
write-ahead logging
fsck
Barrier
reliability
recovery plan
order of operations
data
durability