CSE 451: Section 10 Handout
6/6/2019

File System & Logging.

File System:

Why do inodes need to be written to disk?

Which file do the first and second inodes represent?

The disk cannot be interacted with when holding a spinlock. How can synchronization be handled at the file system level?

When overwriting 1 block of data, how many times will `bwrite` be called?

When appending 1 block of data to an existing file, assuming the extent is large enough, how many times will `bwrite` be called?

When adding a new file, how many times will `bwrite` be called?
This would include allocating a new extent.

CSE 451: Section 10 Handout
6/6/2019

File System & Logging.

Logging:

Describe a scenario which can leave the file system corrupted (assume no logging).

What is a transaction?
How can the log be synchronized among transactions?

What's the point of the commit flag for the log?

What's the difference between a REDO and UNDO log?
What are some of the trade-offs between them?

Let's call the metadata for the log region the Log Header. What type of metadata needs to be stored in the Log Header?

For the following cases, describe what should happen on reboot, and how the state of the file system will look to the user after reboot:

- The system crashes during writes to the log region.
- The system crashes after the log has been written to disk and is committed.
- The system crashes after the log has been written to disk, but the commit bit has not been set.

Imagine the user makes a write system call that adds 1 block to a file and the system does not crash.
Assume REDO logging.
How many times is `bwrite` be called?