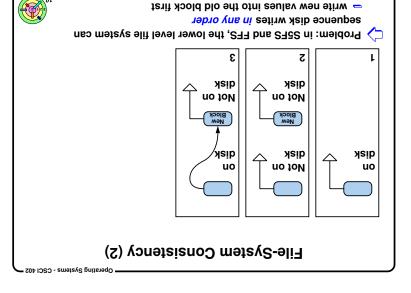
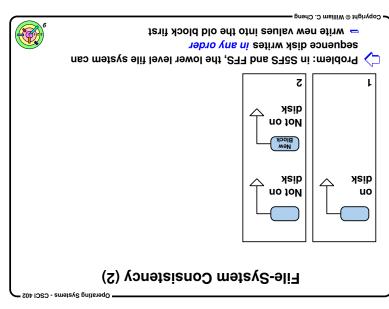
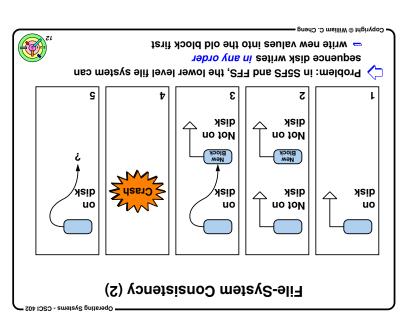
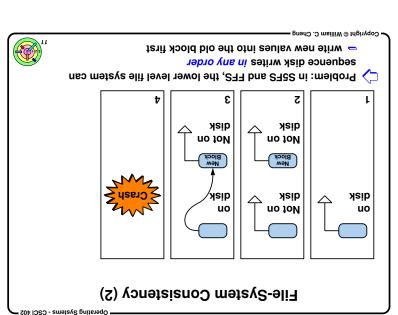


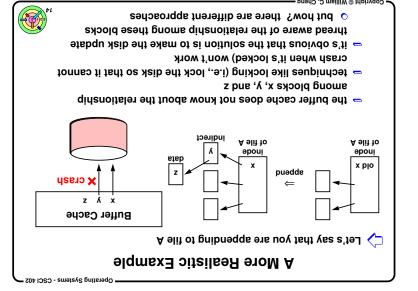
Copyright © William C. Cheng

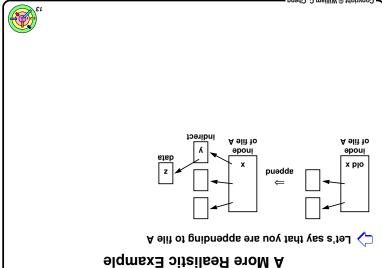


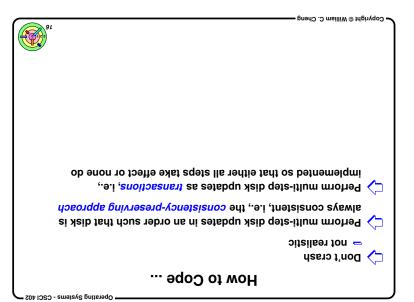


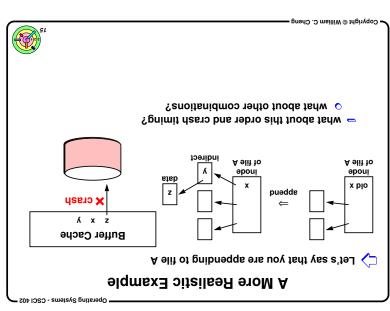


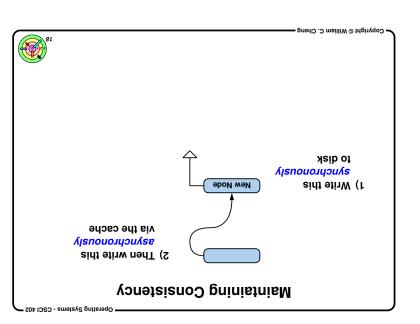


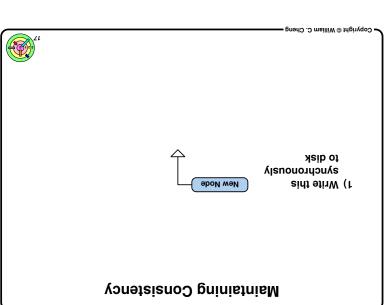


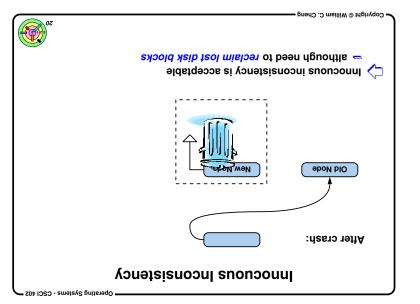


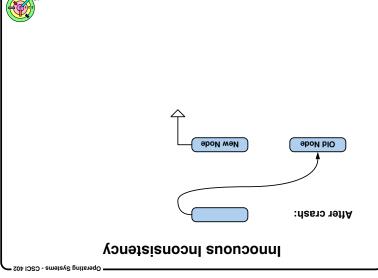


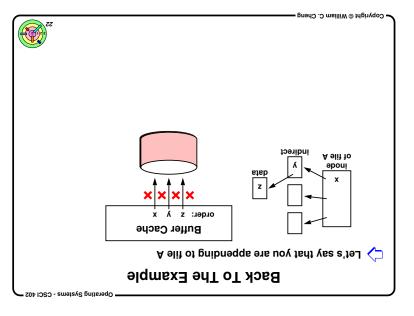


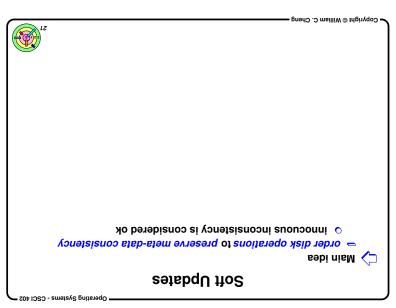


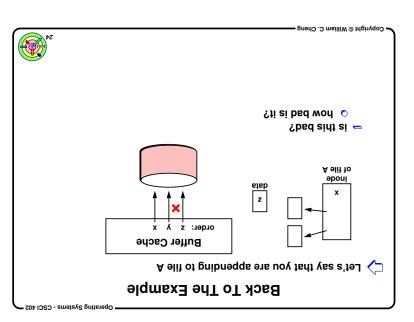


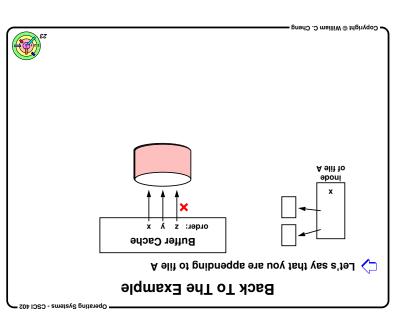


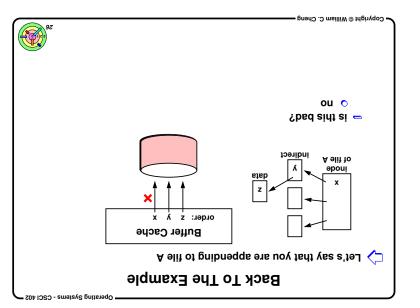


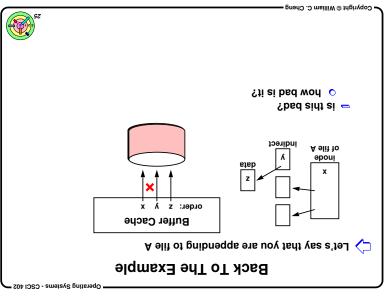


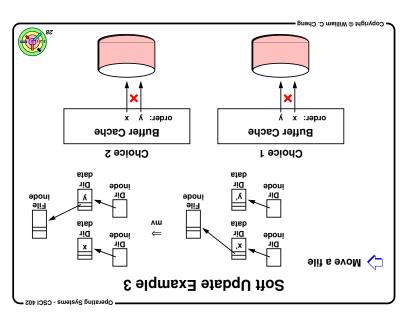


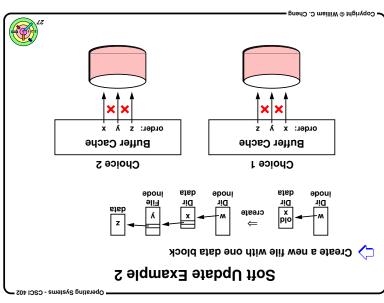


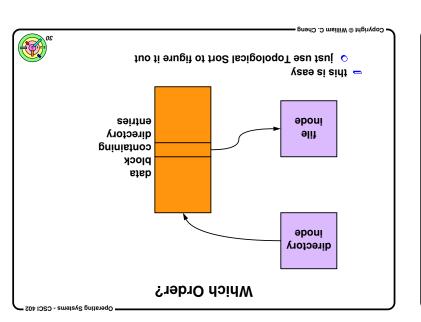


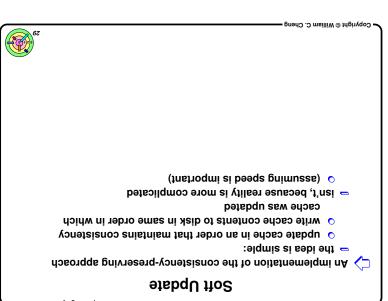


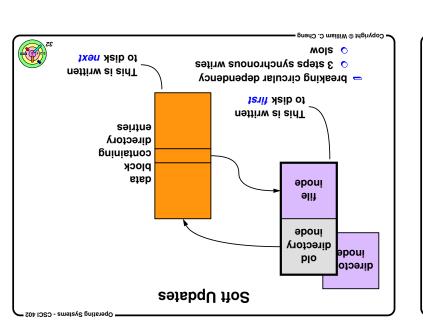


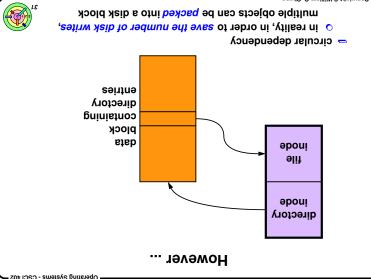


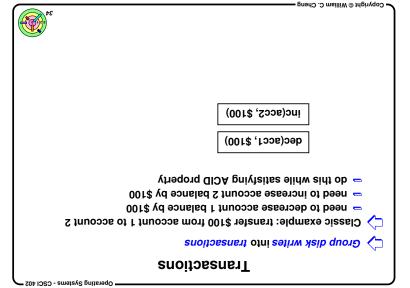


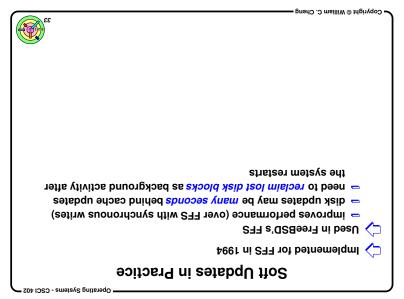


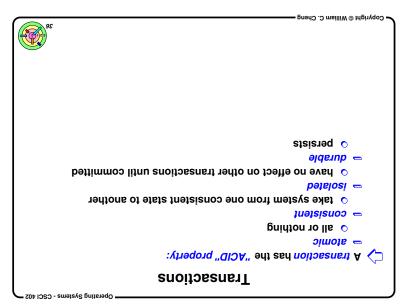


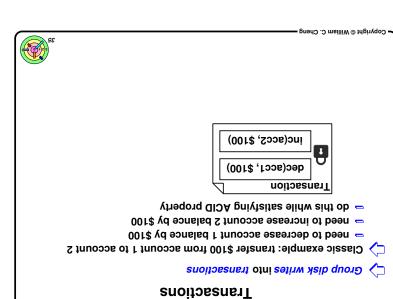


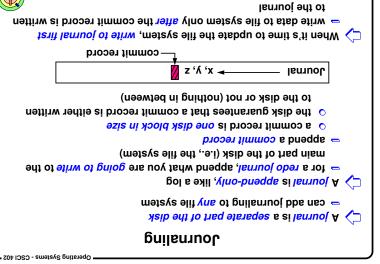


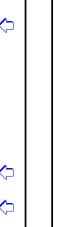












Copyright © William C. Cheng

opyright © William C. Ch

ou O

A elif to

Sped sidt si =

Journal -The journal is on the disk

A elif of gnibneqqe are appending to file A

- single write switches old state to new = steps of transaction written to disk, but old values remain gnigsq wobsd2 🔷 * "after images" of disk blocks are written into the journal vecord new contents: redo journaling ◇ "before images" of disk blocks are written into the journal

HOW?

record previous contents: undo journaling

- before updating disk with steps of transaction:

C Yournaling

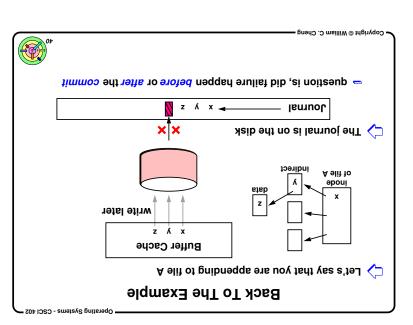


Весолегу

- finding all committed transactions If a redo journal is used, recovery involves
- o if system crashes in the middle of a recovery, no harm is done - redo (replay) all these transactions
- can perform recovery again and again
- doing it twice has the same effect as doing it once copying a disk block to the file system is idempotent, i.e.,
- end of the last committed transaction After recovery, the state of the file system is what it was at the

♦ dec(acc1, \$100) is not idempotent

- therefore, in a consistent state

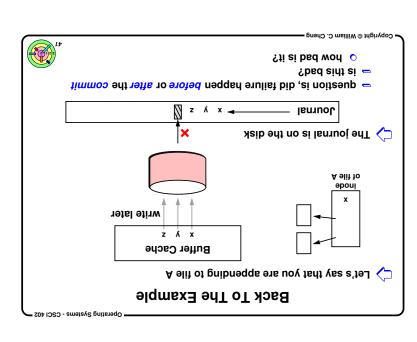


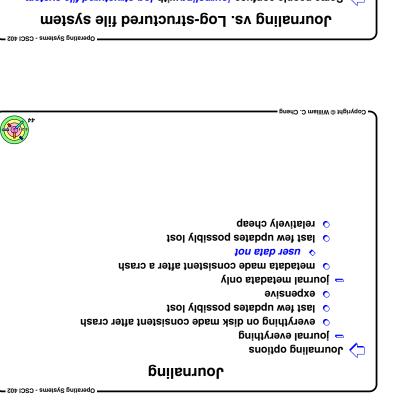
duestion is, did failure happen before or after the commit

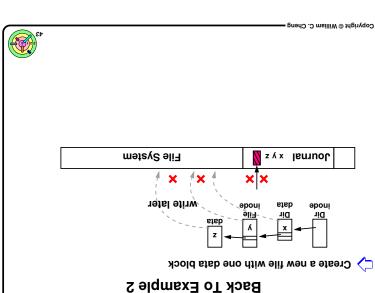
write later

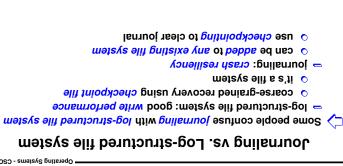
Buffer Cache

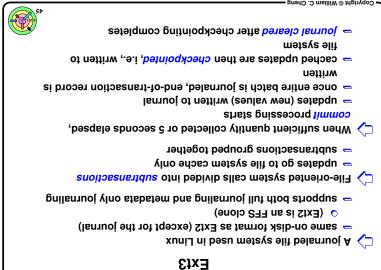
Back To The Example



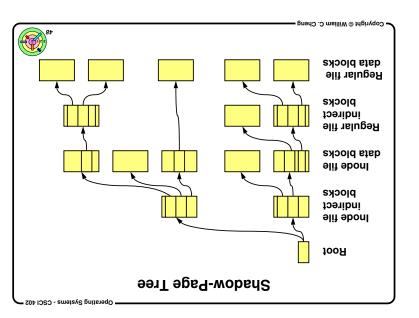


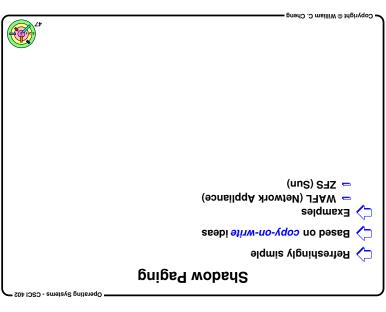


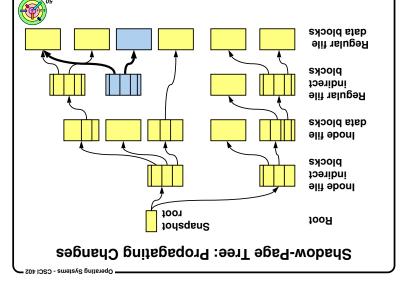




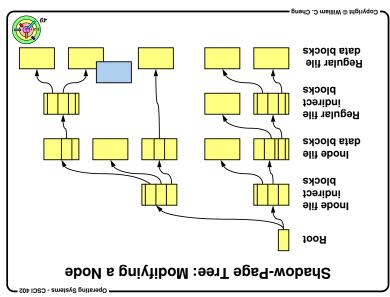
Oberating Systems - CSCI 402

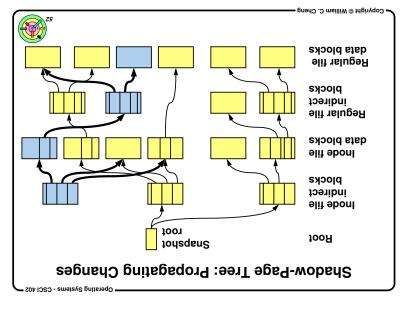


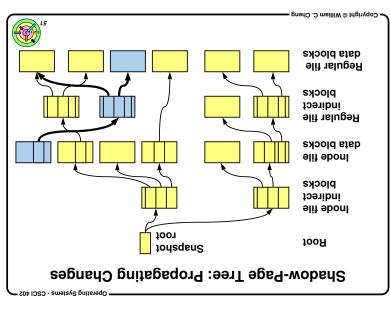


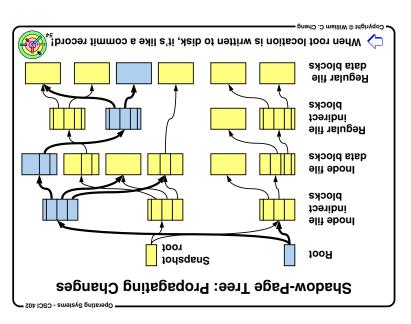


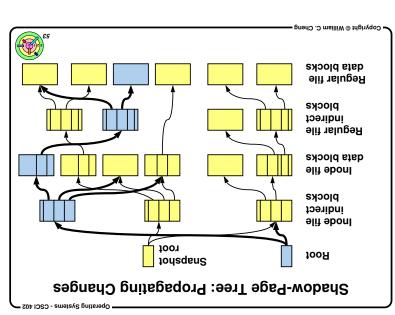
Copyright © William C. Cheng

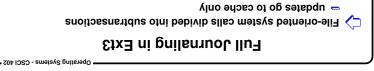












- subtransactions grouped together
- commit processing starts When sufficient quantity collected or 5 seconds elapsed,
- updates (new values) written to journal
- once entire batch is journaled, end-of-transaction record is
- acached updates are then checkpointed, i.e., written to
- journal cleared after checkpointing completes file system

