

Write-Ahead Logging

Once you have the create and append operations implemented, the next step will be to add crash protection.

Note: There are many ways to configure your log and API, I'm just going to introduce the one that seems the simplest for a single transaction at a time log.

The way I'm going to structure our log today is of the form:

LS	LS + 1	LS + 2	LS + 2	LS + 3	LS + 4	LS + 5	LS + 6
Start i	Tag x	Blk x	Tag y	Blk y	Tag z	Blk z	Commit i

Where:

- LS: the first block of the log (denoted in super block)
- Start i: Meta block that indicates the start of transaction i
- Tag x: A block to indicate the next block will be placed at block number x on successful commit
- Blk x: Is the data in block number x to be committed.
- Commit i: Indicates the end of transaction i. Once this block is written to disk, the in memory dirty blocks can be flushed to their respective blocks from the buffer cache.

API:

- `log_start_tx()`: Will write the start block to the log (having a local static variable incrementing tx id)
- `log_write(struct buf *)`: This will "replace" `bwrite` instead of flushing the buffer block to disk you will want to set it's dirty bit (the flags, see `B_DIRTY`) and add the tag and block data to the log.
- `log_end_tx()`: Will write the commit log (with id) to the log. After the commit record is flushed to disk, you can flush all your dirty blocks to the disk.
- `log_recover()`: Will walk through the log and analyze what needs to be done to recover from a potential crash.

Things to think about:

- How are you going to keep track of which blocks are part of this transaction?
- How big should your log region be on disk?
- How does recovery work?
- When should the recovery procedure be run?
- In what order should you perform the writes that make up a single transaction to ensure consistency?

2 Scenarios (How many blocks do you think will be in the log at commit?):

- 1) File create (____ blocks)
- 2) Single block append
 - a) Worst case (____ blocks)
 - b) Best case (____ blocks)
- 3) Multiblock append (as a function of n blocks) ***Not required for your implementation***
 - a) Worst case (____ blocks)
 - b) Best case (____ blocks)