1. FileHandler Important Methods:

1.1 open(String, OpenOption)

Big Picture

When a file is opened, I will check if the local copy is up to date. If not, I will download the content. Otherwise, just use the local copy

Corner Cases

* When writers want to write on the file, writers will allocate their own space without interfere with readers and other writers.

* If one slow reader reads a file and the file is updated in the middle, the slow reader will not see the dirty writes during its reads. If the second reader comes at this point, the second reader will see the updated file.

1.2 close(int)

Big Picture

When a file is closed, I will check if the file is modified in between. If the file is dirty, then the dirty copy will be pushed back to server.

Corner Cases

* If a writer allocate a private copy of the file, then the writer will delete this private copy during close.

1.3 unlink(String)

Big Picture

If a file is removed, I will first remove the local **used** cache copy and then call the remote server to remove the remote copy.

Corner Cases

If the local copy is being accessed by other readers, then the local cache copy will not be removed.

2. LRU design:

Big Picture

All of my cache function is within the Cache.java. It encapsulates all the LRU details.

Corner Cases

If the proxy is keeping read-only copies, nothing tricky. If the proxy is doing writes, then the cache will be updated during every write. In case the file becomes longer and eviction is needed during writes.