File Caching Proxy:

This system uses an open -close session semantics. The design of each of the file functions is explained below.

Open:
The proxy checks if it has the latest version of the requested file from the server. If there is a cache hit, a new private copy is created upon a write-open. Otherwise, the read copy is update to have the correct reader count. In case of a cache miss, the file is downloaded from the server. The versioning is done via the last modified time. All such times are relative to the server and the cahche time is not considered at all, as there may be inconsistencies in the clocks of the server and the client. A map is maintained to avoid such cases.

Close:
If the file to be closed is a read only copy, it is checked if the version of the file is not the latest and deleted accordingly. If it is a write copy, the changes are pushed to the server and the file copy is deleted.

Write/Read/Lseek: These functions are similar to the regular write/read/lseek.

Unlink: The unlink operation is perfomed only on the server side. Subsequent open operations will fail during the file information check from the server.

Cache:
The cache uses an LRU eviction policy. This is done by maintaing a linkedlist of the files that are used by proxy. Private write copies are never deleted by the cache class.

The files are downloaded/uploaded from the server in chunks of 1MB.

All the cache files are uniquely named and the subdirectories are handled properly. Testing locally didn't yield any errors/undefined bevhaviour. However, I am unable to find the reason why the read cache test is failing.