Suggestions to improve code:

1) Here cache is implemented with a double linked list. Searching cache is a very costly operation (O (n), where n is the max elements in cache). It can be implemented with hash table or radix tree where look up takes only minimal time. (My bad :( I started with doubly linked list and went too ahead with coding before realizing search takes much time.)

2) If the client tries to pin more files than available free cache, thread will block.

3) Don't have a proper interface for client to send request. Client operation is chosen on a random basis.

4) A timeout value can be set for an entry to get evicted from cache. With current implementation, once pinned entry will stay until client manually unpins it.

5) In file\_cache\_mutable\_file\_data (), no need to write data to local storage, if the file is already dirty. Buffer can be appended with new data and can be written together at once at the time of unpinning or file\_cache\_destroy () there by reducing write () calls;