FileClient:

```
main() {
    new FileClient();
    Naming.rebind("rmi://localhost:arg[1]/
                   fileclient")
    client.loop();
}
void loop() {
    while (true) {
        writebackThread.start();
        read filename or quit
        read w or r
            ↓
        exit!
        writebackThread.kill();
        if the requested file is not
            in the cache {
            if the current cached file
                is dirty
                    file.upload();
            file.download(filename, w);
        }
        file.launchEditor(w);
    }
}
boolean writeback() {
    file.state = writeback;
}
boolean invalidate() {
    file.state = invalid;
}
```

if writeback
file.upload()

FileServer:

```
main() {
    new FileServer();
    Naming.rebind("rmi://localhost:arg[0]
                   /fileserver");
    while (true)
        if receiving quit or exit
            ↓ save all cached files
        exit;
}
boolean upload(String client, String filename,
               FileContents contents) {
    search file cache for filename
    create a file if not found.
    file.upload(client, contents);
}
For each cached file object:
    synchronized boolean upload(client, contents) {
        for each client sharing this file
            Naming.lookup(...);
            client.invalidate();
        empty this list;
        save contents in this cache.
        change this file state;
    }
FileContents download(String client, String filename, String mode) {
    search file cache for filename
    send null, if not found
    file.download(client, mode);
}
```

For each cached file object:

synchronized FileContents download (client, mode r, w) {

chang this file state;

if necessary, call

* client.writeback();

return FileContents;

}