Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Distribute File System

by

Saquib Khan, 17311921

Master of Science in Computer Science
University of Dublin, Trinity College

Introduction:

A distributed system is a network that consists of autonomous computers that are connected using a distribution middleware. They help in sharing different resources and capabilities to provide users with a single and integrated coherent network.

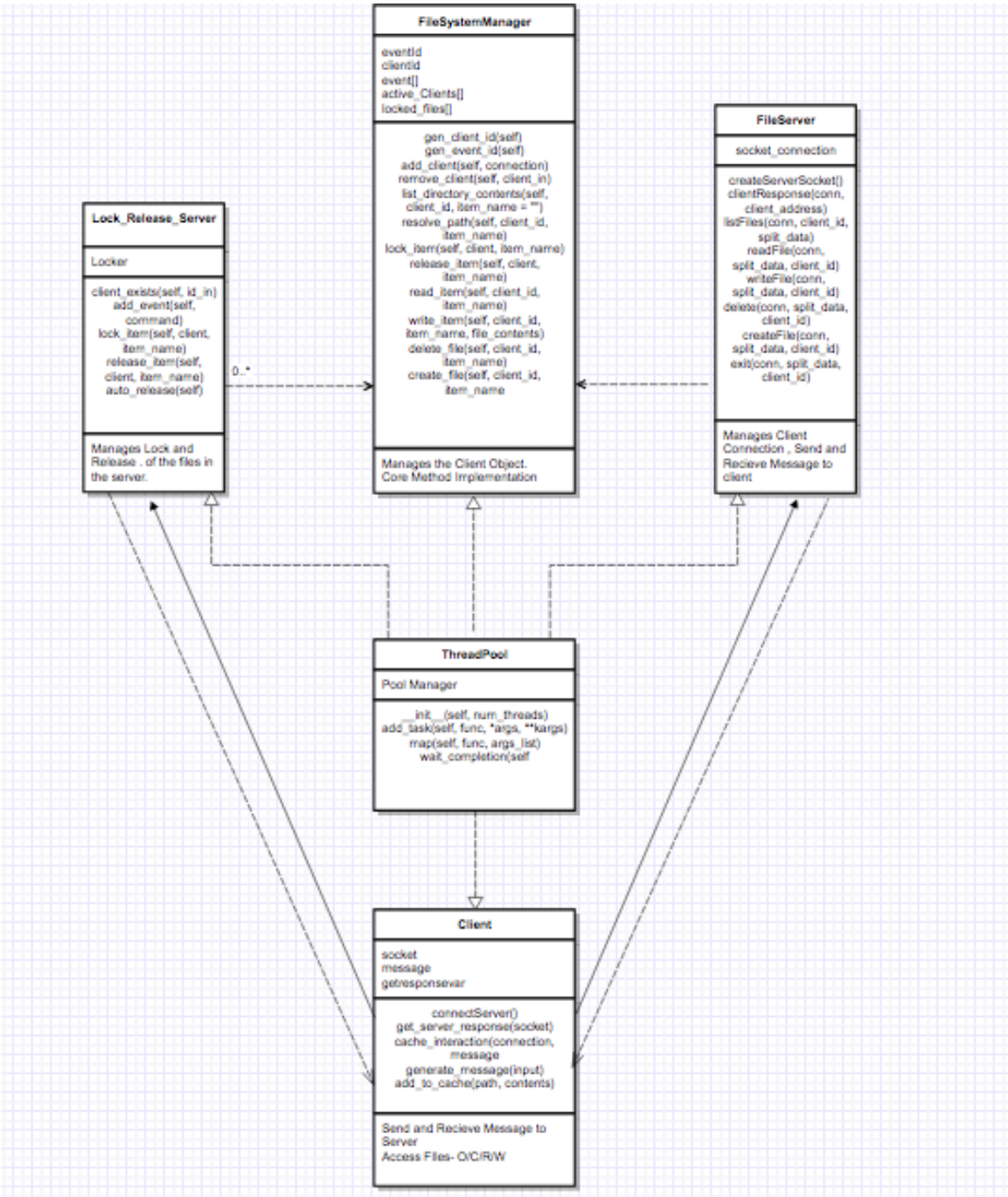
Key Features:

- Components in the system are concurrent. A distributed system allows resource sharing, including software by systems connected to the network at the same time.
- The components could be multiple but will generally be autonomous in nature.
- A global clock is not required in a distributed system. The systems can be spread across different geographies.
- Compared to other network models, there is greater fault tolerance in a distributed model.
- Price/performance ratio is much better.

Implementation:

I have implemented the below features, details are followed:

- Distributed Transparent File Access
- Directory Service
- Caching
- Lock Service



UML Diagram of Implemented Architecture

Details:

Distributed Transparent File Access/File Server:

This can be called as baseline of distributed file system and consist of a server which provides access to files on the machine on which it is executed and a client side file services that provides a language specific interface to the file system. Through various command user client can communicate with the fileserver and can read, write, create, delete any file on the servers.

Clients are unaware of the distribution of the files. The files are present on a totally different set of servers which are physically distant apart(may or may not be) and a single set of operations is provided to access these remote as well as the local files.

Directory Service:

Directory Service is a directory on the Fileserver(though it can be present on other server as well), through which client can list all the present files and create new files. Apart from it client can perform other functions as well such read, write and delete.

Client object is stored in the File and through various function client object can interact with the server entities.

Caching:

The purpose of caching is to make files quicker to access.. As expected, list of recently viewed items is stored on disk at the client. Functionality of auto lock release has been implemented to make sure items are discarded after a set period of time

Lock Service:

The lock service is an important user tool. To avoid deadlock a client must have mutually exclusive rights on file before writing it. In the current architecture locks are stored in a list on the server. A thread has the task of checking that all of the locks have an owner who is active on the server. This process runs every minute.

Working

As described in the above UML diagrams, The system has been implemented using the sockets services in Python environment.
One can start the Client Services by using the below command:

- python Server.py <Default Port: 9090>
- python Lock_Release_Server.py <Default Port: 9091>
- python Client.py

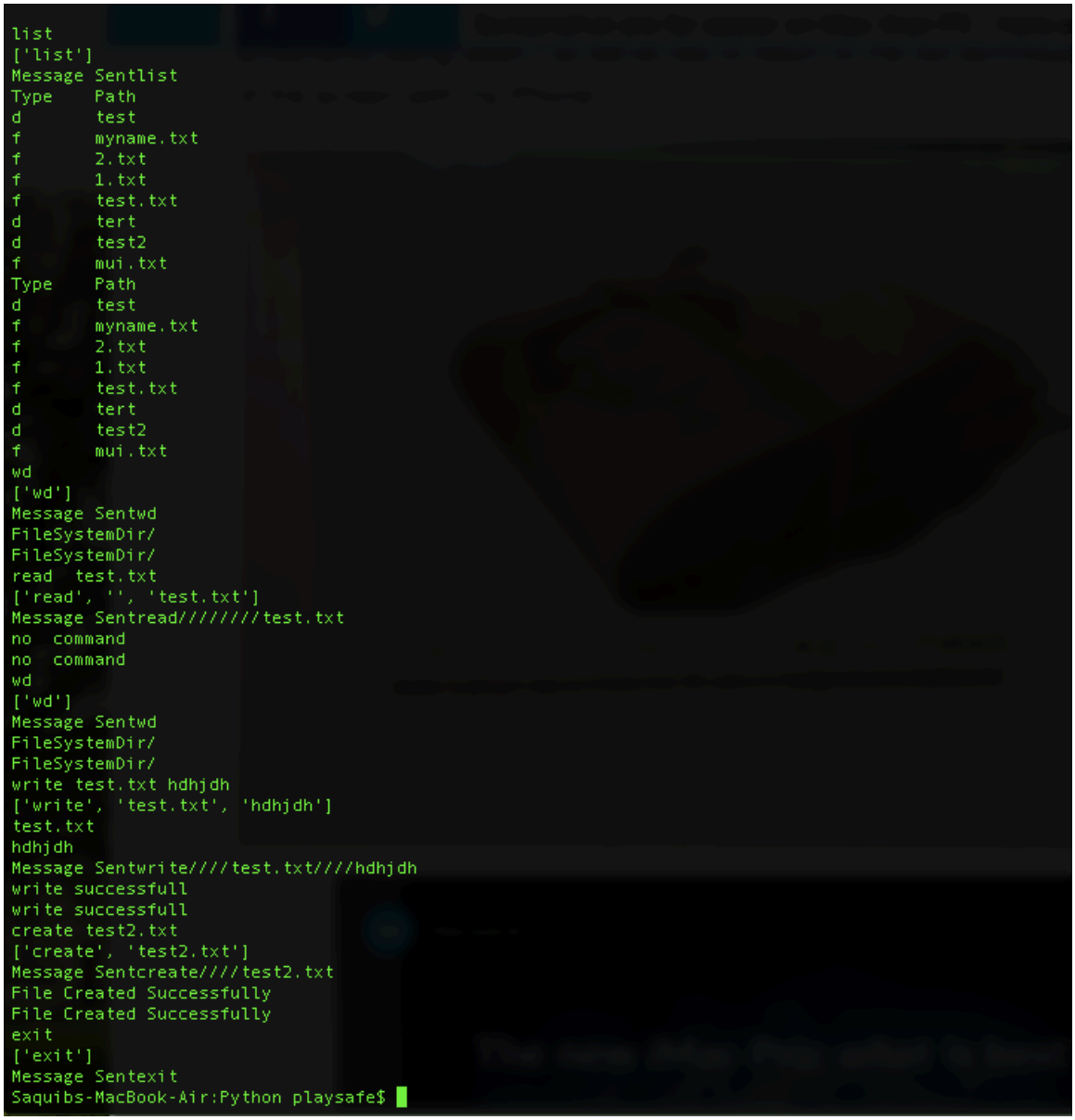
After starting all the services, Client can use the below set of commands for file operations:

- list: For listing the file in Current Directory.
- wd: For getting the information on current working directory.
- create <filename>: For creating a new file.
- read <filename>: For reading any file present in the directory.
- write <filename>: for writing in any file present in the directory.
- delete <filename>: For deleting any file present in the directory.
- lock <filename>: For taking a lock on any file.
- release <filename>: For releasing the file, which was locked previously.
- exit: To disconnect from the server.
- Kill Service: To shutdown the server.

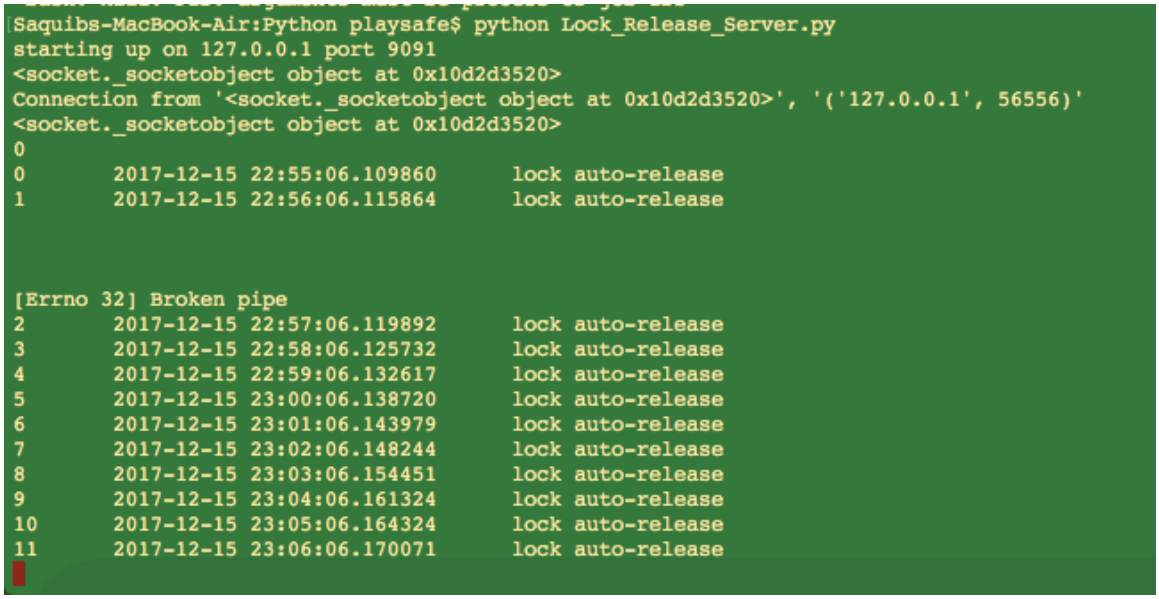
Screenshots:

```
Saquibs-MacBook-Air:Python playsafe$ python FileServer_Main.py
starting up on 127.0.0.1 port 9090
Connection from '<socket._socketobject object at 0x1071841a0>', '('127.0.0.1', 56555)'
0
list
[u'list']
I am sending to fs
0       2017-12-15 22:54:24.191966      lock auto-release
wd
read////////test.txt
wd
write////test.txt////hdhjdh
In Locking
FileSystemDir/test.txt
1       2017-12-15 22:55:13.531912      lock FileSystemDir/test.txt
2       2017-12-15 22:55:13.534023      write FileSystemDir/test.txt
3       2017-12-15 22:55:13.534091      release FileSystemDir/test.txt
4       2017-12-15 22:55:24.198036      lock auto-release
create////test2.txt
5       2017-12-15 22:56:04.859704      Created Filetest2.txt
6       2017-12-15 22:56:24.204469      lock auto-release
exit
cannot concatenate 'str' and 'list' objects
7       2017-12-15 22:57:24.208605      lock auto-release
8       2017-12-15 22:58:24.215933      lock auto-release
9       2017-12-15 22:59:24.219138      lock auto-release
10      2017-12-15 23:00:24.225864      lock auto-release
11      2017-12-15 23:01:24.230121      lock auto-release
12      2017-12-15 23:02:24.234616      lock auto-release
13      2017-12-15 23:03:24.238706      lock auto-release
  14    2017-12-15 23:04:24.246031      lock auto-release
15      2017-12-15 23:05:24.252831      lock auto-release
16      2017-12-15 23:06:24.259058      lock auto-release
```

File Server



Client Screen


Saquibs-MacBook-Air:Python playsafe$ python Lock_Release_Server.py
starting up on 127.0.0.1 port 9091
<socket._socketobject object at 0x10d2d3520>
Connection from '<socket._socketobject object at 0x10d2d3520>', '('127.0.0.1', 56556)'
<socket._socketobject object at 0x10d2d3520>
0
0 2017-12-15 22:55:06.109860 lock auto-release
1 2017-12-15 22:56:06.115864 lock auto-release
[Errno 32] Broken pipe
2 2017-12-15 22:57:06.119892 lock auto-release
3 2017-12-15 22:58:06.125732 lock auto-release
4 2017-12-15 22:59:06.132617 lock auto-release
5 2017-12-15 23:00:06.138720 lock auto-release
6 2017-12-15 23:01:06.143979 lock auto-release
7 2017-12-15 23:02:06.148244 lock auto-release
8 2017-12-15 23:03:06.154451 lock auto-release
9 2017-12-15 23:04:06.161324 lock auto-release
10 2017-12-15 23:05:06.164324 lock auto-release
11 2017-12-15 23:06:06.170071 lock auto-release

Lock Server

References:

Code Reference:
https://github.com/osulld13/DistributedSystemsFileSystem

Conceptual Reference:
http://www0.cs.ucl.ac.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf
http://barbie.uta.edu/~jli/Resources/MapReduce&Hadoop/Distributed%20Systems%20Principles%20and%20Paradigms.pdf