Distributed Systems

Project – 1

Submitted by

Harshil Patel -1001717222

Sreenidhi Timmaiahgari - 1001721679

**Assignment 1:**

**Implementation of the program**

The problem given was to implement a single-threaded file server that supports the four operations i.e upload, download, rename and delete. We used a message-oriented communication protocol UDP to establish a connection between a client and a server residing on a same machine and communicated using a different port.

Following is the step-by-step process of how we solved the problems:

1. We made a simple server that created a socket on localhost port: 8080 and listened to any incoming connections.

2. We made a simple client which connected to the localhost server on port 8080

3.The client sends request through the port once the server receives the request it fulfills the request using the same port.

4.We have created the following file-based operations as follows

Upload: The client reads the file locally and send it to the server through the socket connection. The server then implements the changes at its end accordingly

Download: After receiving download command, the server reads its file locally and sends the date to the client through the socket connection. The client then implements the changes at its end accordingly

Delete: Upon receiving the delete command, the server removes the file from its end

Rename: When the user enters the rename command at client-end, the new name is then sent to the server through the socket, which the server then implements at its end

5. After the client connects to the server, the server creates a handler thread that handles the command sent by the client, one all the packets are received the server again goes to listening mode.

**Issues Encountered**

**Assignment 3:**

**Implementation of the Program**

The problem given was to implement a computational server using synchronous RPC, we used TCP protocol to establish a connection between a client and a server.

1. We created a client stub to pack parameters into a message using function definitions for sending request, getting response, decode response and log response and a server stub to unpack the received parameters using function definitions for encoding the response, processing the request.

2.We also created a client and a server

**Issues Encountered**

**Assignment 4:**

**Implementation of the Program**

**Issues Encountered**

Synchronization between client server in a single threading environment

**Key Learnings**

In this project, we learnt about client server model and its complexities. We have learnt the synchronization techniques needed between a client and server.

**Single threaded and multithreaded version**

1) As a single shared file will be accessed by multiple threads in a multithreaded environment, keeping a synchronized version of the file would be a challenge.

2) The complexity increases as the number of threads grow in a multithreaded environment.

**Locking in a server**

**Distributed locking in a multi-threaded server**

Locking is necessary when a single shared resource is being accessed by multiple threads. In case of our project, this shared resource was a single file that was being read both at the client as well as server side by a single thread. If multiple concurrent accesses are given to the shared resource, it can cause issues which may leave our data inconsistent.