**DISTRIBUTED SYSTEMS**

**PROJECT-1**

**TEAM MEMBERS:**

**Ajay Venkatesha (1001861936)**

**Sayali Dilip Deshmukh(1001966628)**

**DECLARATION:**

I have neither given nor received unauthorized assistance on this work.

**Ajay Venkatesha** : Worked on Part 1 and Partial part 2

**Sayali Dilip Deshmukh** : Worked on Partial part 2 and part 3

Worked together on Readme file and Report

**INTRODUCTION:**

In this programming project, we are implementing a simple file upload and download service and a computation service using remote procedure call (RPC) based communication.

The file server is supporting 4 operations that are

1)UPLOAD

2)DOWNLOAD

3)DELETE

4)RENAME

Used different folders to hold files downloaded to the client or uploaded to the server. Implemented a computation server to support add(i, j), and sort(array A) operations using synchronous, asynchronous RPC

We are using Python Programming language for overall implementation and execution

The file transfer between the client and the server can be made transparent to users and automatically handled by a helper thread.

**SCREENSHOTS AND FILE OPERATIONS DESCRIPTION:**

**Addition and sorting operations of Synchronous server:**

1)Here we provided the inputs for addition which are 8 and 7 and showing a result like 15

2)For sorting we have provided input as [4,7,1,8,2,3,0,9]

and is sorting properly and giving results as shown below

Text

Description automatically generated

**2) Addition and sorting operations of Asynchronous server:**

Text

Description automatically generated

Text

Description automatically generated

This process involves the client requesting an operation and the server acknowledging it. The client continues to go on with his tasks. As a result, the server does not provide the result to the client until it is requested by the client.

**3)File checker running to see if any file operations have been performed**

Text

Description automatically generated

4)During the execution 4 file operations have been performed that are upload, download, rename and delete.

5)Any changes that we are doing on the client side are automatically reflected in the server-side

6)For these operations to be performed there is a function written for every single operation in the python file.

We have implemented the multi-threaded server file that supports the actions upload, download, delete and rename.

**Upload:**

It transfers files from the client directory to the server directory.

**Rename:**

In this process, you can rename the files in the server directory by simply specifying the old and new filenames.

**Download:**

In this, we download the file present in the server to the client

**Delete:**

In this, we can delete files from the server.

During this it creates drop-box-like synchronization between client and server using a helper thread.

**ISSUES WE ENCOUNTERED:**

To close the socket with one thread, we had to change the place where the connections were closed. Overall we face initial problems during socket connection

**WORKING AS A TEAM:**

We worked as a group where we isolated the errands between us and worked together on required functionalities.

**PROJECT OUTCOMES:**

* Implemented RPC calls by sending requests remotely to a server for computation.
* In addition, I learned how to utilize multithreading to synchronize multiple tasks for concurrency
* A client-server model was used to learn how to synchronize files and folders.