2228-CSE-5306-001-DISTRIBUTED SYSTEMS
PROJECT-1

Name	UTA ID
SOUNDARYA POWN	1002036408
SHARNI REDDY ARJULA	1002091315

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

Sign: Soundarya Pown **Date:** 18-09-2022

Sign: Sharni Reddy Arjula **Date:** 18-09-2022

Part 1:

A multi-threaded file server that supports fundamental file operations including upload, download, rename, and delete in Python was implemented in part 01 of the project. We have utilised the xmlrpc package to enable communication between the server and client. All operation functions are defined in the server file, and the client file is where these functions are invoked. To store the files uploaded to the server and downloaded from the server separately, we have created two folders i.e., Client Folder and Server Folder.

We gained knowledge of various libraries, like xmlrpc and threading, while putting this component into practice. Since we are new to these libraries, writing the server file, particularly during the download and upload process, presented the most obstacle for us.

Part 2 :

We implemented a synchronised storage service in Python in Part 2. We have utilised the xmlrpc package to enable communication between the server and client. Additionally, we used libraries like watch dog to track file system events including creation, modification, and deletion of files as well as the FileSystemEventHandler function to respond to those events as they happened.

At first, we attempted to write the synchronisation in C++ but ran into problems and had a hard time locating libraries. After realising that utilising sync from dirsync is not recommended, We used the watchdog library to implement. To hold the files, we have made two folders, ClientFolder and ServerFolder.

Part 3 :

Using synchronous and asynchronous RPCs, we implemented a computation server in Part 3 to enable add(i, j) and sort(array A) functions. We defined the add and sort functions on the server file and called them from the client file to accomplish these tasks. Instead of providing custom input into the code, we have provided input manually. To establish communication between the client and server, Xmlrpc package is employed.

We have learned the distinction between synchronous and asynchronous RPCs while implementing this part. Multicall functions were employed to achieve asynchronous RPC.

WORK DISTRIBUTION:

SOUNDARYA POWN (1002036408)

- Part 1- Implemented Upload and Rename operations
- Part 3 implementation and report
- README file

SHARNI REDDY ARJULA (1002091315)

- Part 1- Implemented Download and Delete
- Part 2 implementation and report
- README file