

Report Project 1

I have neither given nor received unauthorized assistance on this work. I will not post the project description and the solution online.

Sign: 1. Vaibhavi Purushottam Arjunwadkar

Date: 8 Feb 2023

2. Mrunmai Nitin Magar

Date: 8 Feb 2023

Group Member Information:

Member 1: Vaibhavi Purushottam Arjunwadkar (1001826818)

Member 2: Mrunmai Nitin Magar (1002092125)

Implementation:

We have used python's xml-rpc standard library file for handling requests between client and server. The module employs XML in the background. for data serialization between the server and client. We have created two separate files for each of the three tasks, one for the client-side coding and other for the server-side coding. According to requirement in the task, we have created functions in the server and then we execute these functions from the client side. Client can execute by http to rpc calls to server. Client needs to create proxy to access server-side functions. Proxy binds IP address to the port and then this port is used by client to make rpc call. This is how client communicates with server.

Part 1:

How we have Implemented:

- Created two folders one for client and one for server
- Upload option can be used to create the file in the server which we have created in the client
- Rename option can be used to rename the specific file at client as well as at the server side.
- Delete option can be used to delete file from both client as well as from the server side.
- On server side it shows POST and 200 as a status code with IP address and date as I have kept the logging value as a True at the server-side coding.

What we have learned:

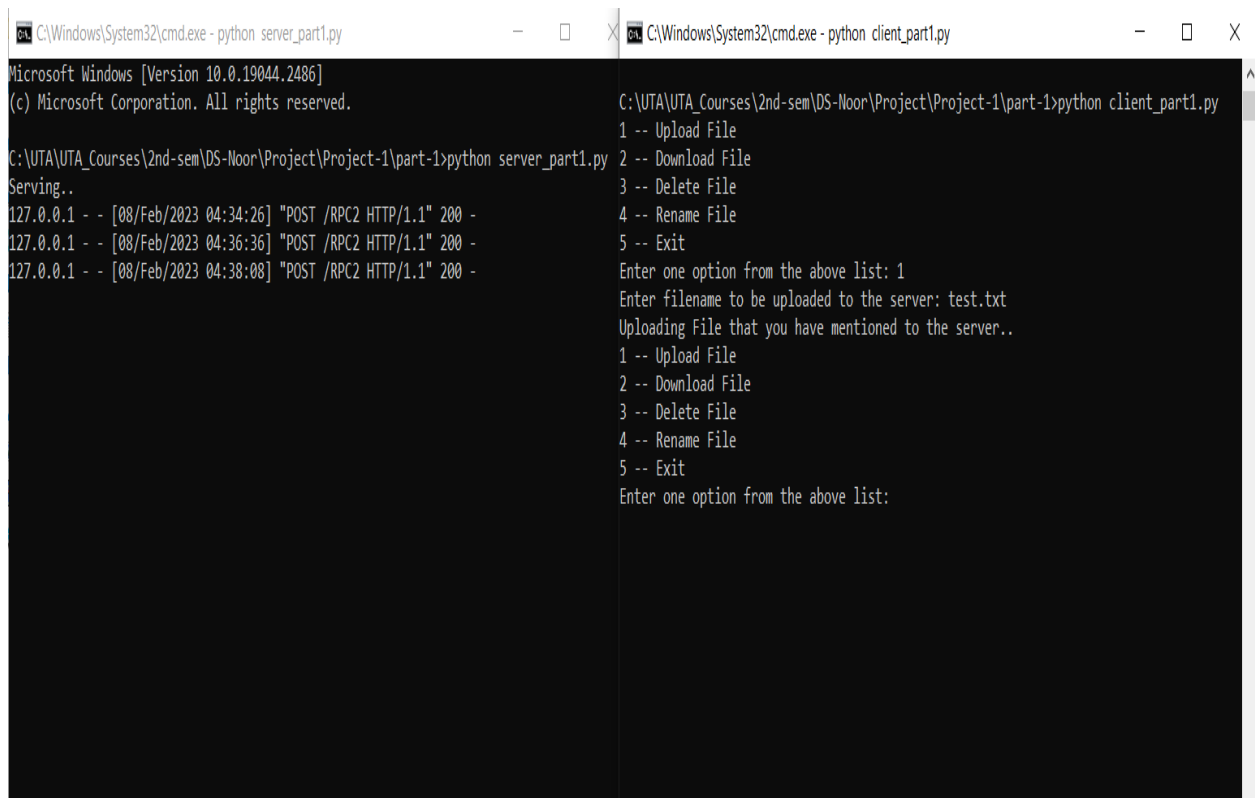
- How the xml-rpc library works in python
- How to follow pythonic way of coding to get a clean and understandable code.
- The benefit of using xml-rpc module which saves lot of time and energy instead of using traditional way of socket programming.

Issues Faced:

We both are new to python language and was not aware of how to use or implement xml-rpc. We have faced lot of issue like what kind of functions are supported by xml-rpc and when to use which function etc.

Screenshots for each operation:

Upload



```
C:\Windows\System32\cmd.exe - python server_part1.py
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-1>python server_part1.py
Serving..
127.0.0.1 - - [08/Feb/2023 04:34:26] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [08/Feb/2023 04:36:36] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [08/Feb/2023 04:38:08] "POST /RPC2 HTTP/1.1" 200 -

C:\Windows\System32\cmd.exe - python client_part1.py
C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-1>python client_part1.py
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list: 1
Enter filename to be uploaded to the server: test.txt
Uploading File that you have mentioned to the server..
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list:
```

Download

```
C:\Windows\System32\cmd.exe - python client_part1.py
C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-1>python client_part1.py
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list: 2
Enter filename to be download from the server: test.txt
File has been downloaded from the server
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list: _
```

Delete

```
C:\Windows\System32\cmd.exe - python client_part1.py
C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-1>python client_part1.py
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list: 3
Enter filename to be deleted on both client and server: test.txt
Removed test.txt from client
Removed test.txt from server
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list:
```

Rename

```
C:\Windows\System32\cmd.exe - python client_part1.py
C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-1>python client_part1.py
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list: 4
Enter filename you wish to rename: test.txt
Enter the new filename: new.txt
Done in client
Done in server
1 -- Upload File
2 -- Download File
3 -- Delete File
4 -- Rename File
5 -- Exit
Enter one option from the above list: _
```

Part 2:

How we have Implemented:

- We have initialized current time and all the files in client folder before running the loop to check consistency.
- We have made the process/thread sleeps for 15 sec
- Scan function runs after every 15 secs and will check the below things:
 - List of files in client folder
 - Check the modified time of each file.
(If the modified time > current time = file is modified)
 - Upload file to server
 - Delete the file in server

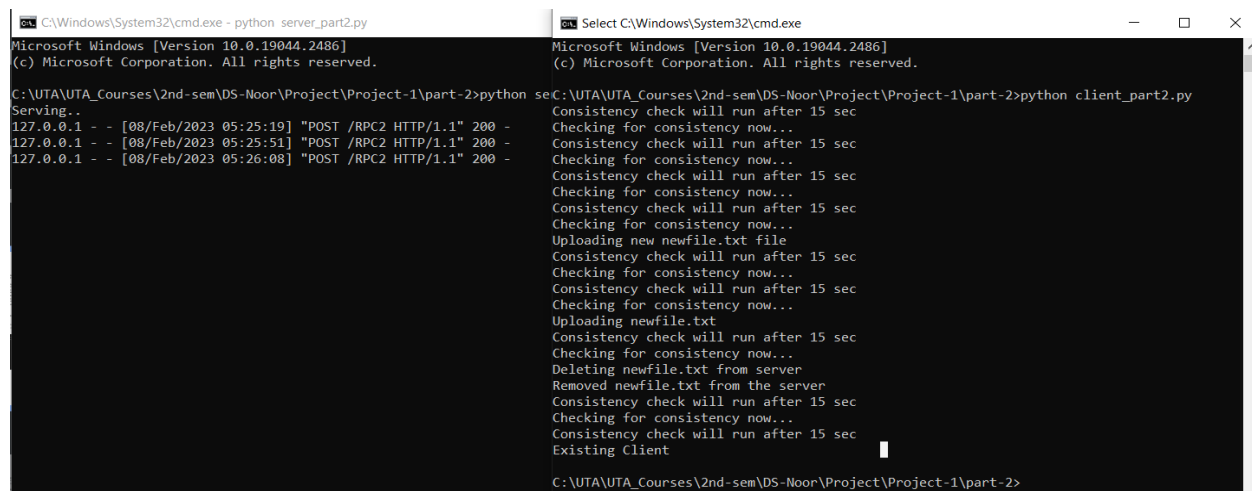
What we have learned:

- How and where to implement sleep function.
- How to reflect changes from client side to server side.

Issues Faced:

- Was unable to implement scan for consistency after every 15 seconds.
- Faced difficulty while comparing the current time and the modified time of the file while deletion.

Screenshot:



```

C:\Windows\System32\cmd.exe - python server_part2.py
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-2>python server_part2.py
Serving...
127.0.0.1 - - [08/Feb/2023 05:25:19] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [08/Feb/2023 05:25:51] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [08/Feb/2023 05:26:08] "POST /RPC2 HTTP/1.1" 200 -

C:\Windows\System32\cmd.exe
Select C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-2>python client_part2.py
Consistency check will run after 15 sec
Checking for consistency now...
Consistency check will run after 15 sec
Checking for consistency now...
Consistency check will run after 15 sec
Checking for consistency now...
Consistency check will run after 15 sec
Checking for consistency now...
Uploading new newfile.txt file
Consistency check will run after 15 sec
Checking for consistency now...
Consistency check will run after 15 sec
Checking for consistency now...
Uploading newfile.txt
Consistency check will run after 15 sec
Checking for consistency now...
Deleting newfile.txt from server
Removed newfile.txt from the server
Consistency check will run after 15 sec
Checking for consistency now...
Consistency check will run after 15 sec
Existing Client

C:\UTA\UTA_Courses\2nd-sem\DS-Noor\Project\Project-1\part-2>
```

Part 3:

How we have Implemented:

- Created two python files naming server_part3 and client_part3.
- To execute synchronous operations, we have used basic xml-rpc module function that triggers function from client to rpc call to server function
- To execute asynchronous operations, we have used threading module to start thread, save the entry in list.
- When addition is chosen, the user is required to enter the numbers to be added, and the result is displayed on the prompt.
- When sorting is selected, the user is asked to input the array's elements. At the end of the computation, a sorted list is expected.
- Result is saved on server side with id and result, and for displaying the result select from the menu given on client side by inputting given thread id.

What we have learned:

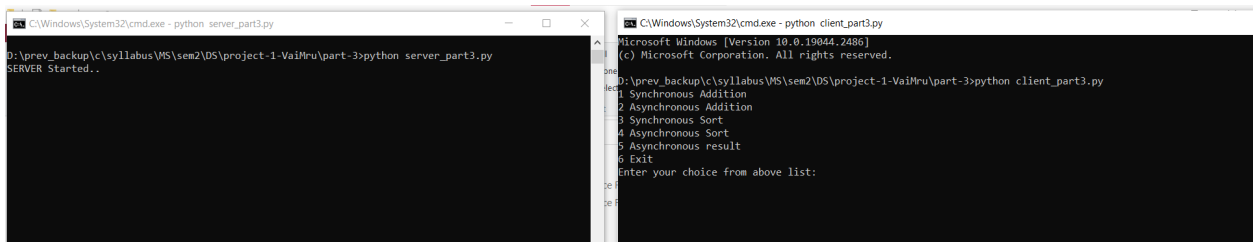
- we learnt to save result in server side instead of client side which is comparatively easy to implement.
- Learned about main thread and one of the limitation of python that is one thread policy.

Issues Faced:

- While writing and executing code for synchronous addition operation, we were having the output as a concatenated string of numbers and not the result of addition operation. We got rid of that error by correcting the logic for that block of code.

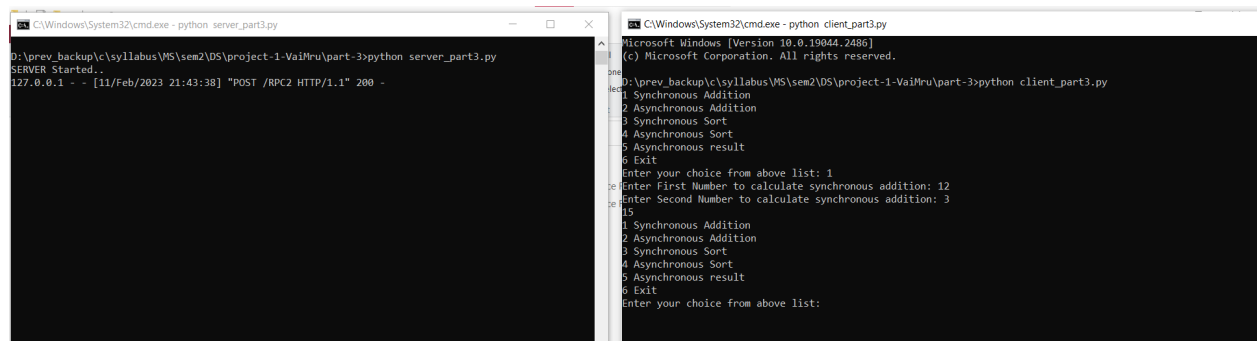
\Screenshots for each operation:

1. Displaying menu



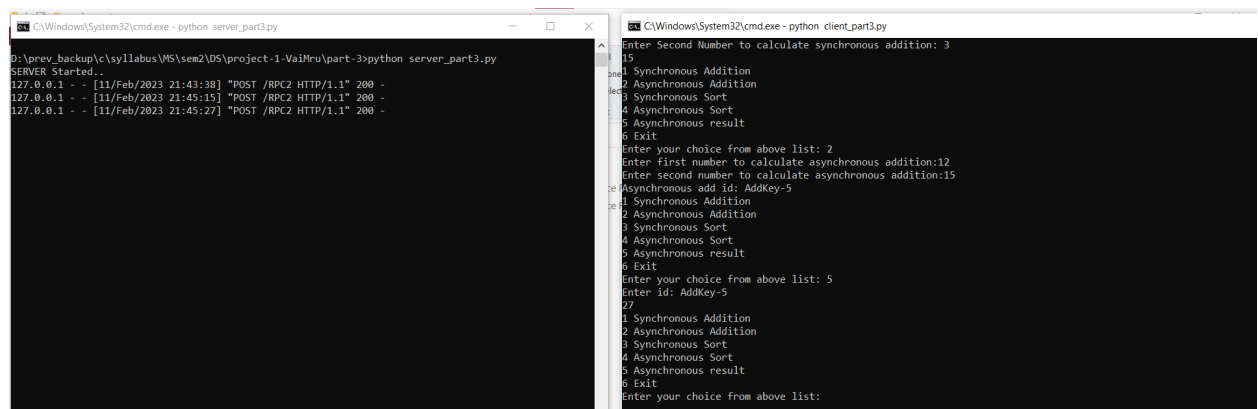
The image shows two side-by-side terminal windows. The left window, titled 'C:\Windows\System32\cmd.exe - python server_part3.py', displays the output of running the server program: 'D:\prev_backup\c\syllabus\MS\sem2\DS\project-1-VaiMru\part-3>python server_part3.py' followed by 'SERVER Started..'. The right window, titled 'C:\Windows\System32\cmd.exe - python client_part3.py', shows the output of running the client program: 'Microsoft Windows [Version 10.0.19044.2486] (c) Microsoft Corporation. All rights reserved.' followed by a list of menu options: '1 Synchronous Addition', '2 Asynchronous Addition', '3 Synchronous Sort', '4 Asynchronous Sort', '5 Asynchronous result', '6 Exit', and 'Enter your choice from above list:'.

2. Synchronous operation (Addition)



The image shows two side-by-side terminal windows. The left window, titled 'C:\Windows\System32\cmd.exe - python server_part3.py', displays the output of running the server program: 'D:\prev_backup\c\syllabus\MS\sem2\DS\project-1-VaiMru\part-3>python server_part3.py' followed by 'SERVER Started..' and a log entry: '127.0.0.1 - - [11/Feb/2023 21:43:38] "POST /RPC2 HTTP/1.1" 200 -'. The right window, titled 'C:\Windows\System32\cmd.exe - python client_part3.py', shows the output of running the client program: 'Microsoft Windows [Version 10.0.19044.2486] (c) Microsoft Corporation. All rights reserved.' followed by the same menu options as in the first screenshot. The user has entered '1' for 'Synchronous Addition'. The client then prompts for 'Enter First Number to calculate synchronous addition:' and the user enters '12'. It then prompts for 'Enter Second Number to calculate synchronous addition:' and the user enters '3'. The client displays the result '15' and then shows the menu options again.

3. Asynchronous operation (Addition)



The image shows two side-by-side terminal windows. The left window, titled 'C:\Windows\System32\cmd.exe - python server_part3.py', displays the output of running the server program: 'D:\prev_backup\c\syllabus\MS\sem2\DS\project-1-VaiMru\part-3>python server_part3.py' followed by 'SERVER Started..' and three log entries: '127.0.0.1 - - [11/Feb/2023 21:43:38] "POST /RPC2 HTTP/1.1" 200 -', '127.0.0.1 - - [11/Feb/2023 21:45:15] "POST /RPC2 HTTP/1.1" 200 -', and '127.0.0.1 - - [11/Feb/2023 21:45:27] "POST /RPC2 HTTP/1.1" 200 -'. The right window, titled 'C:\Windows\System32\cmd.exe - python client_part3.py', shows the output of running the client program: 'Microsoft Windows [Version 10.0.19044.2486] (c) Microsoft Corporation. All rights reserved.' followed by the same menu options as in the first screenshot. The user has entered '2' for 'Asynchronous Addition'. The client then prompts for 'Enter first number to calculate asynchronous addition:' and the user enters '12'. It then prompts for 'Enter second number to calculate asynchronous addition:' and the user enters '15'. The client displays the result '27' and then shows the menu options again.

4. Synchronous operation (Sort)

```
C:\Windows\System32\cmd.exe - python server_part3.py
0:\prev_backup\c\syllabus\MS\sem2\DS\project-1-VaiMru\part-3>python server_part3.py
SERVER Started..
127.0.0.1 - - [11/Feb/2023 21:43:38] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:45:15] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:45:27] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:46:42] "POST /RPC2 HTTP/1.1" 200 -

C:\Windows\System32\cmd.exe - python client_part3.py
1 Synchronous Addition
2 Asynchronous Addition
3 Synchronous Sort
4 Asynchronous Sort
5 Asynchronous result
6 Exit
Enter your choice from above list: 3
Enter array elements to be sorted: 12 54 11
['11', '12', '54']
1 Synchronous Addition
2 Asynchronous Addition
3 Synchronous Sort
4 Asynchronous Sort
5 Asynchronous result
6 Exit
Enter your choice from above list:
```

5. Asynchronous operation (Sort)

```
C:\Windows\System32\cmd.exe - python server_part3.py
0:\prev_backup\c\syllabus\MS\sem2\DS\project-1-VaiMru\part-3>python server_part3.py
SERVER Started..
127.0.0.1 - - [11/Feb/2023 21:43:38] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:45:15] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:45:27] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:46:42] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:47:41] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:48:17] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2023 21:48:35] "POST /RPC2 HTTP/1.1" 200 -

C:\Windows\System32\cmd.exe
1 Synchronous Addition
2 Asynchronous Addition
3 Synchronous Sort
4 Asynchronous Sort
5 Asynchronous result
6 Exit
Enter your choice from above list: 4
Enter array elements to be sorted: 12 1 33
Asynchronous add id: SortKey-2
1 Synchronous Addition
2 Asynchronous Addition
3 Synchronous Sort
4 Asynchronous Sort
5 Asynchronous result
6 Exit
Enter your choice from above list: 5
Enter id: SortKey-2
['1', '12', '33']
1 Synchronous Addition
2 Asynchronous Addition
3 Synchronous Sort
4 Asynchronous Sort
5 Asynchronous result
6 Exit
```