EEL-4736/EEL-5737 Principles of Computer System Design

Homework #4

Assigned: 10/8/2019; Due on 10/18/2018 – To be done individually

Part A

a) Problem set 9
b) Problem set 10
c) Exercise 6.4

Part B

In previous assignments, you implemented a file system that stored data in local memory. In this assignment, you will extend it to support storing data across a network, using the client-service approach.

The aim of this homework is to change the design of the file system such that the block storage resides in the memory of a server. You can use the code you have designed for the previous assignments as a starting point of this homework. Your focus will be on the memory interface – in previous assignment, the memory interface was programmed to call functions of a local memory storage; in this assignment, you will instead build upon an implementation of RPC for Python – XMLRPC – to issue remote methods to a service.

https://docs.python.org/2/library/xmlrpclib.html provides an introduction to the Python XMLRPC library.

To send/receive and store the data into the server requires you to serialize/marshal it. You may use the existing Python library pickle for serializing the data. Your implementation should allow configuration of server endpoints at the client by means of command-line arguments. Your client/server implementation should behave in exactly the same manner as the implementation of homework #3 if there are no failures in the network/server.

To complete this assignment, you must:

a) Implement the client-service functionality using XMLRPC, limiting your changes to client_stub.py and server_stub.py as needed.
b) Test and describe the behavior of your system when there is a failure of the network
c) Experiment and compare the performance of the client/service approach vs. the local memory approach for a test case of your choosing

It is suggested that you use a virtual machine in your computer to test and evaluate your system. You will need to edit two files one named ‘server_stub.py’ and the other named ‘client_stub.py’.

The server stub should contain a reference to a Memory.Operations() object and establish registered functions for each method in the Memory.Operations() instance. Instantiate the server and register the functions in server_stub.py. An excerpt of reference code is shown below

```
server = SimpleXMLRPCServer(("localhost",8000))
print ("Listening on port 8000...")

server.register_function(configure,                "configure")
```

Then you need to write the client_stub.py file. Inside the client stub, instantiate a proxy using

```
 self.proxy = xmlrpclib.ServerProxy("http://localhost:8000/").
```

To run the filesystem, open a terminal and run server_stub.py. Open another terminal and run FileSystem.py or whatever file you have as the top layer. If you have two different folders for the server partition of the file system and the client partition with their own copy of the config.py file, make sure they match. In order to use the client stub, changes need to be made to MemoryInterface.py. I have provided a new MemoryInterface.py file for your reference that contains calls to the client stub rather than to the memory layer.

Submission guidelines:

Turn in through Canvas following three files (attach individually):

1. Homework4partA.pdf – Solution to the questions in Part A
2. client_stub.py, server_stub.py – Your Python code
3. Homework4design.pdf – PDF describing the design of your implementation and tests you have conducted to check the functionality of your code.

Make sure your Python code is well commented and tested before submission. Assignment will be graded on the basis of design and functionality. There might be several ways to implement this assignment, choose one which is most feasible, concise and modular allowing you to have minimum changes in the existing code when you extend its functionality. Copy and the paste the python code (client_stub.py, server_stub.py) at the end of your homework4design.pdf file.