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

Homework #4

Assigned: 10/11/2017; Due 8 pm on **10/20/2017** – To be done individually

Part A

1. Problem set 10
2. Exercise 6.4

Part B

In this homework, you will build a client/server file system, which stores its data into a remote server instead of local computers. The aim of this homework is to change the design of the FUSE file system such that the storage of the file system data takes place in the memory of server(s) (not the client). You can use the code you have designed for the previous assignments as a starting point of this homework.

In this design, you will store file system data into two kinds of servers – a metaserver and a dataserver. Use metaserver to store the metadata of the files/directories and dataserver to store the data of the files. You should use an XMLRPC based server – simpleht.py (code provided). If needed you can modify the methods in the server code. The client (modified memory.py) will act as the FUSE handler as well as an XMLRPC client in your remote file system which connects to both metaserver and dataserver.

The metaserver will hold the keys (path of files/directories) and the values are the metadata of the corresponding file/directory. In data server key would be path and value would be the data of the file. To send/receive and store the data into the server requires you to serialize/marshal it. Use the existing Python library pickle for serializing the data.

Your implementation should allow configuration of server endpoints at the client by means of the command line arguments. Your implementation should also support a hierarchical name space, and splitting of file data into blocks, as per previous homeworks.

Program arguments and guidelines:

Servers (metaserver.py and dataserver.py) should take the port argument in the format specified below. The client program should take arguments as <fuse_dir> <metaserver-port> <dataserver_port>.

Example:

```
python metaserver.py --port 2222
python dataserver.py --port 3333
```

```
python remoteFS.py fusemount 2222 3333
```

Testing:

Your program will be tested for all the filesystem operations like previous assignments.

Submission Guidelines:

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

1. homework4partA.pdf – Solutions to questions in part A
2. remoteFS.py – Client that implements the remote file system with FUSE and XMLRPC
3. metaserver.py – Modified simpleht.py that hosts your metaserver
4. dataserver.py – Modified simpleht.py that hosts your dataserver
5. homework4design.pdf – PDF document describing the design of your implementation and tests you have conducted to verify the functionality.

Make sure your Python code is well commented and tested before submission. Assignment will be graded on the basis of design and functionality. Copy and the paste the python code (remoteFS.py) at the end of your homework4design.pdf file.