README:

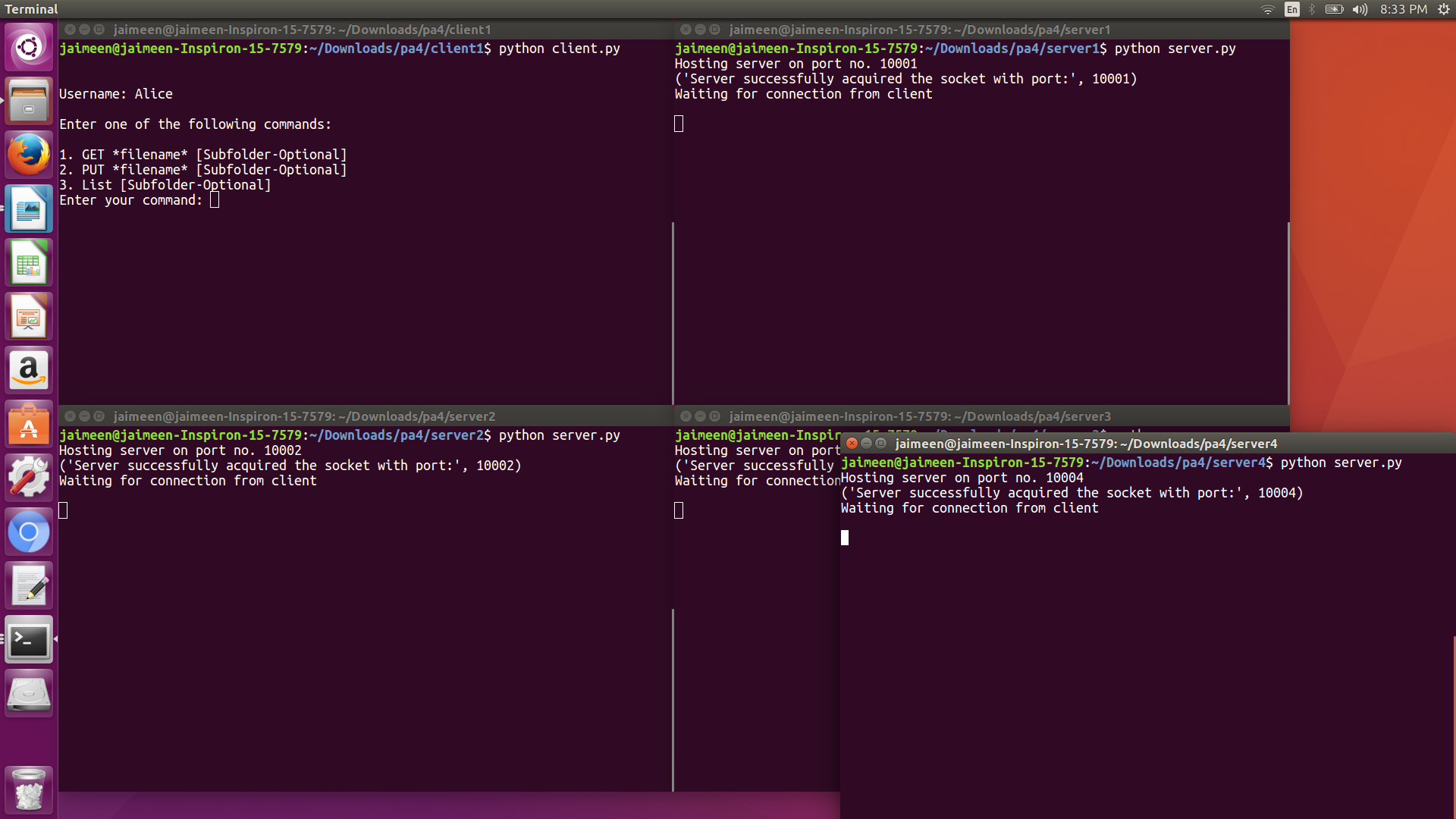
Name: Programming Assignment 4

Purpose: A Distributed file storage system with 4 servers and multiple clients

Date: 2 Dec 2017

Python version: 2.7.14

To run the code:



The client code searches for configuration file. If configuration file is not found or if formatting of configuration file is changed then an error is message is printed and client code quits. User name and password are obtained from configuration file and appended to a dictionary

The client code asks user to enter a command GET, PUT, LIST, followed by a file name. There’s a sub-folder option as well. If sub-folder is mentioned, then that sub-folder is created on user directory on server and file parts are saved in sub-folder. If sub-folder is not mentioned, then file parts are saved in user’s main directory on server. Once a valid command is entered, respective function is called which serves the request.

If command entered is PUT, then the file is encrypted and then broken into 4 parts. Hash of the file is calculated. Depending on modulo 4 values of hash, different parts of hash are saved on different servers. Before transferring content, we first check whether user name and password are valid or not. If user name and password are valid then we send the name of file part, then length of part file and finally, the contents of part file. File is sent in chunks of 1024 bytes. If final chunk is less than 1024 bytes, then remaining bytes are sent.

If command entered is List, then data about list from all files is appended to a list. This list is then checked for repetitive elements and they get removed. If parts of file are present in list, then presence of all 4 parts is checked. If all parts are present in list, then file’s name is appended to a new list. Finally, this new list is presented.

If command entered is Get, then file is requested from server 1 and server 3. If either of these servers are down, then file is requested from server 2 and server 4. User name and password are verified. First, the presence of file is checked on server. If file is present, then size of file part is received, then name of file is received. Finally, the contents of file are received. File is received in chunks of 1024 bytes. Last chunk is less than 1024 bytes. It is received by calculating the length of remaining bytes. At server, opposite actions occur.