Exercise 4: (Suggested time 3 hours)

This exercise involves creating a sample remote logging server and logging client applications.

Requirement

1. On the server side we need to wait for the client to connect.
2. Client should read data from kivi.log file line by line and send it to the server.
3. Server should write the data received from the client line by line to the log file.
4. The name of the file should identify which client the data was received from.

Implementation suggestions:

1. Create a Server class for network handling on server side
2. Create a Client class for network handling on client side
3. Create a Logging class to read and write the log file
4. Use the read method of the Logging class to read file in client
5. Use the write method of the Logging class to write file in server

Exercise 4: (Suggested time 3 hours)

You can create your client and server classes by extending the standard socket.socket class provided by Python
Eg:

```
class MySocketServer(socket.socket):
    def __init__(self):
        print "Init Server"

class MySocketClient(socket.socket):
    def __init__(self):
        print "Init Client"
```

Note:
This excerise will be a prerequisite for exercise 5 for the next session. In that exercise we will cover handling of multiple connections and processing of the log data on the server side.