Server Classes:

Server:

Contains main method.

ExitManager:

Allows the server to exit cleanly.

ServerRunnable:

Listens for requests from clients on its own thread, and sends responses.

Has a while loop that acts as a single state machine with the getRequest() function listening for events.

When getRequest() function receives data on a socket it sends the request to the serveRequest() function. If the data is in the form “SEND [path]” serveRequest() treats it as a “rdt\_send(data)” event, and constructs a Message, response, with the file described by path, which is like “packet=make\_pkt(data)”. Finally it calls response.sendMessage() which is like “udt\_send(data)”.

Request:

Contains information about clients’ requests.

Message:

Manages packetizing data and sending it.

Client’s Classes

Client: Creates the client processes which sends the message “hello server” in packets

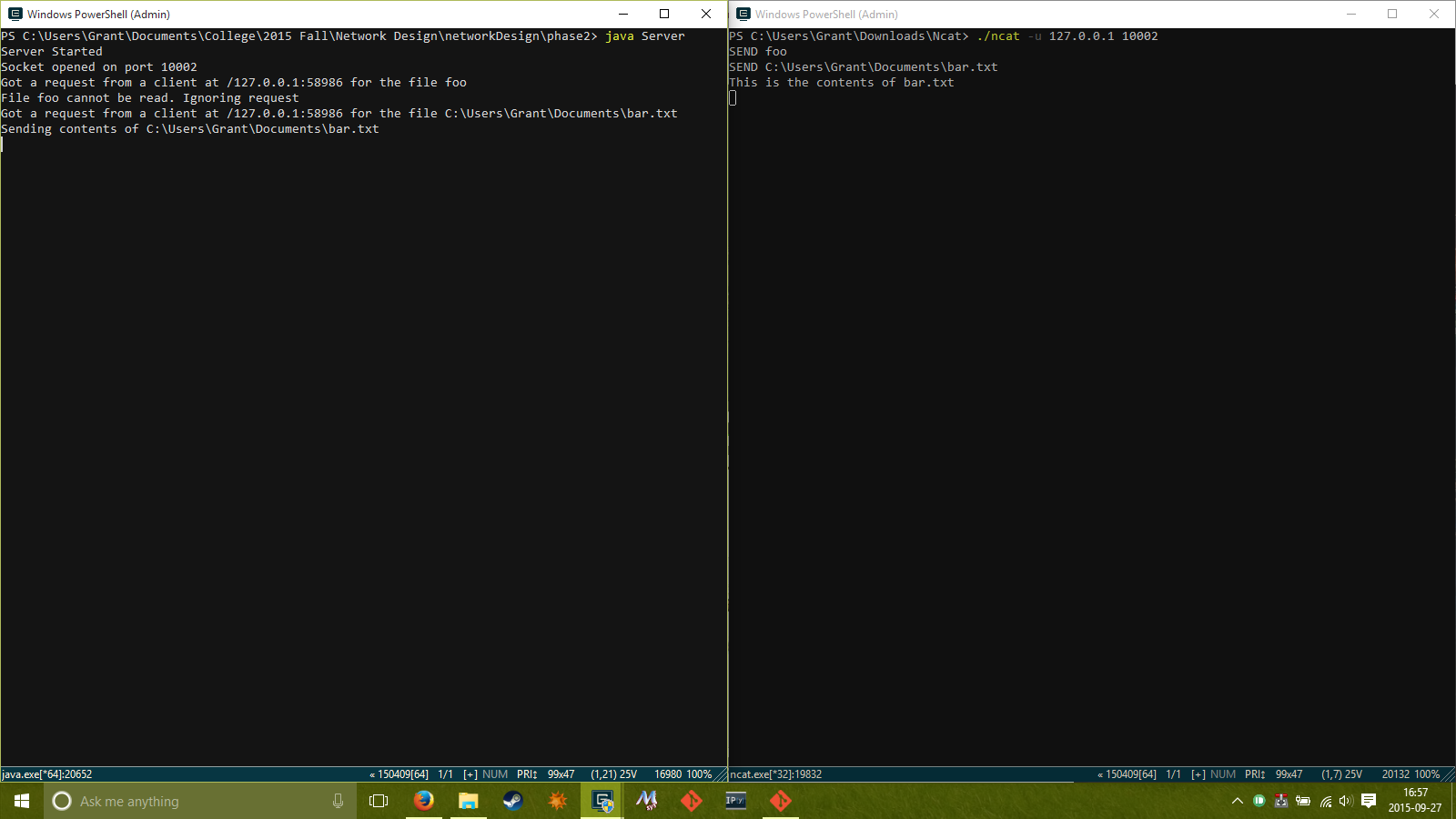
getResponse: Reads the response from the Server and formats it correctly according the Response.java formatting.

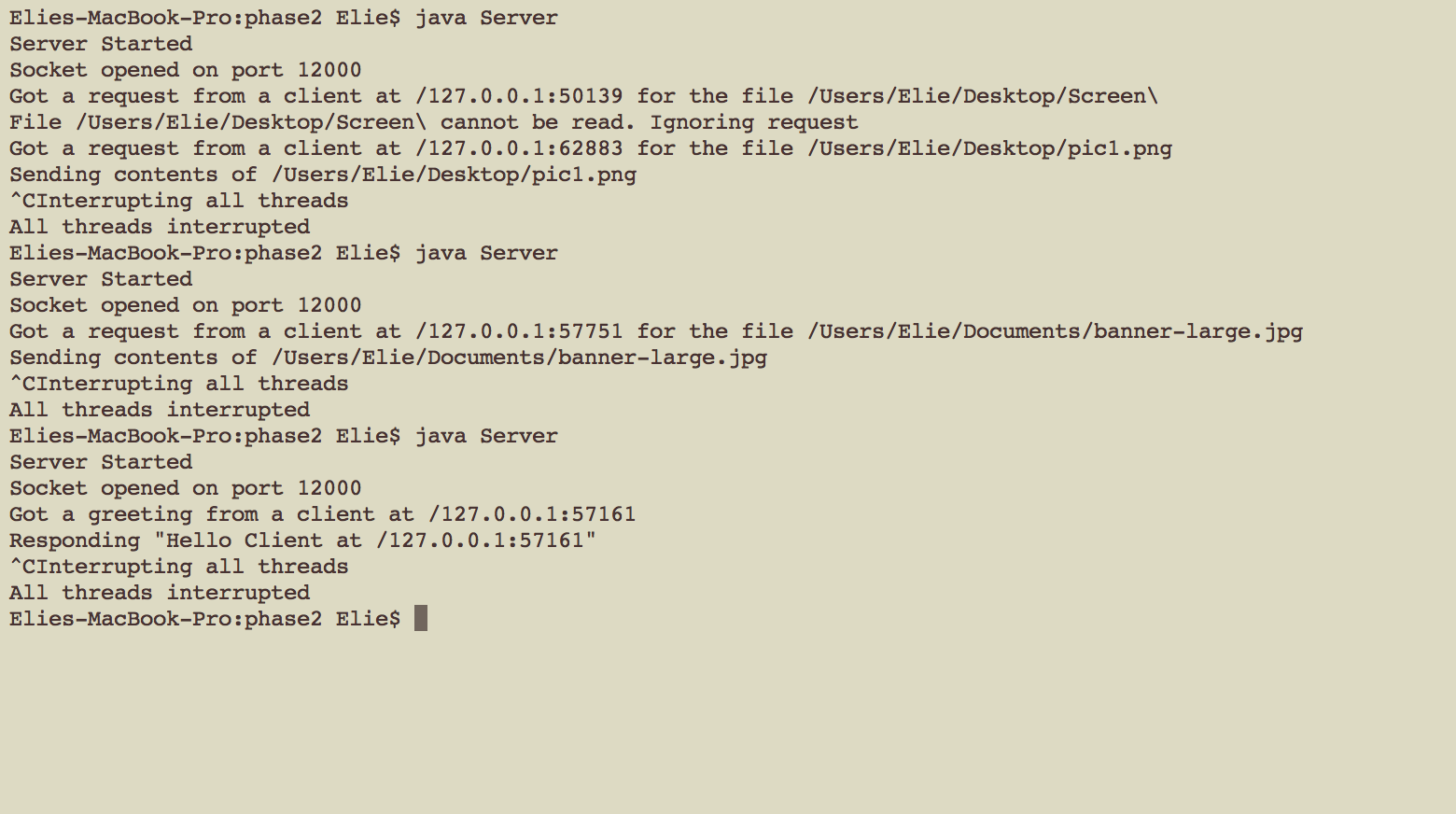
writeToFile: writes the reponse from the server to a specified path

Response.java contains classes that are used in both the server and client processes in terms of how to recive and manipulate the packets coming and going and what appropriate system outputs are there.

Working Case

**Server Process**



****

**Client Process**

