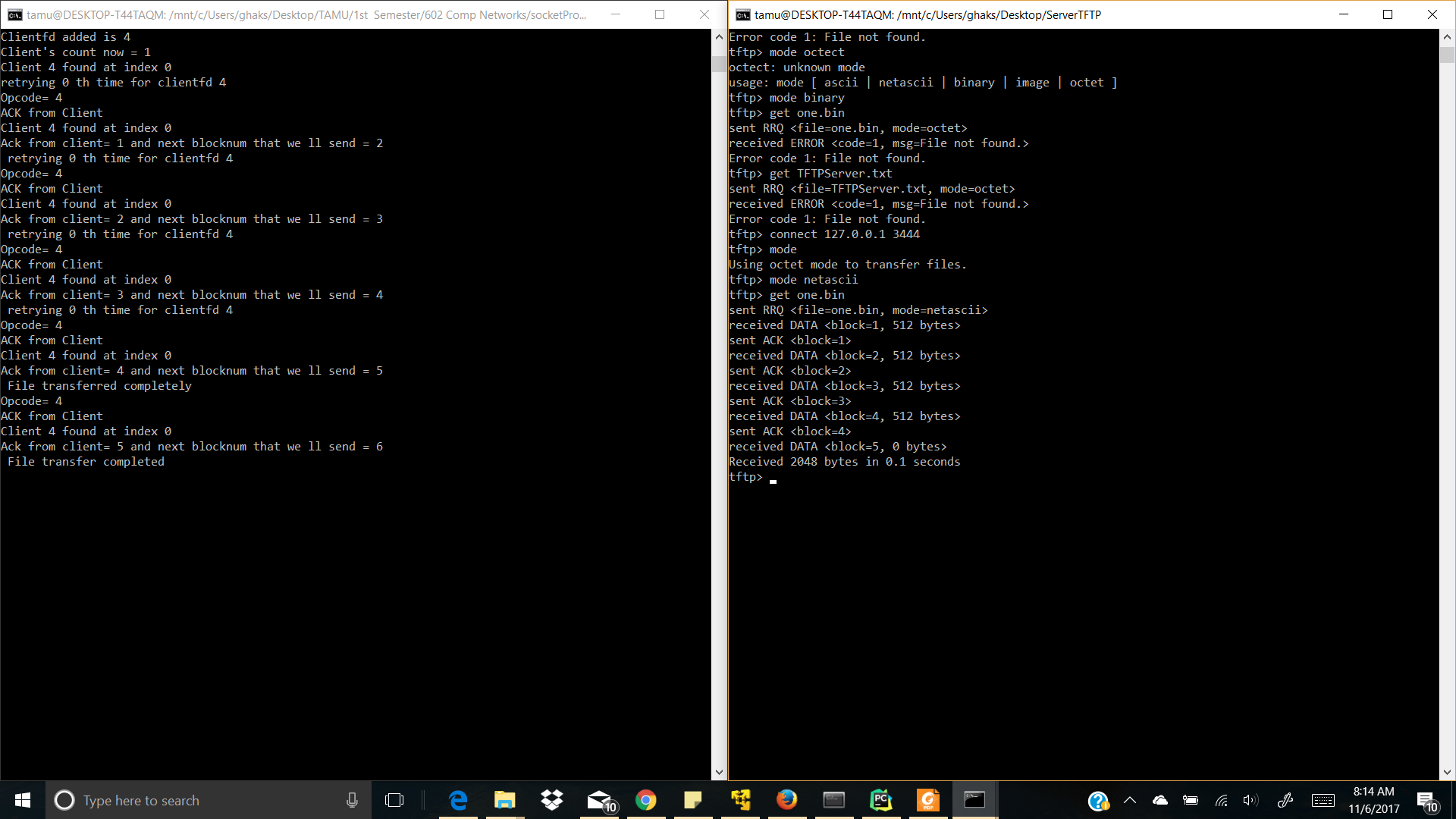
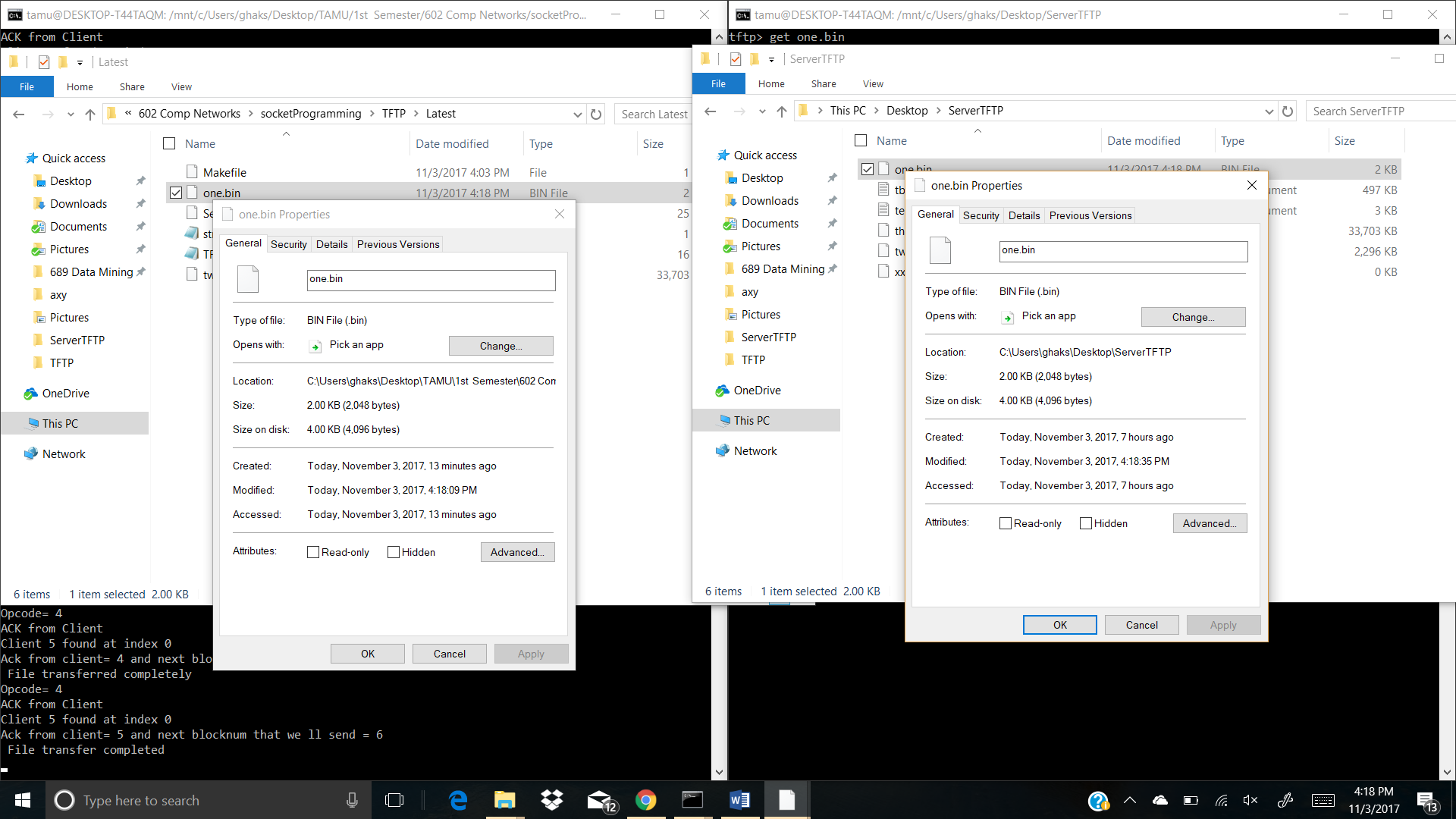
**Transfer of a binary file of 2048 bytes**

Binary Mode:

Netascii mode:



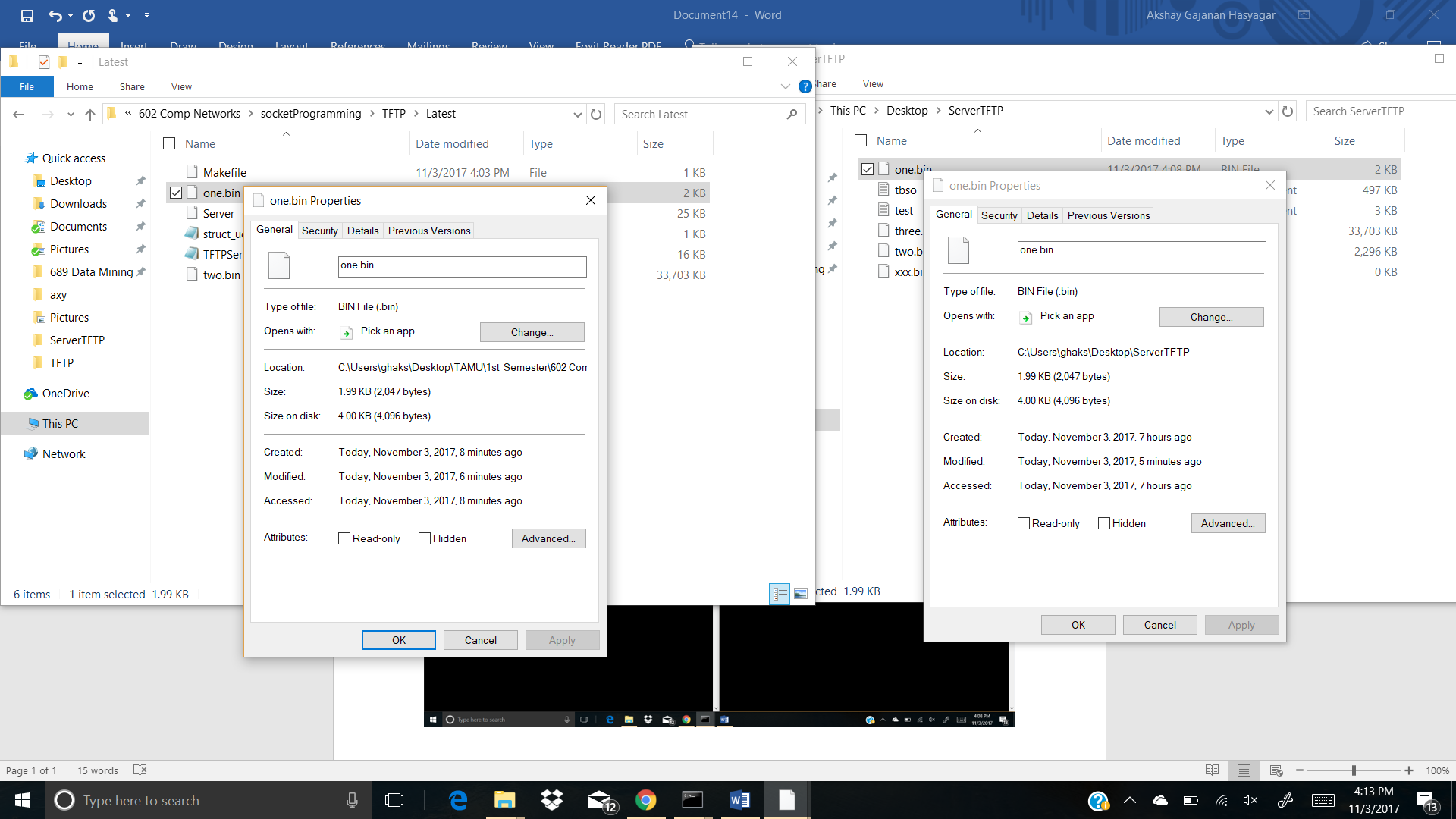
The source and destination file are of the same size



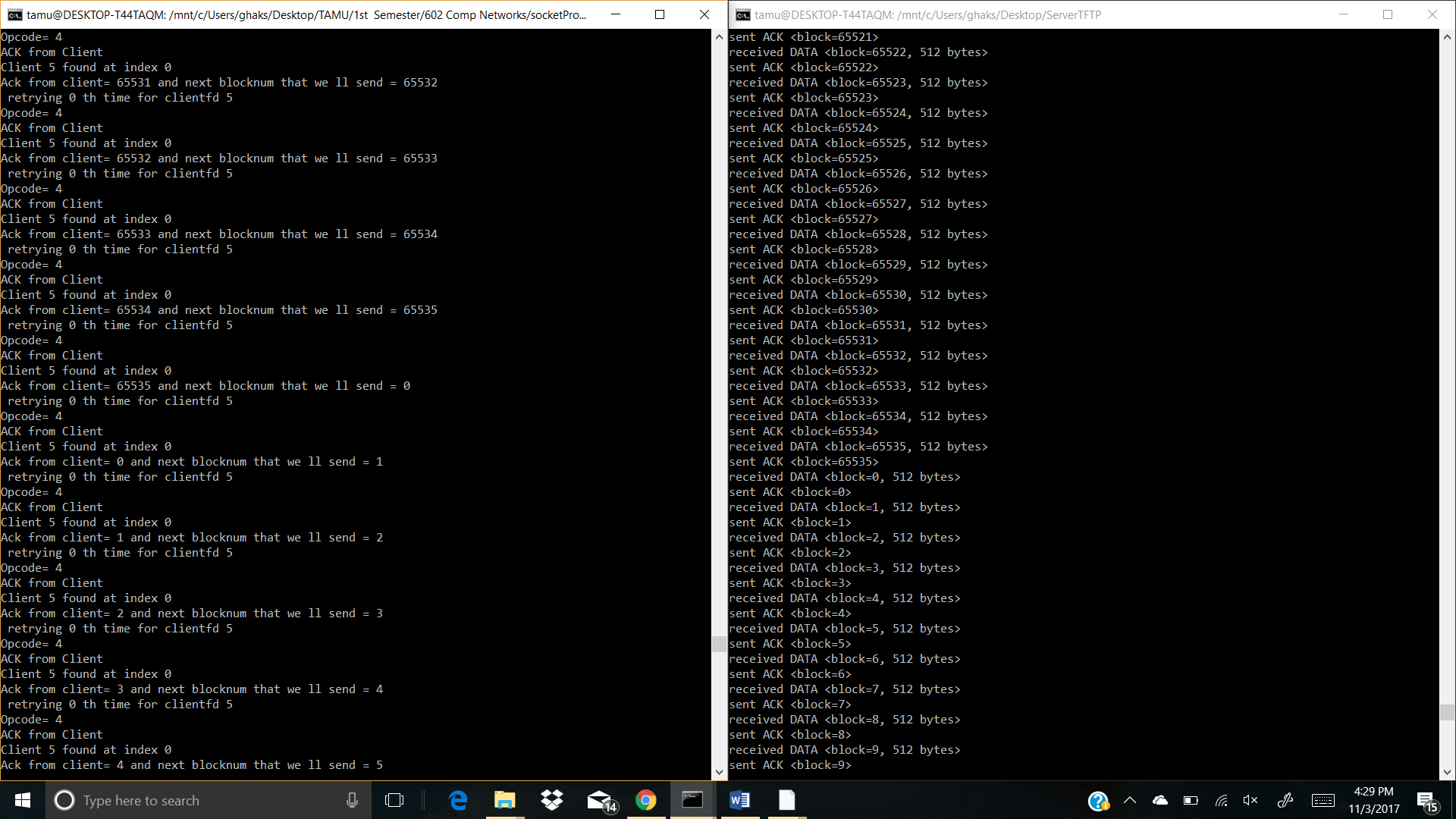
**Transfer of binary file of 2047 bytes**



The source and destination file are of the same size



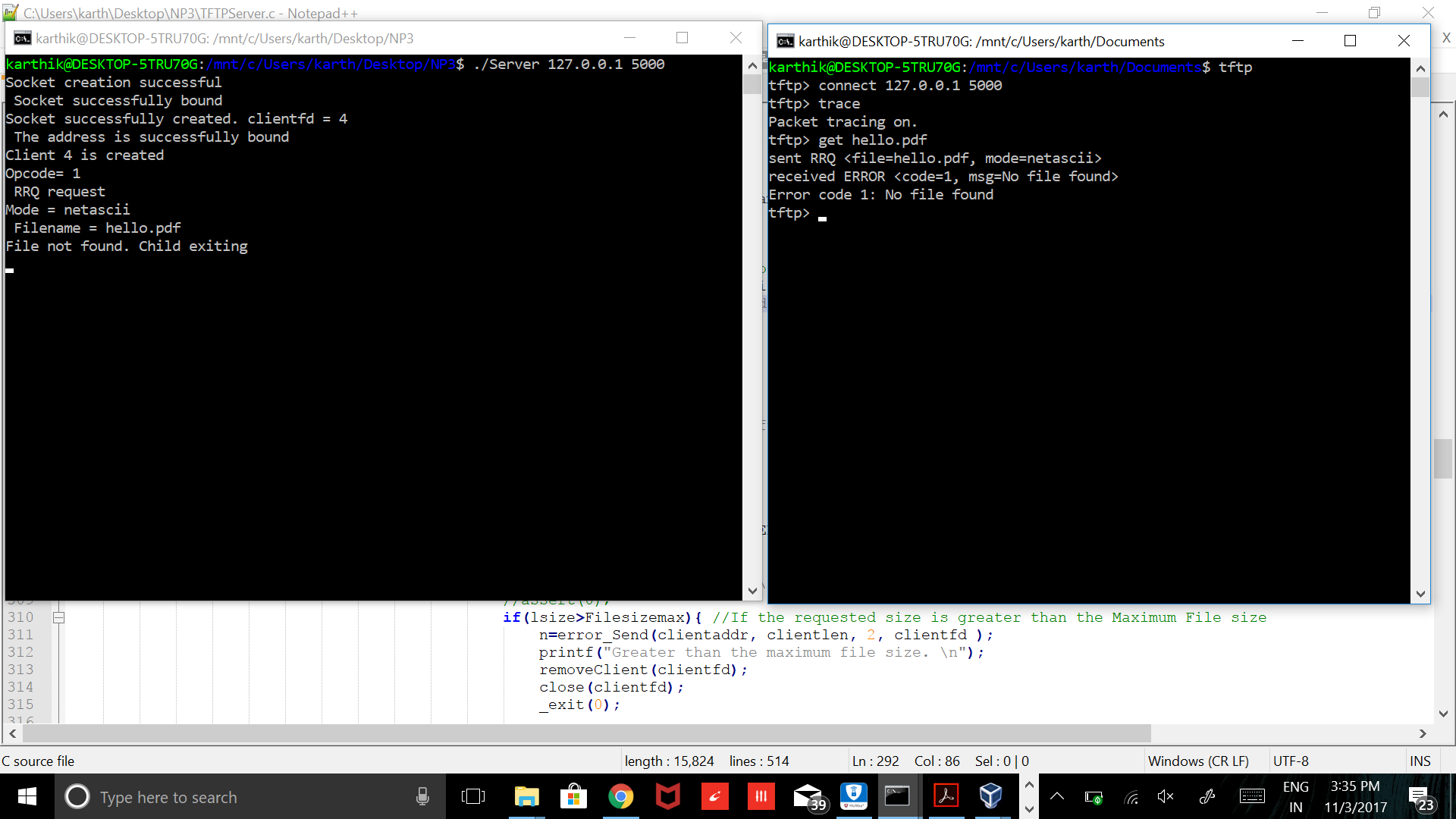
Transfer of binary file more than 34 MB. You can see below the block number rapping around .

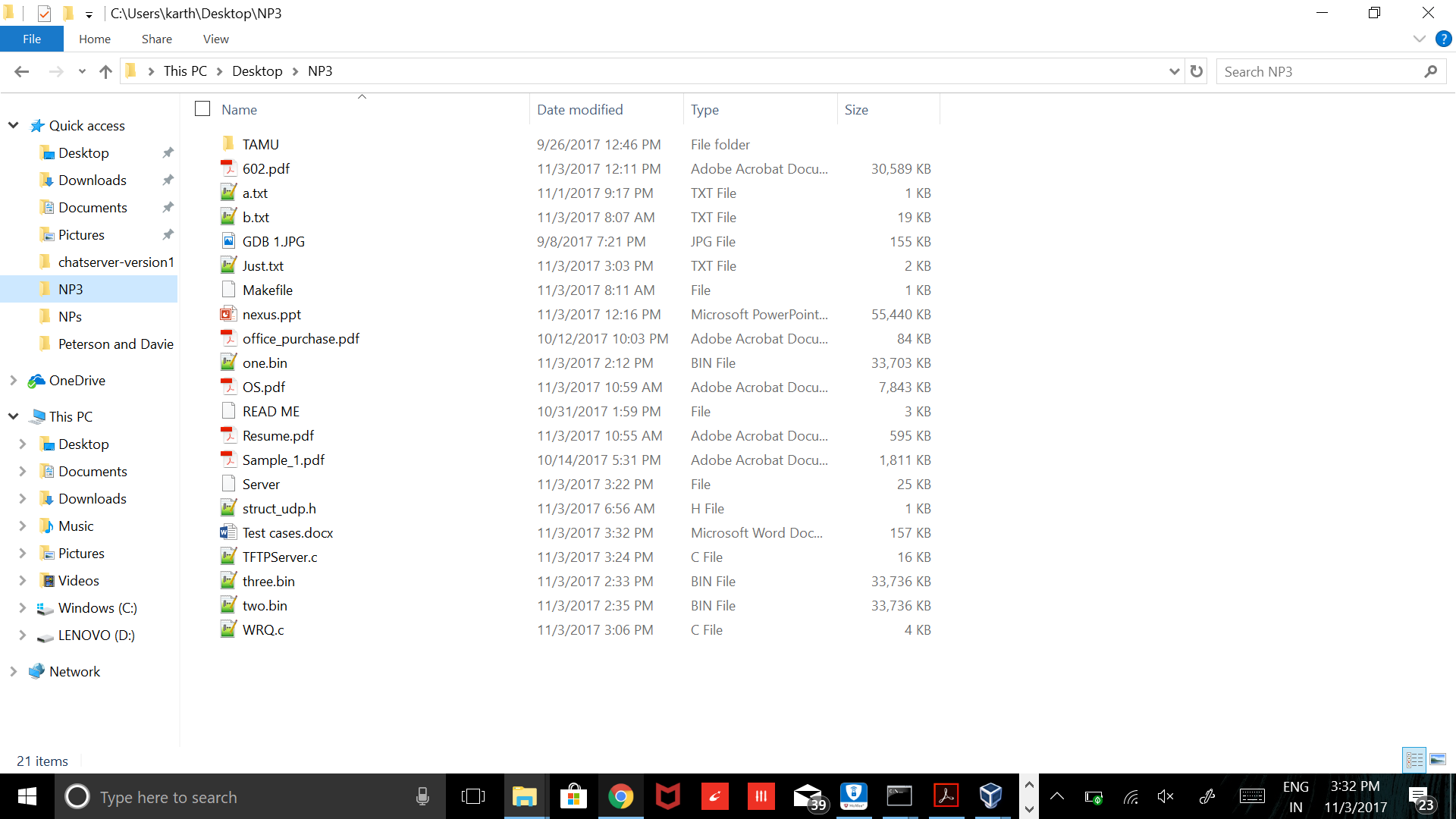


**check that you receive an error message if you try to transfer a file that does not exist and that your server cleans up and the child process exits,**

The client receives opcode 5 that indicates the message is an error message. Along with error opcode (5), it receives a message ‘No file found’.

On the server, removes the client from the client list, closes the socket and exits the child process.

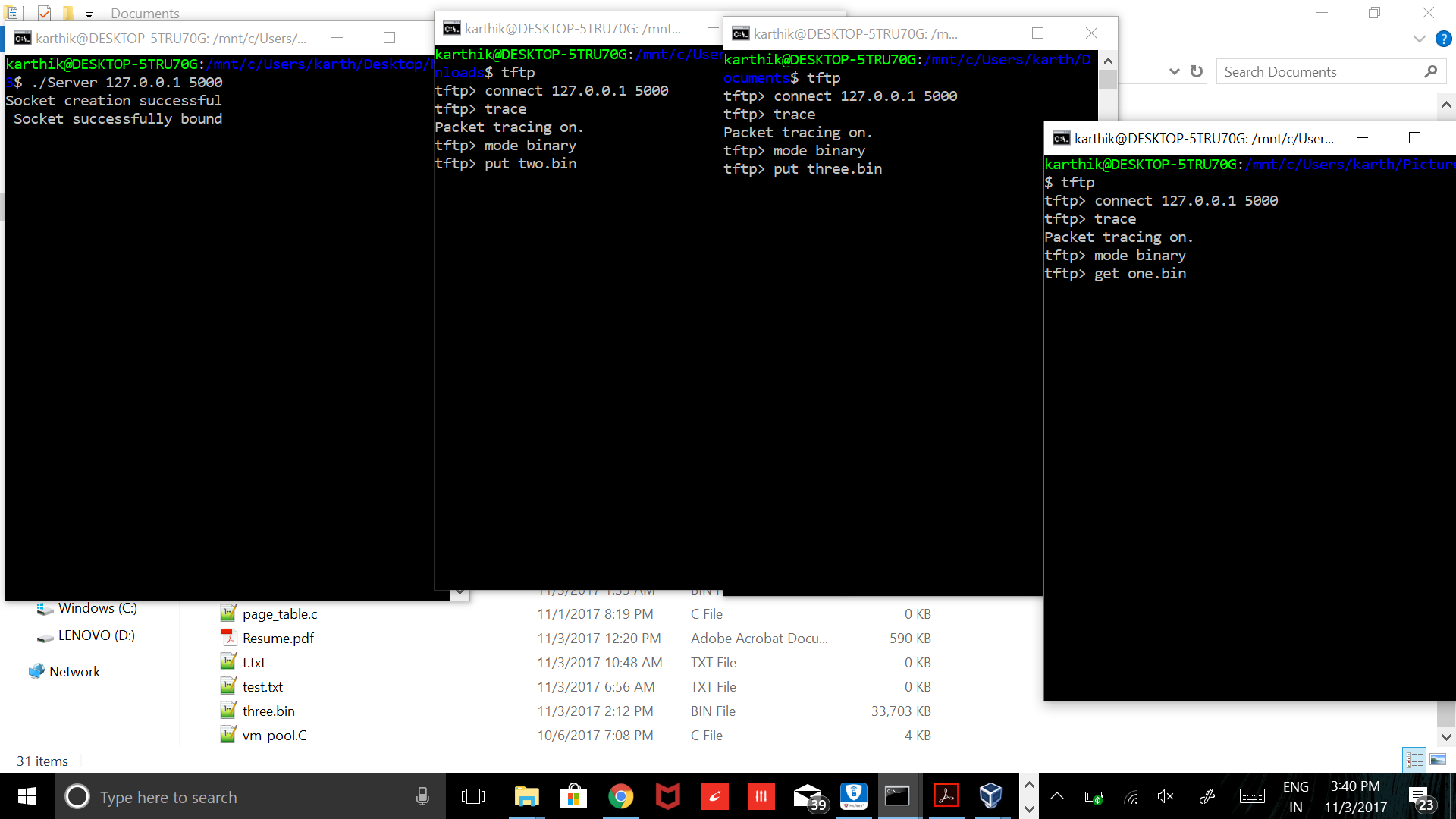




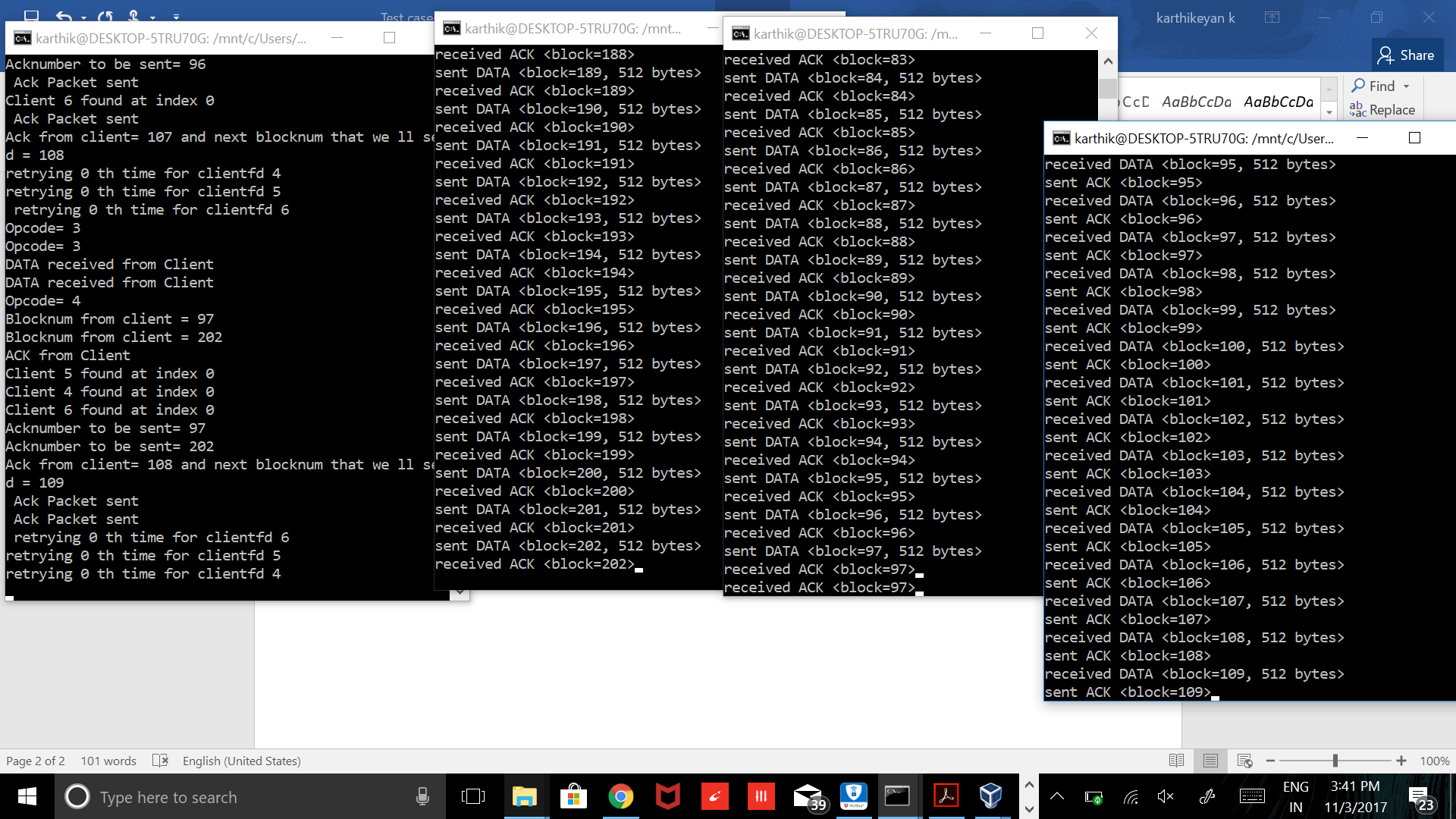
The home directory of my server

**Connect to the TFTP server with three clients simultaneously and test that the transfers work correctly (you will probably need a big file to have them all running at the same time).**

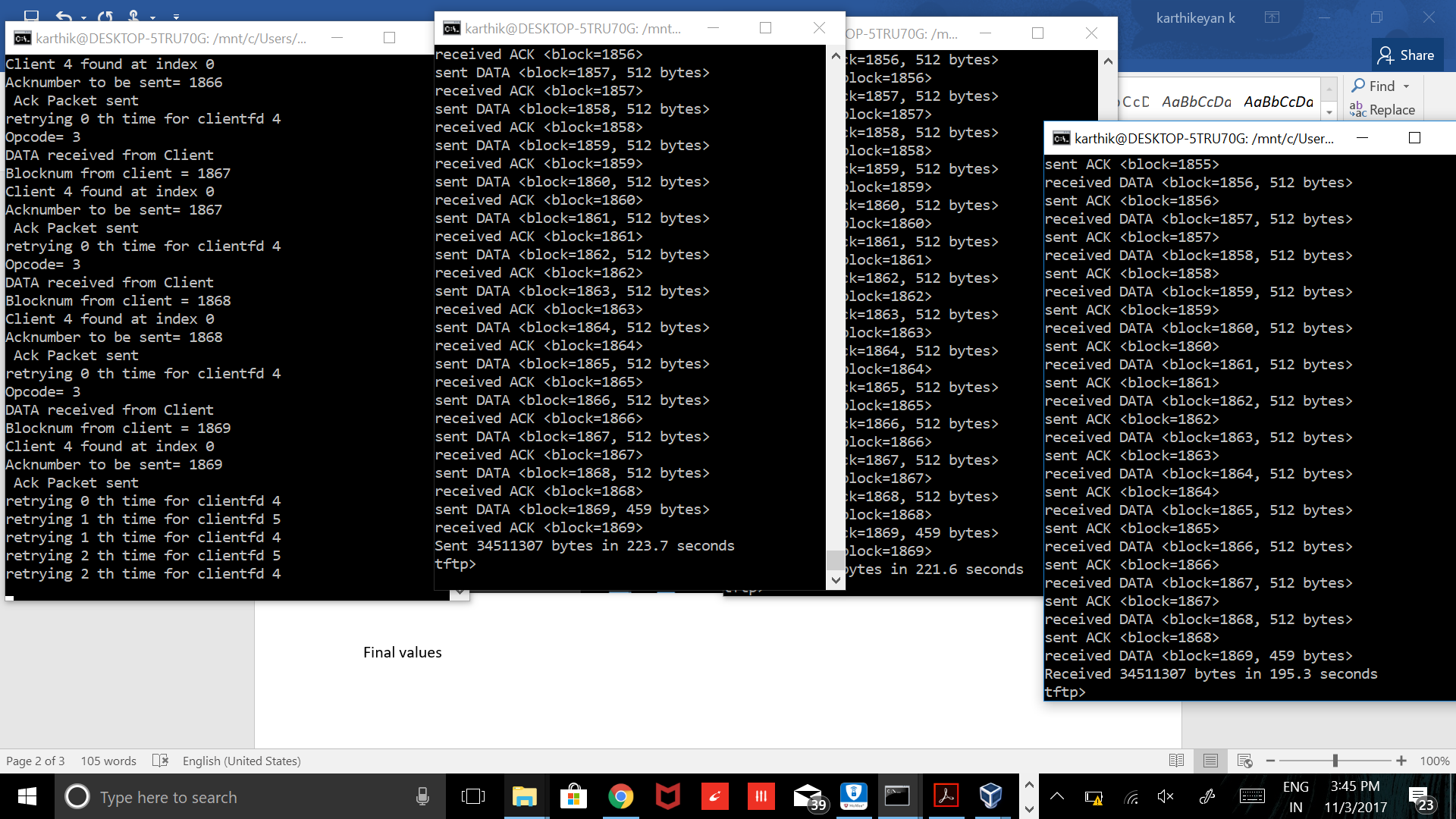
Here 32.9 MB file is used



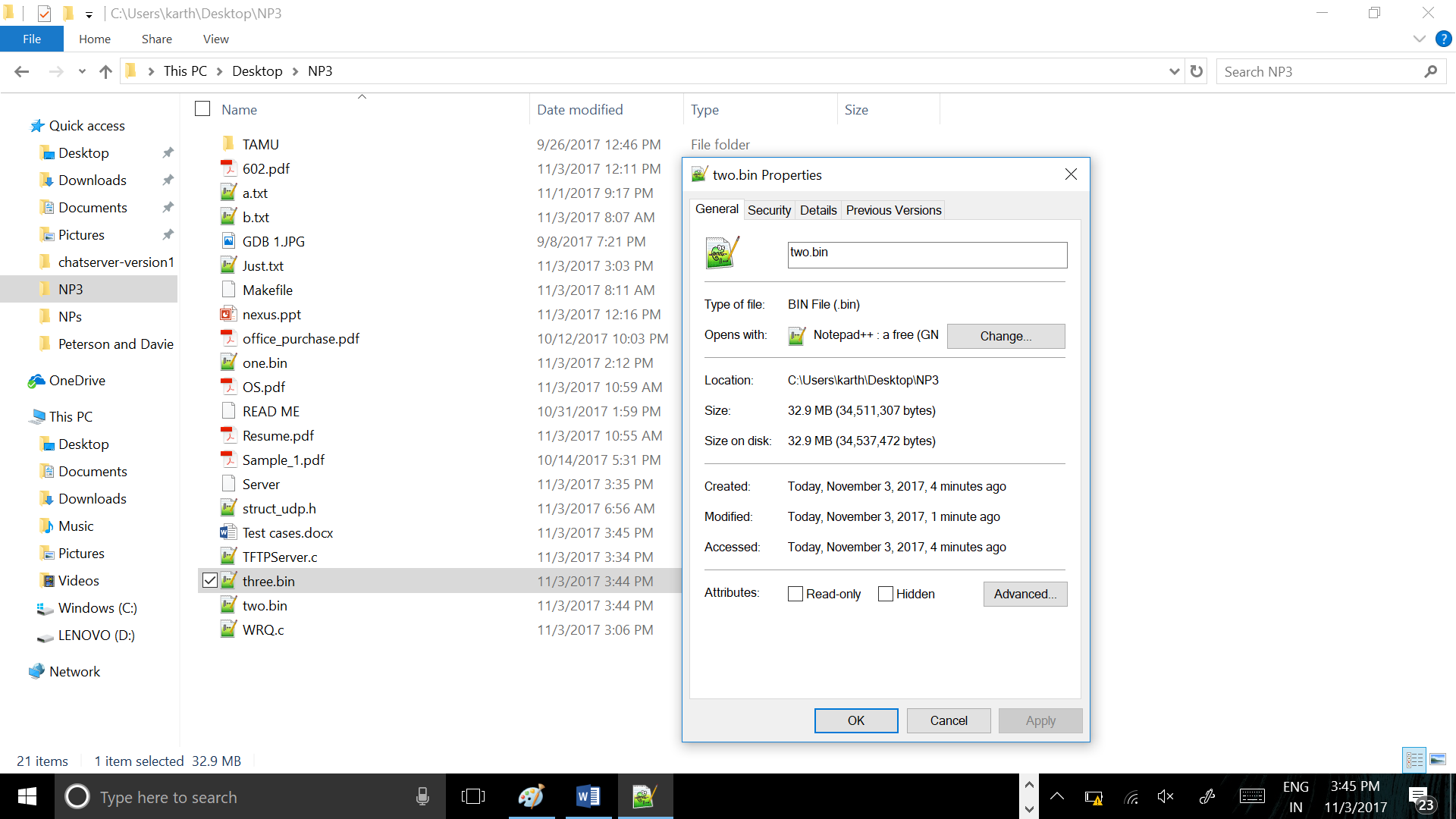
**Date exchange**



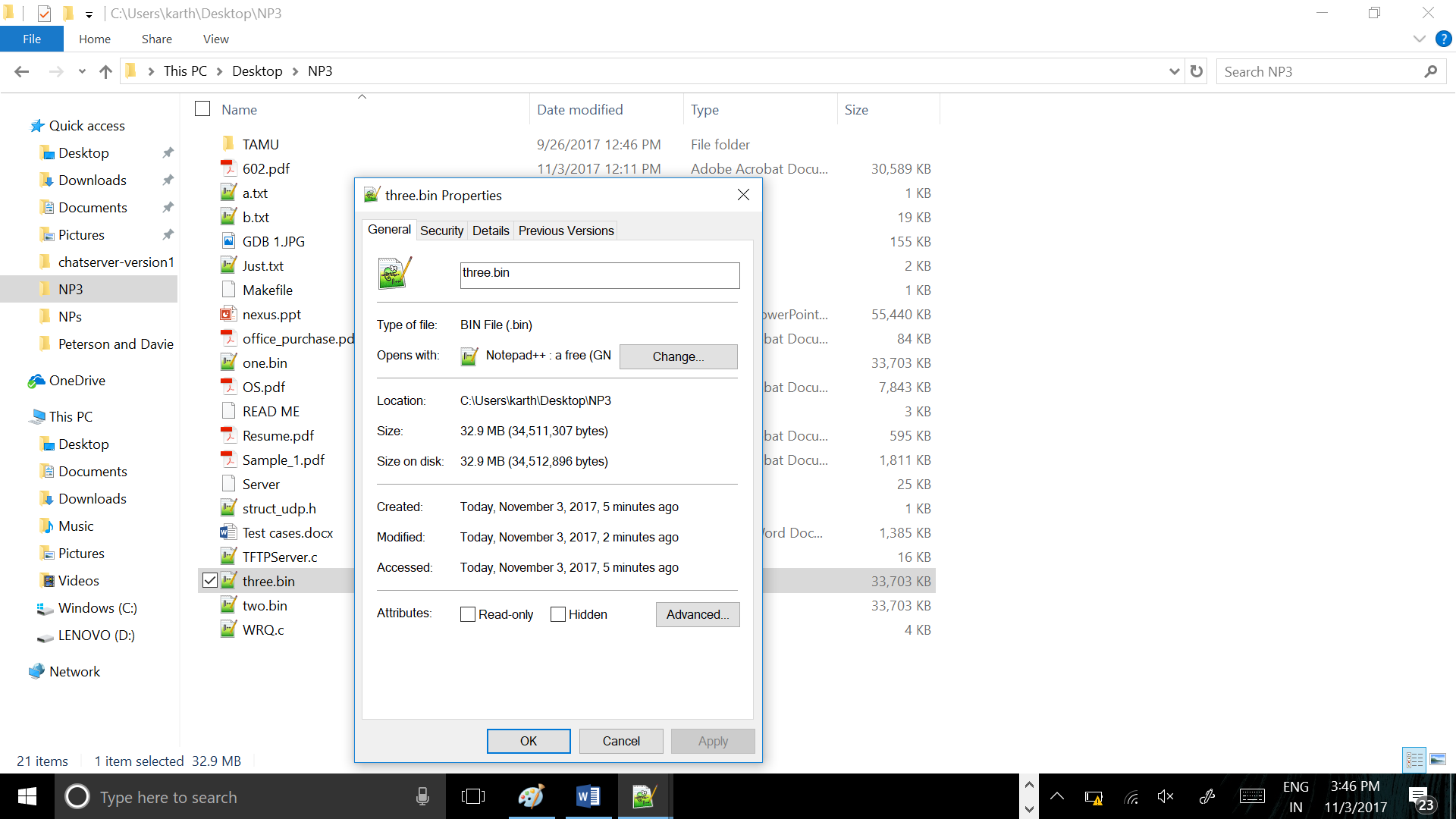
**Final values**



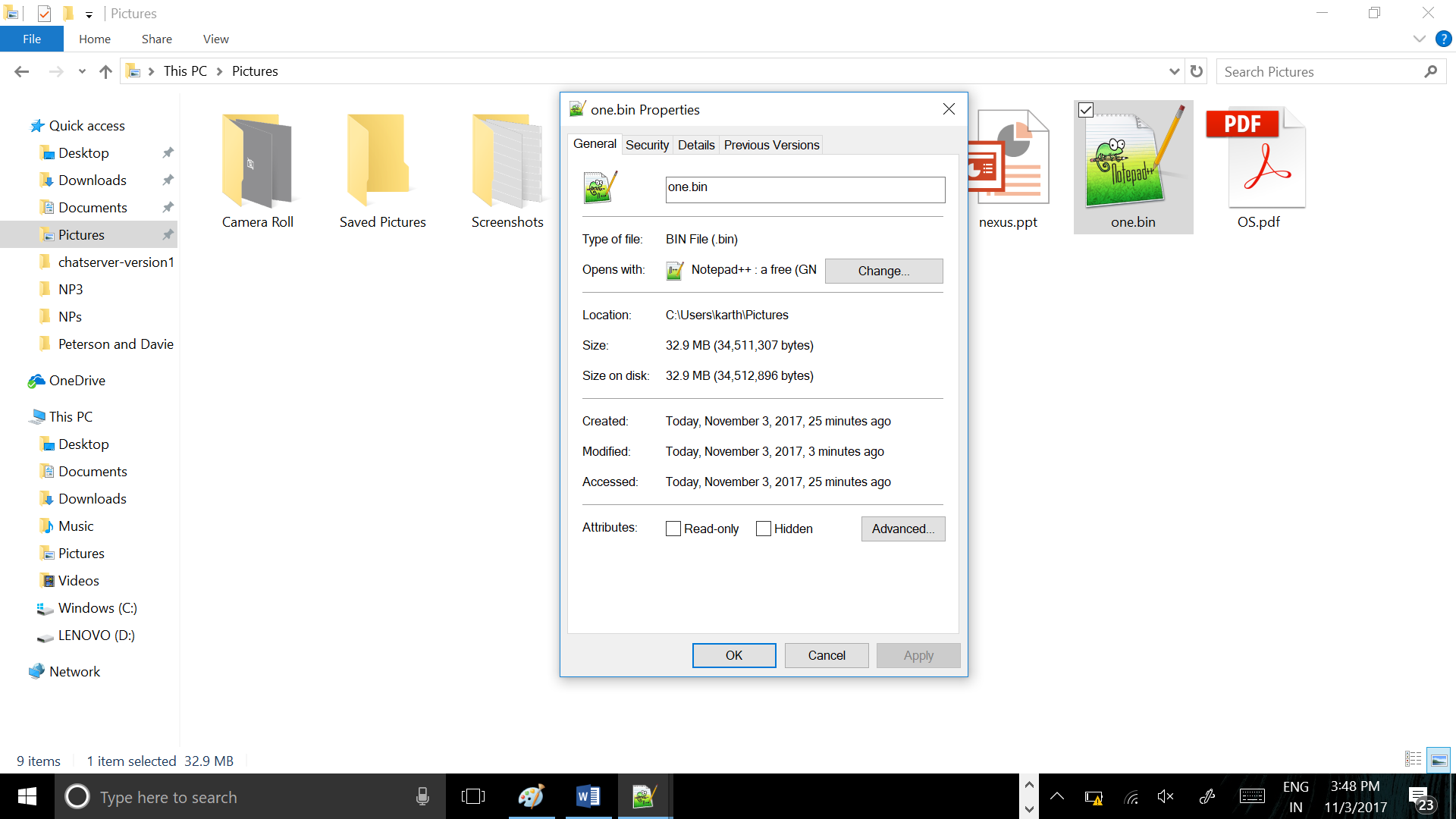
Two.bin - file write to server directory : Desktop/NP3



Three.bin – file write to to server directory : Desktop/NP3



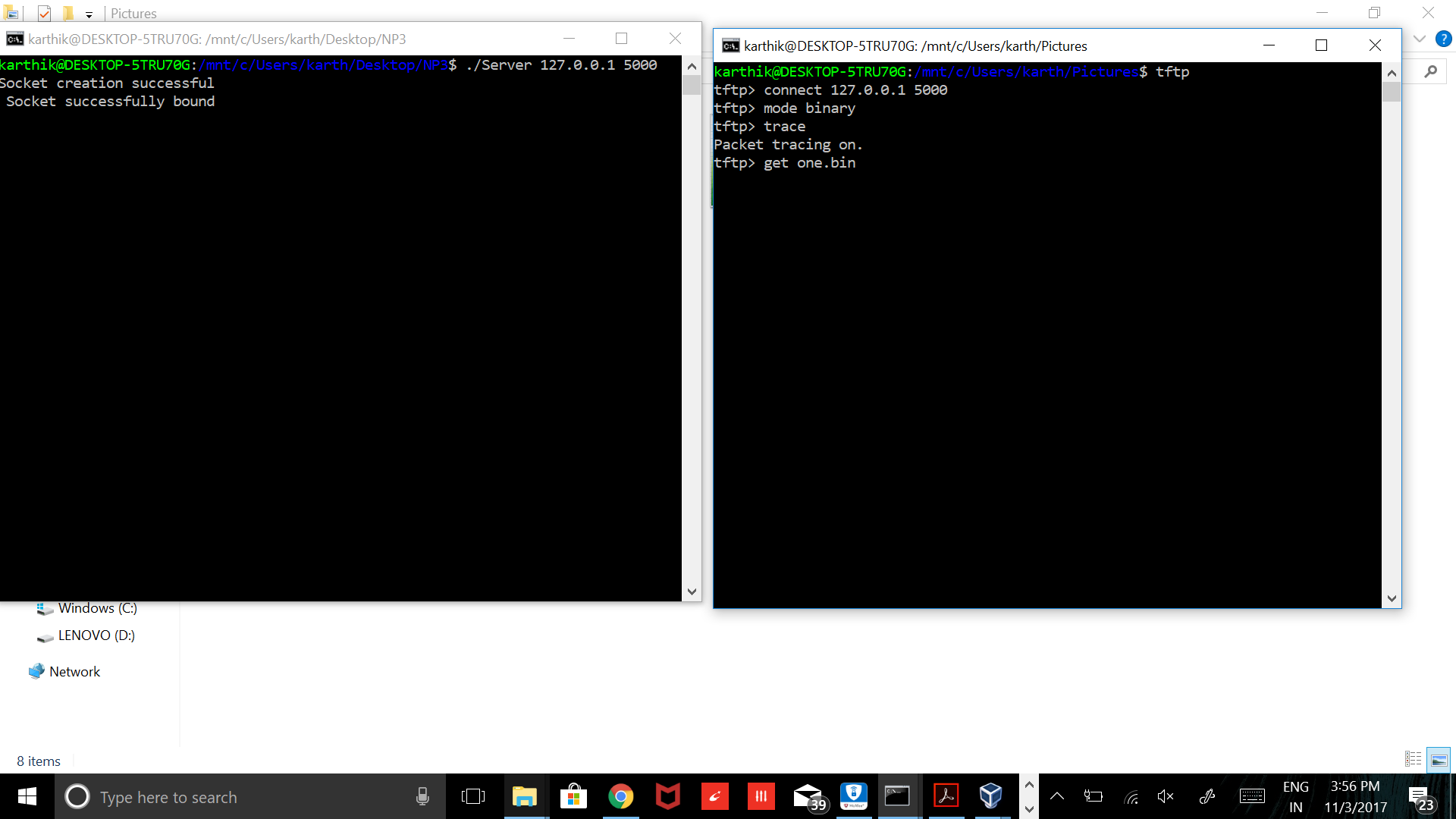
One.bin – file read to client directory : Pictures/

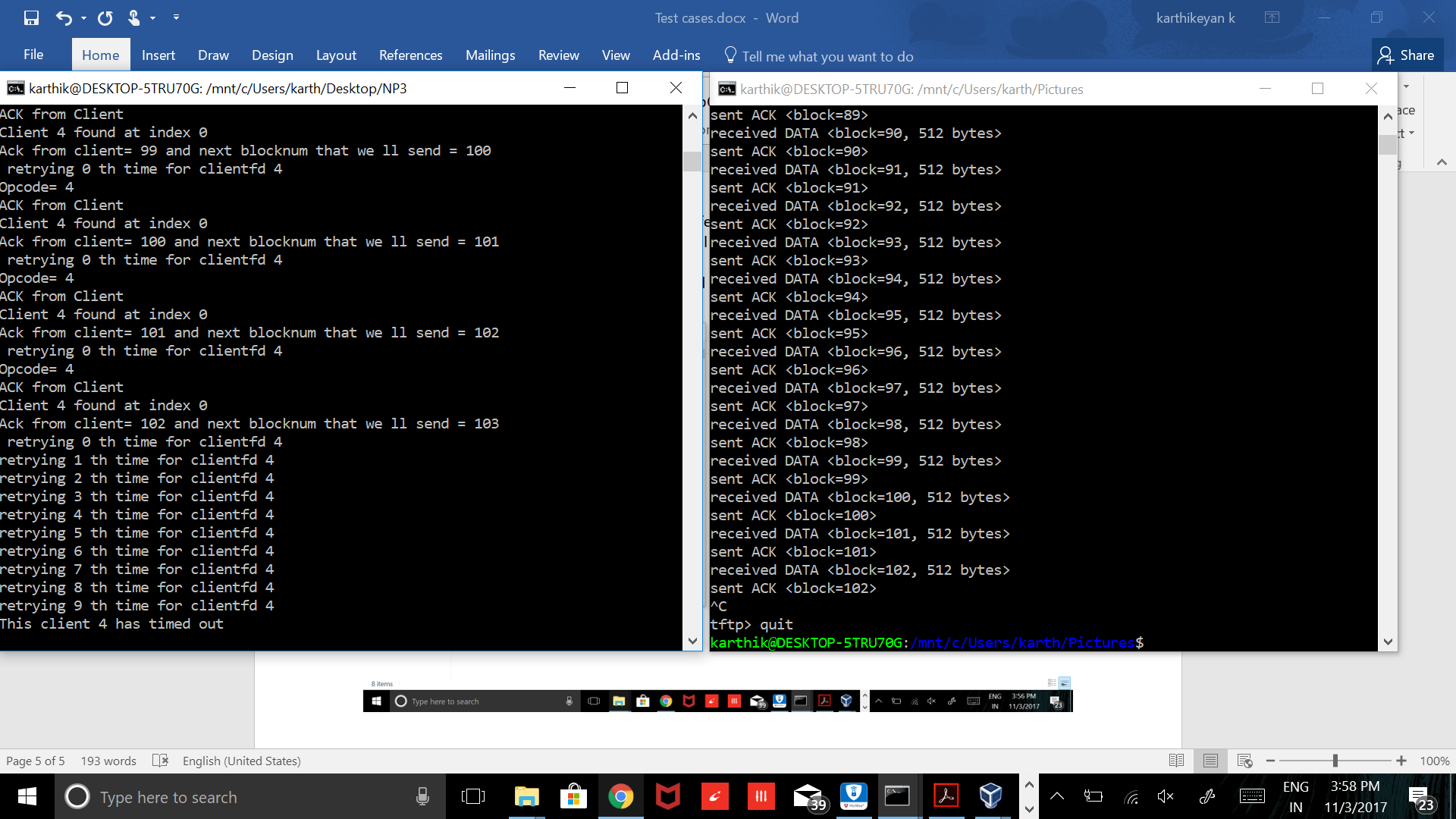


**Terminate the TFTP client in the middle of a transfer and see if your TFTP server recognizes after 10 timeouts that the client is no longer there (you will need a big file).**

10 seconds is the timeout value that we have used and 10 attempts before it declares the client as dead.

After that it removes the client from the client list, closes the socket and exits the child process.

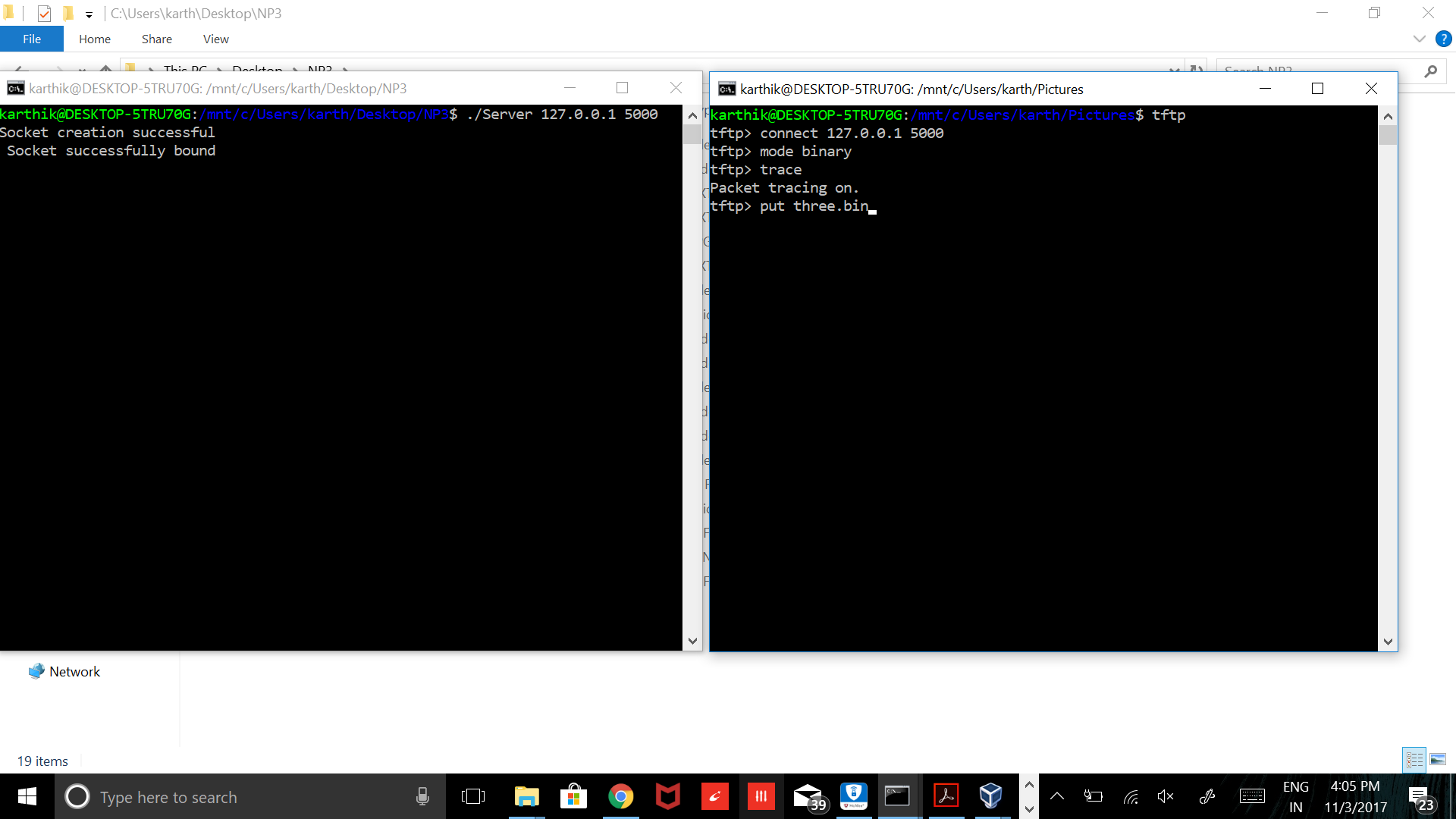




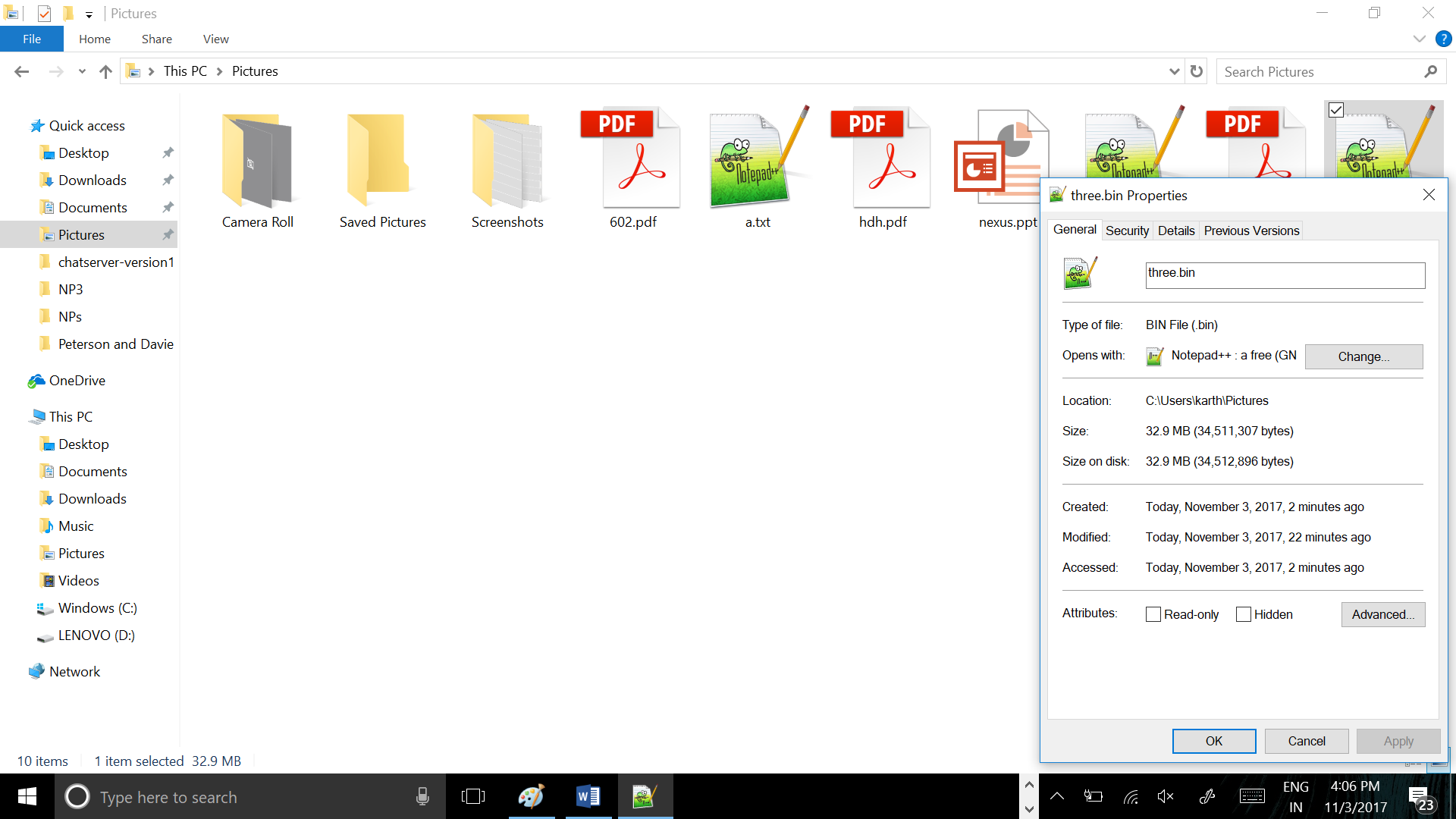
**For a Bonus, you may implement WRQ support in your TFTP server for an additional 30 points. If you implement the Bonus Feature, test that you can send both netascii and binary files from your client to the server, and then transfer back the same files from the server to the client. Compare the original and the copied versions to make sure they are identical.**

NOTE: PLEASE FOLLOW THE TIMESTAMPS TO GET A BETTER IDEA OF THE EVENTS TIMELINE

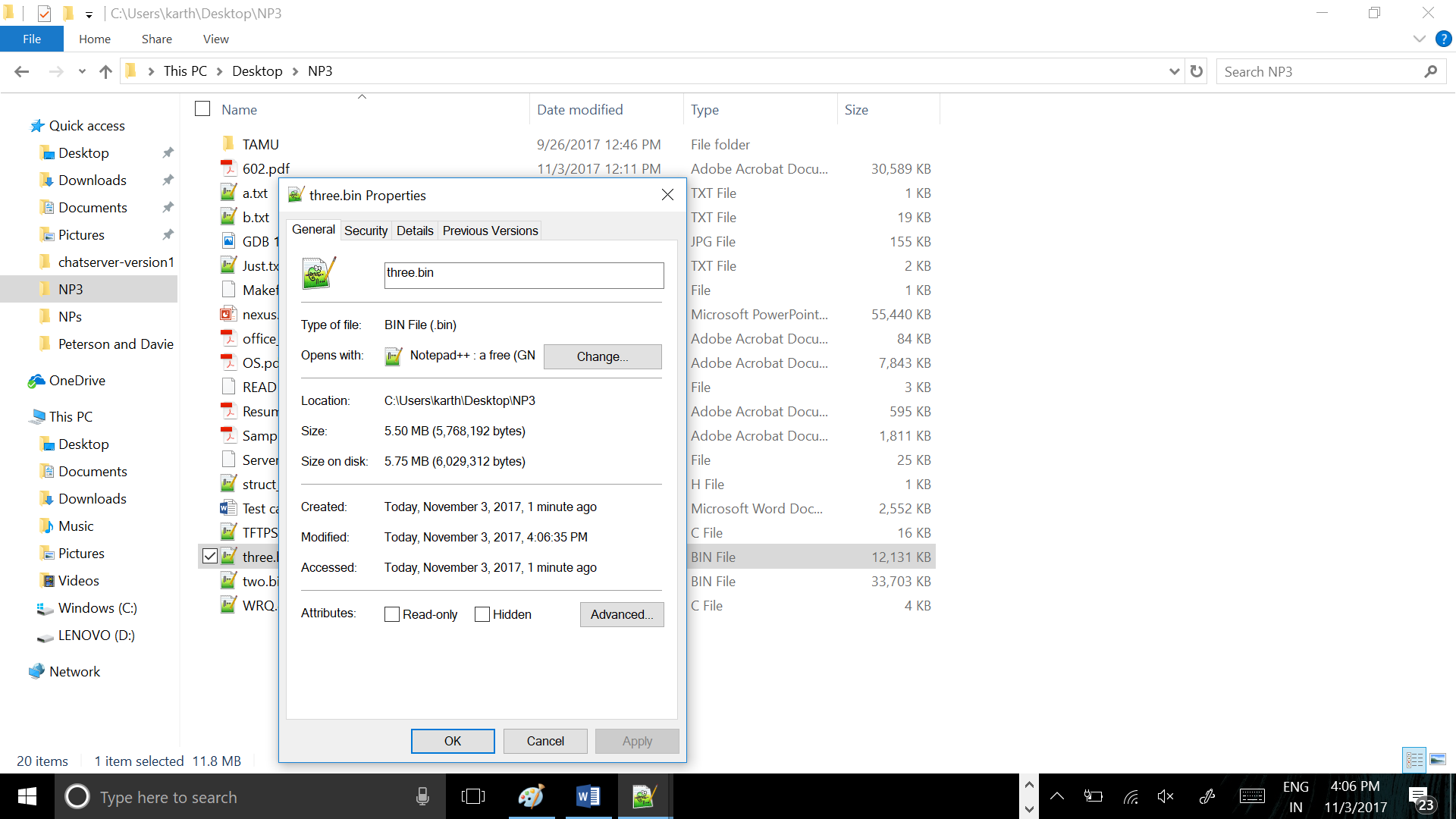
Initial

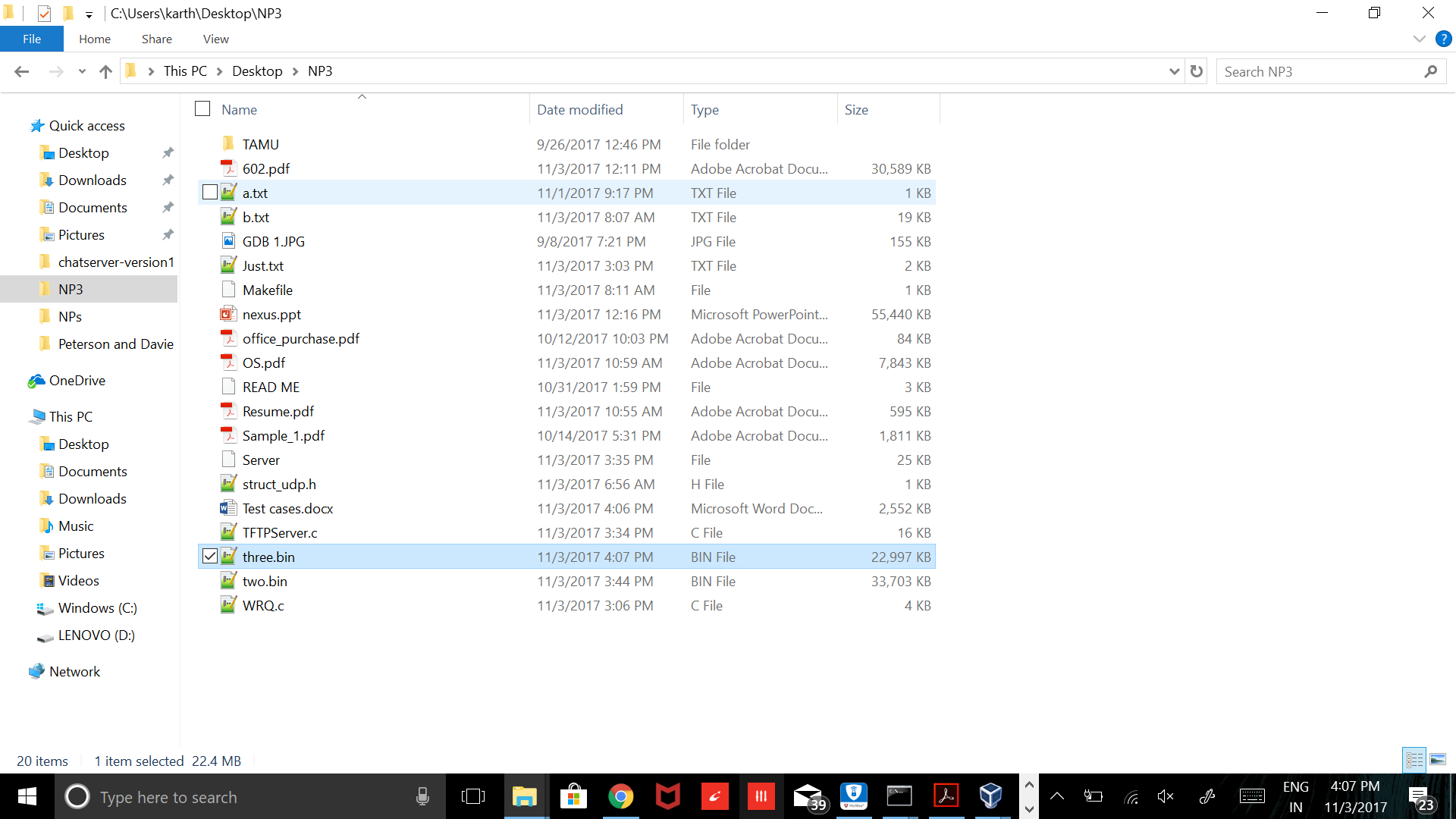


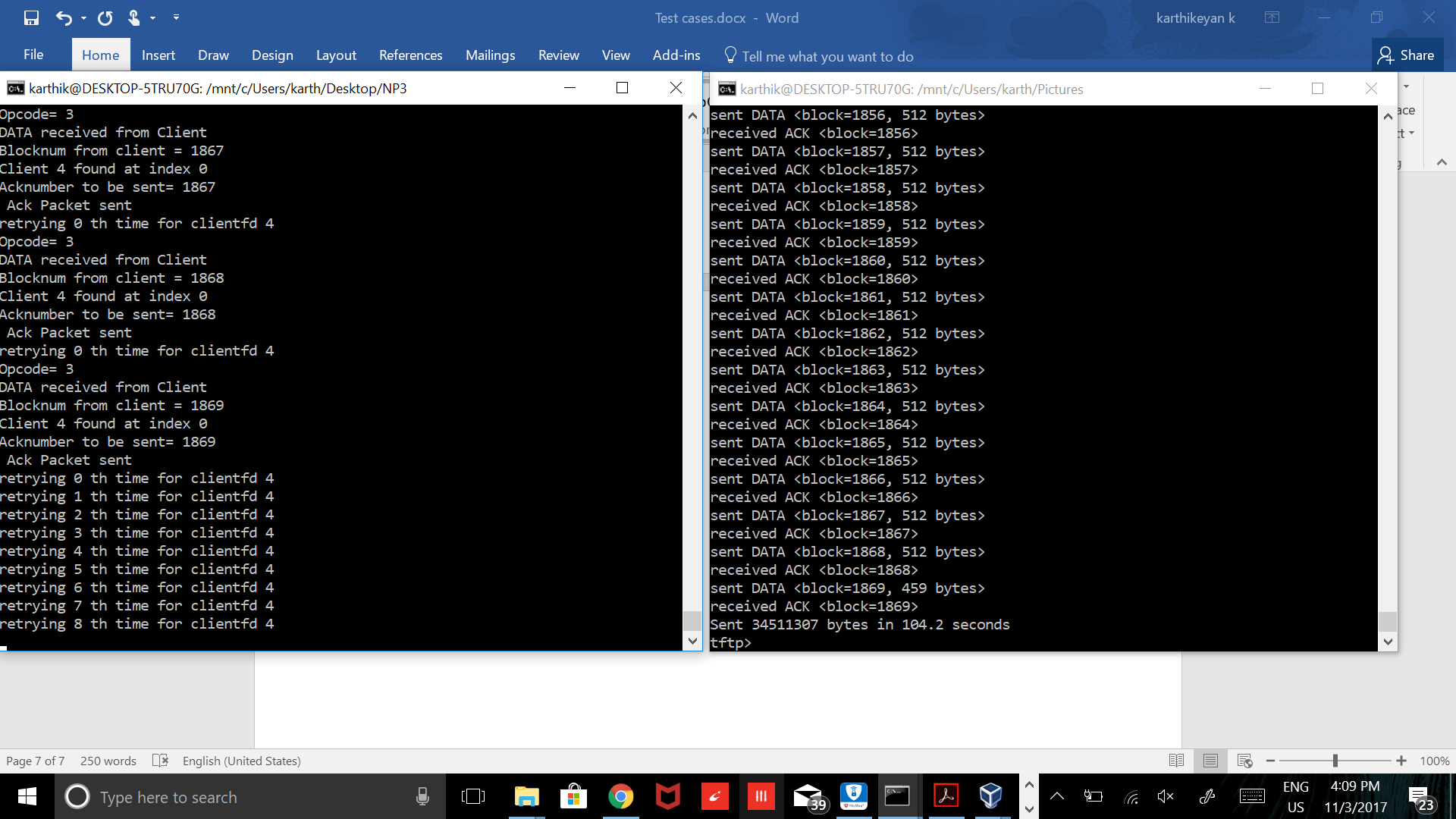
Input file size from /Pictures (client)



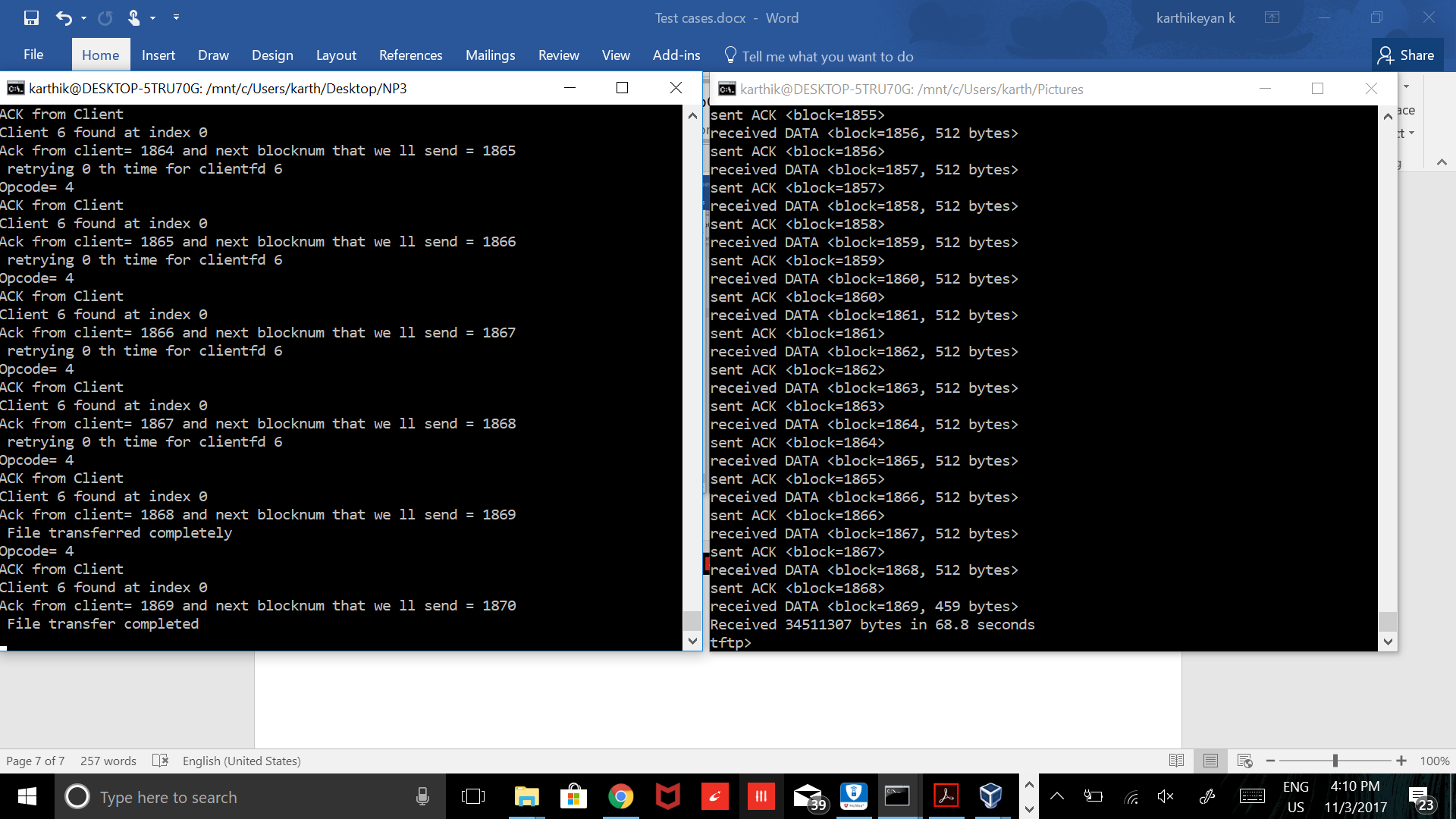
Server directory as it the transfer happens. (See the file size increase)

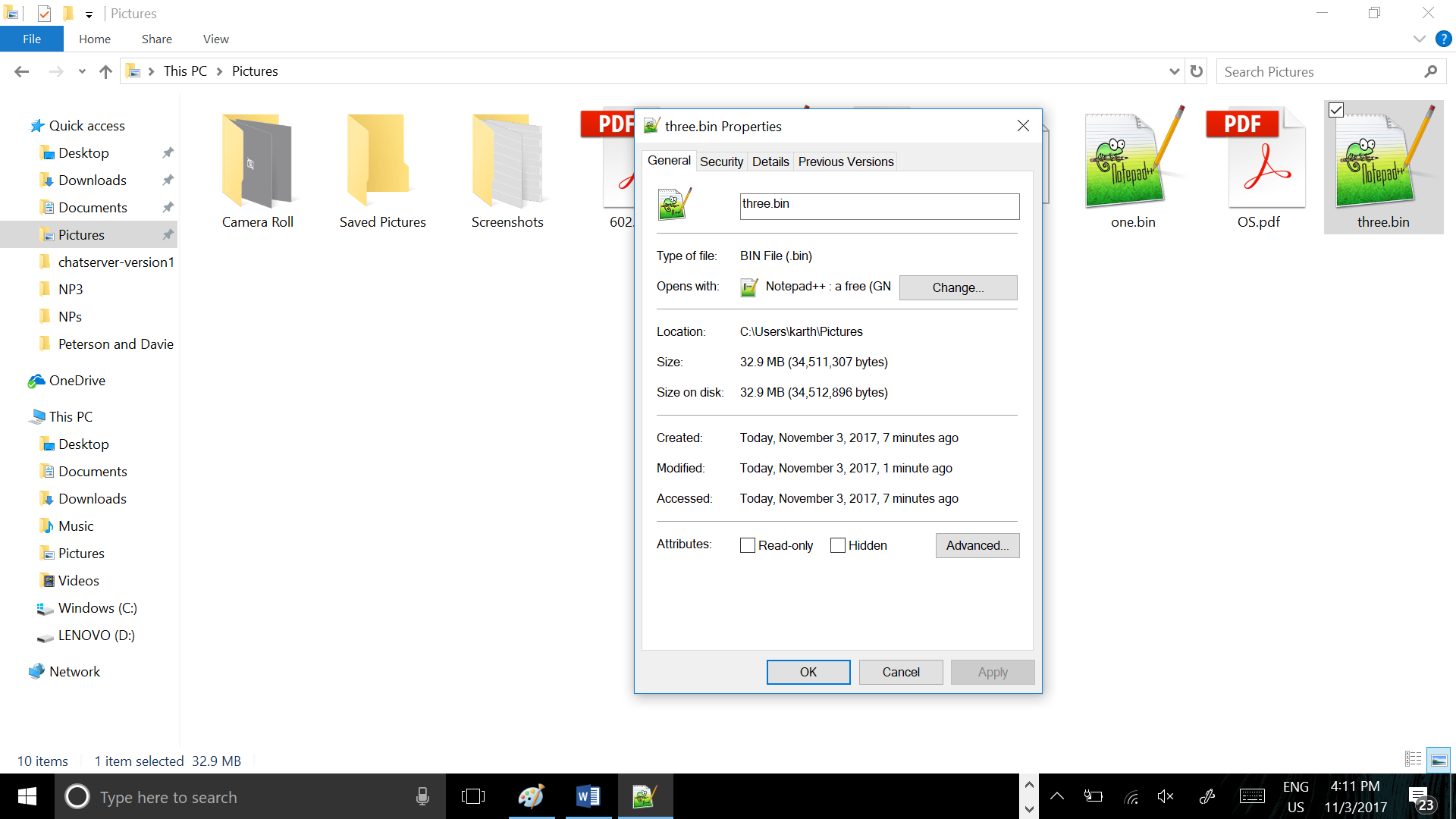






After this putting it back to Pictures



Check the modified timestamp and the latest size

**Bonus Mode**

Get file , Put File and then get file again. The size remains the same.

