

Network Programming Assignment #2
ECEN 602, Fall 2016
Due Oct. 14, 2016 NLT 5:00 pm

Programming Assignment (100 points)

In this assignment you will implement a Trivial File Transfer Protocol (TFTP) server. The protocol specification is IETF RFC 1350. You do not have to implement Write Requests (WRQ); you will just be reading from the server. The server should allow retrieval of files in the local directory from where the server is executed. You can use the native tftp client to test your server implementation.

TFTP is implemented over UDP, which is a connectionless protocol, which means there are very few guarantees of the transmission medium. You will have to resolve these issues with the protocol's retransmission mechanisms. TFTP is basically a Stop-and-Wait protocol similar to what we discussed in class.

Notes

1. Just as in the previous assignment, it will be helpful for you to break the project up into parts and implement and test each independently.
2. You will need to pay specific attention to how TFTP manages ports, a request and the subsequent file delivery are treated very differently.
3. You will need to use the same select loop architecture as in the previous project to handle concurrent client access.
4. Make sure you understand what happens to a transfer at the boundary of the maximum transferable file size.
5. Test you server with many different size files. Think about what sized file transfers could cause issue with the stated transfer mechanism, or someone's implementation of that mechanism. Think about the maximum file size that the server can handle.
6. TFTP uses port number 69 by default. But, you will not be able to bind the server socket to this port. Therefore, any other port number (ephemeral port) can be used by both the server and client.
7. Consider situations where concurrent data transfers happen and how the server handles such transfers.

Submission Guidelines

1. The programming assignment should be uploaded on google drive folder before 5:00 pm on the date that it is due. The Google drive folder has been shared to each team. A "NetworkPgm2" folder will be added shortly.

2. Programming assignment submission must include: makefile, readme.txt, and the server code.
3. The readme.txt should contain a summary of your code: architecture, usage, and errata, etc.
4. Make sure all binaries and object code have been cleaned from your project before you submit.
5. Your project must compile on a standard Linux development system. Your code will be graded on a Linux testbed.
6. Should you have any doubts, please ask questions in the recitation session or the TA's office hours.
7. All submissions will go through Stanford Moss to detect plagiarism. Don't copy code from somewhere else.