

CPSC 457 Assignment 2 Report

By Adam Berlak, ID: 30008230

How to compile and configure:

1. Make sure the txt files are in the same directory as the s.c program
2. Compile the client with the command: "gcc c.c -o c -lm"
3. Compile the server with the command: "gcc s.c -o s -lm"
4. To run the server write ./s
5. To run the client write ./c

How to use the program:

The program consists of 3 commands. Get, filenames, and requests for txt commands. The filenames command will retrieve the names of all files in the server's directory. The get command with prompt you to enter a filename, and entering a filename with the suffix txt will retrieve the file from the server.

Justification for implementation:

My implementation is TCP based, where each octoleg and request for data is accompanied by a sequence number and acknowledgment. After my client requests a leg, it will start with sequence number 0 and stores an acknowledgment with the sequence number the client expects from the server. If the sequence number of the data retrieved does not match it will retransmit the request for data. If no data is received at all in the timeout span of five seconds, the client will also retransmit the data again. Until the client receives the appropriate data it will not progress, and keep requesting the same data. The server stores the files based on octoblocks in an array and sends them to the client piece by piece. The client requests and gets the octolegs and octoblocks sequentially, if an octoleg is missing it will not continue to the next octoblock, or leg until it has the information required. The method of transport for all data transfers is UDP. With regards to scalability, my design can retrieve all filesizes up to and including 256 KB. If the filesize is divisible by 8, each segment of the file is put in a 1111KB sized octoleg and sent to the client. If the filesize is not divisible by 8, i.e. with the file 739KB.txt, the filesize is divided by 8, the floor of that is sent as octolegs. The remaining data is sent 1 byte per octoleg in an additional octoblock. For files larger than 8888KB, the amount of full octoblocks required is calculated and sent to the client, the last block is treated the same as 739KB if it is not divisible by 8, allowing my program to also handle files larger than 8888KB that are not divisible by 8. The retrieved file will be saved as downloaded.txt in the same directory as the client file 'c.c'.

Testing:

In order to test my program I used a linux virtual machine from my home computer. In order to test retransition I programmed the client to discard the third request for data if $I = 3$. The server won't get the request and won't send the data. After the client waits for 5 seconds on recv, it will timeout and resend the same request, but this time I will be 4 and the request will go through. I have included this in my code for demonstration purposes. This is why Seq=3 is sent twice.