

This project main purpose was to help us understand socket programming and threads. Our program contains two files, which are libnetfiles.c and netfilesserver.c. The libnetfiles.c holds all the client request such as netopen, netread, netclose, netwrite, and lastly netserverinit. When a client makes a request to each one of the net functions, the client makes to establish a connection to the server side, which will handle all the operations. The server would then bind a connection from the client, and then create a thread for each client request and pass it into our function pointer handleRequestCall methods, which takes care of each client request such as NetOpen, NetRead, NetWrite, and lastly NetClose. To store our file descriptor from each client request, we used a linked list. Our Linked List struct holds the client data. The client data is another struct which contains all the metadata about the client request.

Functions

NetServerInit

- This function is primarily used to check whether a given hostname exists and associate a file with a proper mode. The macros we defined in our libnetfiles.h are
 - UNRESTRICTED 0
 - EXCLUSIVE 1
 - TRANSACTION
- This function must be the first function called before the client makes any other request like NetOpen, NetRead, NetClose, and NetWrite.

NetOpen

- This function takes a pathname and the mode for the file such as O_RDONLY, O_RDWR, and lastly O_WRONLY. If the given pathname does not exist, then netopen will return -1. Else NetOpen will return a negative of the file descriptor back to the user. Within this method we check whether or not any file descriptor within our linked list has more restrictive permission than the client request being inserted into our linked list through the isValid function. This is primarily used for Extension A.

NetRead

- This function takes in a file descriptor, a buffer, and the number of bytes to read from. This function will perform as regular read does, it will return -1, if it was unable to read from the function and send the errno back to the client otherwise it will return a valid file descriptor.

NetWrite

- This function like NetRead takes in a file descriptor, a buffer, and the number of bytes to write to. This function will perform as regular write does, it will return -1, if it was unable to write into our buffer and send the errno back to the client, otherwise it will return a valid file descriptor

NetClose

- This function will be remove the file descriptor from our file linked list.