

Christopher LoPresti

Roslan Arslanouk

CS214

Asst 3: Making a Better Open

**** completed extensions – base, A,C,D ****

**** the makefile provide compiles netfilesserver and libnetfiles ****

This program is designed to simulate a client/server connection, where the client attempts to work with file descriptors on a remote server. The client (your code) should be compiled using our libnetfiles library. In order to run your code you must first run the server on an ilab machine (./netfilesserver), then run your client code on different ilab machine. Your client must first make a call to `netserverinit()`, with two arguments; a host name, and a file mode. A file mode is either `unrestricted/0`, `exclusive/1`, `transaction/2`. A description for each file mode can be found in the assignment description.

Once a `netserverinit()` call is made, the client will either establish an IP and port for the server, or it will set `h_errno / errno` to `HOST_NOT_FOUND`. If the file mode is incorrect the `h_errno / errno` will be set to `-300`, our predefined error for `INVALID_FILE_MODE`.

Once `netserverinit()` has successfully found the server and stored its IP and port, you can go ahead and make your `netopen()`, `netread()`, `netwrite()` and `netclose()` calls. They work exactly like the regular versions of `open()`, `read()`, `write()`, `close()`.

Depending on the file mode you use when making your `netserverinit()` call, you may find that when you try to open a file it might take some time. Your request will be placed in a queue and it will wait until your request is valid given your file mode. If we find that after anytime over 2 seconds your request is not valid, we will time out your connection, send a connection time out error back to the client, and your program will initiate your next net call.