**Q & A**

1. What happens if the file being requested is not inside the chroot? Does it matter?

* After Changing the root directory, the child has to read file from the root
* If it does not find the file there, the program will not throw an error but pass empty message to the client.

2. Explain the mechanism you are using to pass file descriptors between parent and child processes.

* While file descriptors are actually just integers, they are mapped by the kernel (in a per-process manner) to kernel-internal data structures that describe the details of the opened ‘file’.
* The goal of file descriptor passing is to create a new file descriptor (probably with some other integer value) in some other (possibly unrelated) process that is mapped to the same kernel-internal data structure as the original descriptor in the sending process.
* The parent process opens a descriptor, and forks - the child process inherits all descriptors by the parent and if the child process never exec’s, the descriptors stay open.
* We close the file descriptor in the child after forking and then reopen it, ensuring that the file has not been modified in the meantime.

3. What happens if the file size of the disk file exceeds the size of the client’s hardcoded buffer? Does the client have sufficient checks to ensure no buffer overruns occur?

* Server will be reading only first (x) characters where x is the max buffer size of the message variable. Client will have the same character limit of the message.
* If the fileSize is greater than the buffer size, the program will still work since the readCapacity is limited to the length (x) of the messageVariable.