I chose to do my project in python, this is due too a couple of reasons. The main reason is Python is a lot simpler when it comes to networking and abstraction. The first thing I wanted to explain is the client that I wrote. The client takes 4 arguments including the host IP address, host port number, request type i.e. PUT or GET and the filename of the file the request is being done too. The first make a URL string which consist of the request type, file and host Ip address and this is sent to the sever. When GET request is used, we wait for the server to send us the file requested in 1024-byte sections. When a PUT request is used, we open the file we want to send to the server and send it line by line in 1024-byte sections. Then we wait in a response and print the response.

The server take a port as an argument and works by waiting on s connections, when a connection is established it saves the IP address and port it came from. Next we save the request as a variable then we pull apart the request. The request send contains the filename and request type which we also save as variables. If the request is a GET we send the requested file to the client. If the request is type PUT we save the file sent to us and send a status back saying the file was received and saved. If at any point there was a IO error we send a 404 not found.

There were not many tradeoffs I could make since python is very abstract and anything is possible. The one tradeoff I did make which may have made the project a bit harder was not using some libraries that made HTTP communication easier.

Some of the test I ran where given in the assignments prompt such as” “myclient www.cnn.com 80 GET index.html”. I tried to use pc1.cs.uml.edu but that host name went unresolved even with a ping test. I also tried making a file that want on the server yet going to the web address that the file would be at, send the file with a PUT request and then going to the web address and it worked. Everything to my knowledge works and I even implemented the multiclient Concurrent Server.





