Networks – Web Server – Diary

**What I did:**

I started work on this assignment around 2 weeks before the deadline. It seemed quite difficult as I’ve never done concurrency in C before.

I began by taking a look at the sample code provided by Ian. It was reasonably straightforward to read through the code and find out how it actually worked. I edited the code to try and understand how exactly it works.

After I built up an understanding of how concurrency works in C, I wrote a multi-threaded server from scratch using the same style as Ian’s code. I got it working up to the level of Ian’s code which was the server just sending back what the client sent it.

Following that, I started to implement the web parts of the server. I first wrote the code to parse a HTTP request and grab the Method, URI and Version which I stored in a struct. Then, I wrote the code to deal with a GET and write the contents of the file back to the client.

My next task was to implement the list of files feature in the directory which I completed (including nested files). Finally, I implemented the checking of the HTTP request to check for errors in the request such as 400, 505, 501 and 404.

Lastly, I checked for memory errors and erased them.

**Challenges**

1. Catching segmentation faults and memory errors in a concurrent C application is really difficult. It takes a long time to even find out where the error is coming from.
2. Parsing the HTTP request was a bit cumbersome. String parsing in C is not as straightforward as Java.
3. I couldn’t implement POST as I working with a filesystem instead of a database.
4. Testing on a OSX environment was quite different to a Linux environment. I got segmentation faults on Linux which wasn’t there on OSX.
5. Using a virtual machine for testing on Linux was quite difficult as the virtual machine was really slow on OSX.
6. Implementing GZip compression was difficult as I was writing files line by line instead of all at once.

**What I would do differently**

1. Test memory errors as implement each feature instead of at the end.
2. Attend more lab sessions to get more help from the instructors.