Hassaan Saeed (hs770)

Haneef Pervez (hp408)

Systems Programming,

Section 07

# Assignment 3 WTF?

**Data Structures:**

- ManifestNode
  - Char * path
  - Char * version
  - Char * hash
  - Struct manifestNode *next
- updateNode
  - Char * tag
  - Char * path
  - Char * hash
  - Char * version
  - Struct updateNode * next

**Implementation:**

We basically coded a network protocol to send over files and their contents between content server by inputting the file length, filename, file content length, and file content into a single string which can then be tokenized into their respective counterparts and the information can be used to clone the files and projects or update or remove content accordingly. We also implemented the SHA 256 hash function in order to create a unique hash. From there, we would input the hash, the file's path, and the version into a manifest Linked List. The Linked List would be traversed and the contents of each node will be written into the .Manifest file accordingly to create the .Manifest. A comparison method was created to compare different .Manifests of different servers and clients. We implemented multithreading and mutex locks to avoid critical sections and different hosts from occupying the same project. We created some minor functions to check for error conditions and such whether a directory or file exists and if so to neatly exit the program. Lastly, we implemented the socket network connection of TCP/IP for a stream connection to read and write data between client and server. To make sure that the client connects to the server's specific host and port, we implemented the getaddrinfo() method which was able to provide the addresslength, the address itself, and the type of IP address making the identification of the hostname much easier and is more reliable than gethostbyname().