Carlos Chasi-Mejia

2-21-2024

Python, DNSServerV3.py

To compile and run sever code, have the python file in the directory and open terminal to the desired directory and run py -3 DNSServerV3.py on the command line. The server will start running and display “Server is listening….” on the terminal.

Description:

The server initializes the values for the server host and port and checks if the DNS mapping file exists , and if it doesn’t exist it creates one. It then reads the DNS file and saves the cache information to the dictionary aka the local cache. Then the server runs, creating a server socket for the host and port and starts listening for connections. Two threads are then started, one to save the DNS cache, and the other to monitor the user input to quit. When a client connects, a connection socket is created by accepting the client and creates a new thread which processes client’s sent query. The server receives the query using receive() and checks if it is in the local cache. If found, the server responds directly from the cache and resolution method is set to “CAHCE”, otherwise it uses gethostbyname() to query the local machine’s DNS API to get the IP address and sets the resolution method to “API”. Then it prints the response to the terminal and sends the encoded response back to the client using send(). The connection socket is then closed, and the new entry is appended to the log file. The thread continues receiving and sending data until the monitor quit thread is called upon. The save file thread constantly checks to see if there were any updates made to the local cache every 15 seconds and writes the updates to the dns file by rewriting every data from the local cache onto the dns file. The monitor quit thread receives the input from the server terminal, and if the input is “exit” the thread opens the dns file and writes all the local cache data into it before the process is killed.