**Written Report: Computer Networks - Socket Programming**

Kelsea Canaday

11/10/2023

**Abstract:**

The file transfer application presented for this assignment establishes a fundamental server-client model to facilitate the transfer of files. Through the implementation of socket programming in Python, the application allows clients to upload files to a server and delete files from the server. The practical utility of this application lies in its simplicity and efficiency, providing a solution for scenarios requiring direct file exchange between systems.

**Introduction:**

The increasing need for efficient data exchange in networked environments motivated the development of this file sharing application. In today's world, where collaboration and data sharing are commonplace, having a reliable means to transfer files is necessary. The application's server-client architecture is designed to address this need, offering an interface for users to upload and delete files on a remote server.

**Methodology:**

The server component is responsible for listening to incoming connections and processing commands from clients. Two main commands, UPLOAD and DELETE, were implemented. For UPLOAD, the server receives the filename and content from the client, saving the file in a designated server folder. The DELETE command prompts the server to remove a specified file if it exists. Python was chosen for the implementation. The code snippets provided offer insight into the structure and logic of the application.

A computer screen with text on it

Description automatically generated

In the main function of the server file, a socket is made to accept one client. It will then make a Server\_Files folder if it doesn’t already exist. It will continuously listen for a client and once the client connects, it will call the handle client function.

A screen shot of a computer program

Description automatically generated

The handle client function will listen for the command the client gives, and once it receives it, the function will go to one of two if blocks. The first if block, the ‘UPLOAD’ block, will take the file name from the client and continuously grab data from the client until there is no more data involved with the file. It returns once the reading is done, and then that file is uploaded into the Server\_Files folder. The second if block, the ‘DELETE’ block, will grab the file name and delete it from the Server\_Files folder.

A computer screen shot of text

Description automatically generated

The client’s main function will determine if the user sends a valid command and file name. It has multiple methods of catching misinputs. If the upload command is issued correctly, it’ll call the upload file function. If the delete command is issued correctly, it’ll call the delete file function.

A computer screen with colorful text

Description automatically generated

In the upload\_file function, it will connect to the server, open the selected file, and send the data to the server until the full file is sent.

A computer code on a black background

Description automatically generated

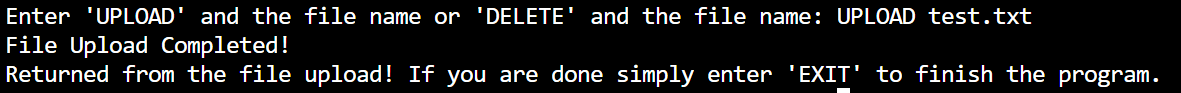
In the delete\_file function, it will connect to the server and then send the name of the file to be deleted. The client program will not terminate until EXIT is entered until the console.

**Experimentation:**

Testing was done to confirm the application's functionality. The application was tested under various scenarios, including uploading files of different sizes and formats, as well as deleting both existing and non-existing files. Screenshots of the program in action were captured to document successful file uploads and deletions. These screenshots serve as concrete evidence of the application's reliability and adherence to the specified requirements.



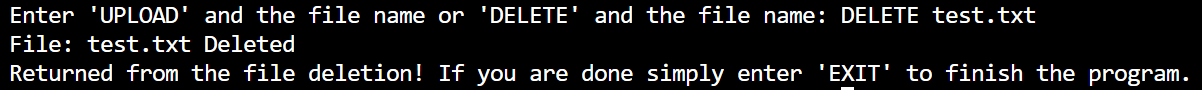
This is an example of a given file not being available to upload onto the server. The console says the file does not exist and returns. It does the same thing in the example of “DELETE no.txt”, instead saying “File no.txt does not exist in Server\_Files”.



A screenshot of a computer

Description automatically generated

When the upload function is called for test.txt, the program reads the data from the original file and puts the read file into the Server\_Files folder.



A screenshot of a computer

Description automatically generated

When the delete function is called for test.txt, it simply deletes the file from the folder.

**Results:**

The experimentation proved the program successful, with the application demonstrating basic file transfer capabilities. The screenshots of successful file uploads and deletions provide a clear visualization of the application's functionality. While the application fulfills the basic requirements outlined, future iterations could include added features, such as encryption for secure file transfers, logging mechanisms, and enhanced error handling to ensure a more efficient and secure file transfer environment.

**Conclusions:**

In conclusion, the file transfer application successfully addresses the fundamental requirements of allowing clients to upload and delete files on a server. The project highlights the effectiveness of socket programming in enabling communication between networked systems. The simplicity of the implemented solution emphasizes its adaptability and potential for use in various contexts where direct file exchange is essential. Moving forward, further refinements can be made to enhance security and reliability, positioning this application as a tool for more advanced file transfer systems. Overall, the project provides valuable insights into the practical application of socket programming and lays the groundwork for future developments in networked file transfer solutions.