CS-342 Operating Systems

Project 1

Section 2

Atilla Alp Yavuz 22102191

Tolga Artun Koçak 22102132

NOTE: The project includes every functionality besides the following features: batch, wsize, quitall, quit.

Number of Clients Hosted vs Elapsed Time Table

The following table represents the time it takes for the client to execute an ls program when the server hosts up to 5 clients, as reflected. (these times were taken by implementing the time.h library into the c code)

Client No	Elapsed Time (millisecond) - ms
1	3.237519
2	13.351973
3	21.678899
4	31.198206
5	32.200332

Interpretation of Results

As the number of clients increases from 1 to 5, there is a clear trend of increasing elapsed time, indicating that the server takes longer to service each additional client. The time taken for a single client is the fastest, at approximately 3.24 milliseconds. With each additional client, the time increases, suggesting that the server's resources or processing capacity is shared among the clients, leading to longer wait times. By the time there are 5 clients, the elapsed time has extended to approximately 32.20 milliseconds, which is nearly ten times longer than the time taken for just one client. This trend could imply that the server has a linear or near-linear scaling challenge in handling multiple concurrent client requests. The results were inconsistent because they showed a high degree of variance. This is most likely due to the limited computing resources of the virtual machine. In cases of inconsistencies we chose the result closest to the average of the results observed.

Evaluation of The Findings

The probable reasoning for the elapsed time of client prompting would be the shared resources within the environment that each client process utilizes. The server's resources, such as CPU, memory, disk I/O, and network bandwidth, are shared among all clients. As more clients make requests, the contention for these resources increases, potentially leading to bottlenecks that slow down the response time. With each additional client, the operating system has to perform more context switches to allocate CPU time to each client's process. Context switching can introduce a delay as the CPU switches from executing one process to another, especially if the number of clients approaches or exceeds the number of available CPU cores. Another reason could be that, as more clients request different data, the cache becomes less effective, causing the server to fetch data from slower storage, thereby increasing the elapsed time.

Another plausible line of logic to predict the trend would be the overhead required for process creation and management of synchronization mechanisms like pipe locks to govern the flow of shared data between the client and the server, the overhead of acquiring and releasing these locks could increase with the number of clients due to contention, leading to longer wait times.