

Iterative Socket Server

Brett Knox & Gary Li



June 30, 2025

CNT4504 - Computer Networks & Distributed Processing

Professor Scott Kelly

# Introduction

# Project Purpose:

*To create:*

* A single-threaded/iterative server that listens to a port & accepts requests from a client server, to then operate to return the requested information to the client
* A multithreaded client that requests information from the server & creates a user-specified number of requests (threads).

# Project Goals:

## ***Server Program***:

* Listen to client requests at a specific port
* serially handle clients
* Perform the following operations:
  + - Determine the date and time on the server,
    - Find how long the server has been running since last boot-up
    - Determine the current memory usage on the server
    - List the network connections on the server
    - Create a list of the users currently connected to the server
    - Create a list of the programs that are currently running on the server
* Collect and send the values obtained from performing the operations above to the client

## Client Program:

* Prompt the user for the IP address of the server to connect to
* Prompt the user for the port to connect to
* Prompt the user to select one of the operations (mentioned in the server portion)
* Ask the user to select how many requests to send to the server
* Output the client & the corresponding response (output requested by that client)

## Testing & Analysis:

* Run/Test the programs and collect the following data for each operation:
  + - Total time to run all requests
    - The number of requests/client threads (corresponding to the other data collected)
    - The average time taken to run a single request
    - The average change in time taken to run a single request based on the number of requests

**Paper – What to Expect:**

* Client-Server Setup & configuration: the design & operations of the client and server programs
* Testing & data collection: collecting the previously specified data from running the programs
* Data Analysis: Analyzing the collected data to determine the effects of increasing the number of clients on the turnaround time for individual clients and on the Average turnaround time in general, as well as determining the primary cause of the effects on the individual client turnaround time and average turnaround time.
* Conclusion: conclusions drawn from data analysis
* Lessons learned: What I learned from this assignment

# Client-Server Setup && Configuration

## Server Program Design & Operation:

The server is designed to listen to a specific port for requests from clients. Once the request is received, the server is to perform the requested operation and send the value returned by that operation back to the client. The client may create multiple requests for the same operation in multiple threads, but the server will only perform one request at a time (serially), thus making it an iterative server. The server is designed to run indefinitely, since it needs to be able to continuously listen to the port for client requests. The operations the server performs implement a “runCommand” method that takes the corresponding terminal commands in string form to run as a parameter, runs the terminal command, reads the result of running the command and stores it in a string variable, then returns that string to be written/sent out to the client.

## Client Program Design & Operation:

The client program is designed to connect to a user-specified IP address and port; to then prompt the user for what operation it wants the server to perform return the results of. Then, the user I prompted for how many requests/ iterations of the previously specified operation they want to perform. Then, the corresponding number of threads is created & performs the requests / sends the request to the server, records the amount of time it takes to get the response back in a list, prints out the response, and ends. Once all the threads have been run, the individual times in the list are added up for the um, then divided by the number of requests to get the average.

# Testing & Data Collection

## How the Programs Were Tested:

We connected to UNF’s VPN and implemented our Programs via .tlp files in Bitvise, then connected the client to the server by inputting the server's IP address and the port the server was listening to. Then, we ran each function to ensure the output was what we expected or aimed for it to be. After ensuring the outputs were correct, we ran each request through the following number of iterations: 1, 5, 10, 15, 20, and 25. We then recorded the Total time taken for all requests & the average time taken per individual request for each load size.

## Data Collected:

The number of requests, corresponding Total Turnaround Time in milliseconds(ms), and corresponding Average individual request Turnaround time ( also in milliseconds(ms)) for each operation: Date & time, Uptime, Memory Used, Netstat, Current users, and running processes.

Date & Time

|  |  |  |
| --- | --- | --- |
| Number of Requests | Total Turnaround Time (ms) | Average individual request Turnaround time (ms) |
| 1 | 0.497 | 0.497 |
| 5 | 1.597 | 0.319 |
| 10 | 4.881 | 0.488 |
| 15 | 3.116 | 0.207 |
| 20 | 4.307 | 0.215 |
| 25 | 6.356 | 0.254 |

Table 1 - date & time

Figure 1 - date & time

Uptime

|  |  |  |
| --- | --- | --- |
| Number of Requests | Total Turnaround Time  (ms) | Average individual request Turnaround time  (ms) |
| 1 | 0.221 | 0.221 |
| 5 | 1.555 | 0.311 |
| 10 | 4.173 | 0.417 |
| 15 | 2.641 | 0.176 |
| 20 | 11.426 | 0.571 |
| 25 | 5 | 0.202 |

Table 2 – Uptime

Figure 2 - Uptime

Memory Used

|  |  |  |
| --- | --- | --- |
| Number of Requests | Total Turnaround Time  (ms) | Average individual request Turnaround time  (ms) |
| 1 | 0.255 | 0.255 |
| 5 | 4.189 | 0.837 |
| 10 | 2.962 | 0.296 |
| 15 | 1.811 | 0.121 |
| 20 | 8.532 | 0.429 |
| 25 | 10.065 | 0.402 |

Table 3 - Memory Used

Figure 3 - Memory Used

Netstat

|  |  |  |
| --- | --- | --- |
| Number of Requests | Total Turnaround Time  (ms) | Average individual request Turnaround time  (ms) |
| 1 | 18.397 | 18.397 |
| 5 | 191.945 | 38.389 |
| 10 | 490.552 | 49.055 |
| 15 | 575.887 | 38.393 |
| 20 | 1243.346 | 62.127 |
| 25 | 2397.806 | 95.912 |

Table 4 – Netstat

Current Users

|  |  |  |
| --- | --- | --- |
| Number of Requests | Total Turnaround Time  (ms) | Average individual request Turnaround time  (ms) |
| 1 | 1.096 | 1.096 |
| 5 | 12.204 | 2.441 |
| 10 | 21.676 | 2.167 |
| 15 | 80.906 | 5.394 |
| 20 | 118.676 | 5.933 |
| 25 | 190.147 | 7.606 |

Table 5 - Current Users

Figure 5 - Current Users

Running Processes

|  |  |  |
| --- | --- | --- |
| Number of Requests | Total Turnaround Time  (ms) | Average individual request Turnaround time  (ms) |
| 1 | 9.785 | 9.785 |
| 5 | 74.731 | 14.946 |
| 10 | 287.234 | 28.723 |
| 15 | 564.704 | 37.647 |
| 20 | 1113.853 | 55.693 |
| 25 | 1463.528 | 58.541 |

Table 6 - Running Processes

Figure 6 - Running Processes

# Data Analysis

## Effect of increasing the number of clients on the Turnaround time for individual clients

Hypothesis tested for each operation at α = 0.05 :

**Null hypothesis (****H₀)**: ρ = 0 (there is no correlation)

**Alternative (H₁)**: ρ ≠ 0 (there is some correlation)

(NS) = Not Significant = fail to reject H₀ = (not enough evidence to support there being a correlation in the population)

(S) = Significant = reject H₀ = statistically significant evidence of a correlation between the number of client requests and the average turnaround time for individual clients.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | date & time | uptime | memory used | Netstat | current users | running processes |
| correlation coefficient | -0.725 | 0.146 | -0.161 | 0.902 | 0.967 | 0.989 |
| t-test | -2.108 | 0.295 | -0.326 | 4.175 | 7.552 | 13.135 |
| t-critical value | 2.447 | 2.447 | 2.447 | 2.447 | 2.447 | 2.447 |
| conclusion | (NS) | (NS) | (NS) | (S) | (S) | (S) |

Table 7 - t-test for correlation between the Average individual response turnaround time and the number of requests

Figure 7 - Change in average individual turnaround time

While some operations( Netstat, current user, and running processes) had a clear change in average individual response time, others (date & time, uptime, and memory usage) had more erratic fluctuation in their times that make it more difficult to determine a possible relationship between the number of requests and average individual response time. At a 95% confidence level, there is statistically significant evidence to support that there is a correlation between the number of client requests and the average individual response turnaround time for the Netstat, current user, and running processes operations, all of which had strong positive correlations. There is insufficient evidence to conclude the same for the date & time, uptime, and memory usage operations.

## Effect of increasing the number of clients on the average Turnaround time

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | date & time | uptime | memory used | Netstat | current users | running processes |
| correlation coefficient | 0.875 | 0.701 | 0.841 | 0.926 | 0.964 | 0.975 |
| t-test | 3.609 | 1.967 | 3.111 | 4.893 | 7.228 | 8.756 |
| t-critical value | 2.447 | 2.447 | 2.447 | 2.447 | 2.447 | 2.447 |
| conclusion | (S) | (NS) | (S) | (S) | (S) | (S) |

Table 8 - t-test for correlation between the Total response turnaround time and the number of requests

At a 95% confidence level. there is statistically significant evidence to support that there is a correlation between the number of client requests and the total response turnaround time for the date & time, memory used, Netstat, and current users, and running processes operations, each of which shows a strong positive correlation. There is insufficient evidence to conclude the same for the uptime operation.

## Primary Cause of the effect on individual client turnaround time and average turnaround time

While we can’t determine a direct causal relationship between (individual client turnaround time & average turnaround time) and the volume of requests based on our collected data, we can make an inference that there is a relationship between the two ( though with some exceptions, variability, and/or other external factor influence ). Logically, one would be able to conclude that since the code implementation should take a specific amount of time to run, that multiple iterations would make the total amount of time increase ( larger loads => larger total time), while the thread creation timing overhead would increase the average individual turnaround time, but through a strictly data-driven lens, you cannot universally determine the primary cause of the change in these times based on this sample/test. On the other hand, we could reasonably assert that the latency of the network Is a more significant factor than the number of requests, since it directly affects the amount of time it take for the requests and responses to be sent, but due to limited scope of the recorded data, we cannot formally confirm this relationship.

Conclusion

At a 95% confidence level. there is statistically significant evidence to support that there is a correlation between the number of client requests and the total response turnaround time for the date & time, memory used, Netstat, and current users, and running processes operations, each of which shows a strong positive correlation. There is insufficient evidence to conclude the same for the uptime operation.

At a 95% confidence level, there is statistically significant evidence to support that there is a correlation between the number of client requests and the average individual response turnaround time for the Netstat, current user, and running processes operations, all of which had strong positive correlations. There is insufficient evidence to conclude the same for the date & time, uptime, and memory usage operations.

# Lessons Learned

* How to run our programs through the UNF/ FortiClient VPN & Bitvise
* How to create a multithreaded client
* How to make a server listen to a specified socket
* How to send and retrieve data to/from a client and server
* How to perform a 2 tailed t-test in Excel
* You can add Word documents to a GitHub repository.
* How to execute Linux system commands in our code
* To make my time variables doubles instead of longs for more accurate recording