

Lesson 2: Mastery Project

Installation 1

Submission

1. Based on what you proposed, describe what will you be submitting for Installation 1.

We are submitting the implementation of client-server communication in installation 1. The code implementation involves the sequence of function calls in socket programming in which the client connects to the server, transfer of connection establishment messages and connection closure. The entire installation is demonstrated using command line interface which will later be modified into GUI in future installations. Also, we are sharing the packet trace of the client-server communication which is captured using Wireshark.

2. What code files are you submitting for Installation 1?

The two major code files of installation 1 are client.py and server.py. The code will focus on command line implementation of basic client-server communication. Using the menu options provided in the command line, client tries to make a connection with the server. Also, these files contain function declarations (function name, arguments) which will be defined/implemented. Function related to connection establishment has been implemented in this installation and some of the other important function declarations like create_connection(), upload_file(), download_file(), list_files() and quit() etc....are included.

3. Provide a URL to your video demo. Remember, the contents will be evaluated against what you proposed for the demonstration in Lesson 1.

https://nau0-my.sharepoint.com/:v/g/personal/vg588_nau_edu/EX9eRB1Y35tGpE2kXuopO6YBWjbSfcNC4JSyJzXW4tYHPA?e=Ous3Z3

4. Paste the Installation 1 Evaluation Rubric that you developed in your proposal below. This will be used to help evaluate your Installation.

	Fail (0)	Pass (1)	Result
[Item 1]	This installation fails to demonstrate mastery if server is not running on the specified address and port.	This installation succeeds at demonstrating mastery by running server at a specific address and port number.	Pass

[Item 2]	This installation fails to demonstrate mastery if server is not able to respond to the client requests.	This installation succeeds at demonstrating mastery by server accepting the incoming connections from the client.	Pass
[Item 3]	This installation fails to demonstrate mastery if server is not able to serve more than 5 clients.	This installation succeeds at demonstrating mastery by serving at least 5 clients at a time.	Pass
[Item 4]	This installation fails to demonstrate mastery if server and client are not able to run in the windows machine.	This installation succeeds at demonstrating mastery by running the client and server programs in windows OS.	Pass

5. What were some challenges that you encountered during the implementation of this installation and what are some specific ways you can work to overcome these challenges?

The challenges encountered during the implementation of installation 1 are mentioned below :

- ➔ When client connected to server, if client has terminated its execution abnormally then server went into deadlock (infinite loop) which was further resolved by handling an exception on server side. This happened because the server was continuously fetching the data which didn't exist in the buffer which led to deadlock.
- ➔ When client has sent a connection closure request, the server closed the connection. But as server runs in an infinite loop it was again trying to read the data from a closed connection which created an error. This error was further resolved by changing the flow of our implementation.
- ➔ We faced issue when server tried to handle more than 1 client in parallel. This was handled by implementing the concept of Multithreading which allowed to serve multiple clients at a time.