COMP9331 – Assignment Report

Programming Language: Python 3.13

Running Platform: WIN11 OS; Tiger VNC (CSE) or Local machine

Features implemented:

UDP: login (authentication, registration), exit forum, create thread, list thread, post message, read thread content, edit message, delete message, remove thread (& related files)**.** ***TCP & UDP:*** upload file & download file.

Data structure implemented:

Dictionary and List, easy handling in Python. For example,

- *user_credentials(dict) store {username: password};*
- *active_users(dict) store {username: client address};*
- *thread_metadata(dict) store {title: {"owner": str, "messages": list, "files": list}}, mainly used for finding post owners and record lists.*

Application Layer Protocol:

- Based on request – response model on text commands
- Communication Process**:**
    1. After the client starts, go through the authentication process (UDP).
    2. The client displays a prompt to the user and receives user commands after successfully authentication.
    3. The client sends the command (req_user, threadtitle, message etc.) to the server.
    4. The server receives the request then parses the command and arguments.
    5. The server executes corresponding operation (thread management, UPD, DWN).
    6. The server encodes the results (success or error message) to corresponding client address.
    7. The clients receive and show the corresponding response.

Transport Layer Protocol: TCP (file transfer); UDP (command exchange)

Potential problems or existing problems:

1. *Code structure optimization*, too many if…else… syntax, the logic is simple but hard to read and debug.

2. *Error handling*, basic error checking implemented, split() syntax could fail because of unexpected formats, error massage could be more specific (format output with more parameters).

3. *UDP reliability*, the current client retransmission mechanism is very basic (fixed number of times, fixed timeout), cannot handle complex operations, and cannot guarantee that messages will be delivered.

Program design and use:

1. Run server.py, *python3 server.py <port number>*
2. Run one (or multiple) client.py, *python3 client.py 127.0.0.1 <port number>*
3. Interaction between clients and server (For example, user 'Batman' from client x and user 'Superman' from client y and 'WonderWoman' from client z interacting through server).
    - User A runs client.py 127.0.0.1 <port number> on terminal 1, logs in/registers as "Batman".
    - User B runs client.py 127.0.0.1 <port number> on terminal 2, logs in/registers as "Superman".
    - "Batman" enters CRT BvSScripts. Server responds Thread BvSScripts created.
    - "Batman" enters LST. Server responds Active threads:\n BvSScripts.
    - "Batman" enters RDT BvSScripts. Server responds Thread is empty.
    - "Batman" enters MSG Do you bleed? You will! Server responds Message posted.
    - "Superman" enters RDT BvSScripts. Server responds 1 Batman: Do you bleed? You will!
    - "Batman" enters MSG BvSScripts You were never a God. You were never even man! Server responds Message posted.
    - "Superman" enters MSG BvSScripts Save Martha. Martha. Martha. Server responds Message posted.
    - "Batman" enters RDT BvSScripts. Server responds 3 Superman: Save Martha. Martha. Martha.
    - "Batman" enters UPD BvSScripts Doomsday.exe. Local file Doomsday.exe is uploaded. Client shows Server confirmation: UPLOAD_SUCCESS.
    - "Superman" enters DWN BvSScripts Doomsday.exe. File is downloaded to Superman's client directory. Client shows BvSScripts-Doomsday.exe downloaded.
    - User C runs client.py 127.0.0.1 <port number> on terminal 3, logs in/registers as "WW".
    - "WW" enters DWN BvSScripts Doomsday.exe. File is downloaded to WW's client directory. Client shows BvSScripts-Doomsday.exe downloaded.
    - "Superman" enters XIT. Server responds Goodbye, Superman! client exits.
    - "Batman" enters DLT BvSScripts 1. Server responds Message deleted.
    - "Batman" enters DLT BvSScripts 2. Server responds Message deleted.
    - "Batman" enters RMV BvSScripts. Server responds Thread BvSScripts removed, BvSScripts-Doomsday.exe removed.
    - "Batman" enters LST. Server responds No threads exist.

- "Batman" enters CRT JusticeLeague. Server responds Thread JusticeLeague created.
- etc……

References:

- WebCMS3 - COMP9331 - Programming Tutorial – Python sample solution
- os — Miscellaneous operating system interfaces — Python 3.13.3 documentation
- Regular Expression HOWTO — Python 3.13.3 documentation
- threading — Thread-based parallelism — Python 3.13.3 documentation
- concurrent.futures — Launching parallel tasks — Python 3.13.3 documentation
- Batman v Superman: Dawn of Justice (2016) - full transcript