COMP 3331/9331

Ass 020

Online Discuss

All details are in the specification

- **READ THE SPECIFICATION**

- **READ THE SPECIFICATION (AGAIN)**

- Information about d submission instructions, marking guidelines, various other specifics are in the specification

- Choice of programming languages: on

- This talk provides a high-level overview

Client Server Archi

- Server
  - Always on
  - Responsible for managing the discussion forum
    - Maintains state about the forum
  - Implements a request/   client(s)
  - Prints informative upd

- Client
  - Allows a user to interact with the discussion forum
  - Interaction is through the command line terminal

- Transport Protocol
  - Must use TCP

Execution

- Server
  - Command line arguments:
    - Server port (use a value greater than 1023 and less than 65536)
    - Admin password (us
  - Executed first – wait

- Client
  - Command line arguments:
    - Server IP address (use "localhost" as client(s) and server will be executing on same machine)
    - Server port number (should match the first argument for the server)
  - Let the OS pick an ephemeral port
  - Client should initiate TCP connection with server ("127.0.0.1", server port)

Two Configurations

- **One client at a time** (14 marks – 70% of the assignment marks)
  - At any given time only one client will interact with the server
  - Simple sequential interaction between client and server
  - Should not require multi-threading
  - Multiple clients can co   nt connects, executes some
    commands, quits, seco   mands, quits, etc.
  - You will need to defin   n state of the message
    board (# of active threads, thread creators,   e, etc.)
    - You must manage several files
      - one for each thread and any specific files uploaded by users
- You must design your own **application layer protocol**
  - This includes the syntax/semantics of the messages exchanged between the client and server
    and the actions to be taken upon receiving each message
- Non-CSE Students may chose to only implement functionality for this configuration
  - **MUST request approval** – check spec for details

Two Configurations

- **Concurrent multiple clients** (6 marks – 30% of the assignment marks)
  - Client program should only require minor changes
  - Server program must interact with multiple clients
    - We recommend a **multi–threaded approach**
    - Main thread should l   nts
    - A new thread should   ith each client
    - The interactions with   equential as in the first configuration
    - However, the message board state is now   ple clients
      - Client A adds a new message to a thread. F   n Client B reads the thread, client A's message should be viewable
    - You may assume that each interaction with a client is "**atomic**"
      - Client A initiates an interaction to add a new message to a thread. While the server is processing this interaction, it cannot be interrupted by a command from another client (B). Client B's command will be acted upon after the command from client A is processed
  - You must be particularly careful about managing multiple threads and how they access the shared state (data structures, files, etc.) for the message board

User Authentication

- A credentials file will be available in the server working directory (example is on the assignment page)
- The client will first a
  - Ask user to enter us
  - If username matches with one of the e   redentials file, the user is requested to enter the password
    - If password matches, authentication successful
    - If password doesn't match, the user is asked to enter username again
  - If username does not match it is assumed this is a new user and user is asked to enter a new password
    - User is authenticated and a new entry is created in the credentials file

Discussion Forum

- 11 specific commands

- User is prompted to enter one of the commands

- Basic error checking   guments, etc. should be implemented
  - NOTE: We are more interested in obser   e implemented the "normal" functionality, so our tests for   will be limited and won't include bizarre edge cases

Commands

- CRT – Create thread
  - `CRT threadtitle`
  - Create a new thread – represented as a text file (threadtitle), update state information about active threads and thread creators

- MSG – Post message
  - `MSG thread mes`
  - Add message to the thread – update t (message number, username, message)

- DLT – Delete message
  - `DLT thread messagenumber`
  - A message can only be deleted by the original creator (error if condition not met)
  - Delete message – update the thread file (remove the message and update message numbers if required)

Commands

- EDT – Edit message
  - `EDT threadtitle messagenumber message`
  - A message can only be edited by the original creator (error if condition not met)
  - Edit the thread – upda essage)

- LST – List threads
  - `LST`

  - List all active threads
  - Read data structure that stores this information and send it to client

- RDT – Read thread
  - `RDT threadtitle`
  - Display all messages contained in thread
  - Read the corresponding thread file and send relevant information to client

Commands

- UPD – Upload file
  - `UPD threadtitle filename`
  - Create filename in server working directory, and store all file data received from client
  - Update thread file wit   e uploaded filename)

- DWN – Download file
  - `DWN threadtitle filename`
  - Create filename in client working director   le data received from server

- RMV – Remove thread
  - `RMV threadtitle`
  - A thread can only be removed by the original creator (error if condition not met)
  - Remove all state associated with the thread (thread file, uploaded files)

Commands

- XIT – Exit
  - XIT
  - User quits
  - Update state about

- SHT - Shutdown
  - SHT adminpassword
  - Server shutdowns – all state maintained is removed, all active clients are informed

How to start with Clp

- Focus on **first configuration**
- Get the client and server to setup a TCP connection
- Implement authentication
- Implement one command ... ext
- Suggest the following ord ... , DLT, UPD, DWN, RMV, SHT
- Once all individual commands are done and te ... e sure you test extensive interaction with a single client (meaningful co ... mmands)
- Next progress to **second configuration – multi**
- **STRONGLY SUGGEST TO DEVELOP YOUR IMPLEMENTATION IN VLAB ENVIRONMENT (NOT ON YOUR NATIVE MACHINE)**
- Added benefit – CSE accounts are backed up

Resources

- Many program snippets are on the web page
  - Including multi-threading code snippets
- Your socket progra   2 and 3 will be useful
- Repository of resour https://eduassistpro.github.io/

Testing

- Test, Test, Test
- Server and client(s) executing on **same machine but different working directories**
- Emphasis on correct b
- Basic error checking
- **MUST Test In VLAB environment**
- If we cannot run your code, then we ca ou any marks
- Your assignment will be MANUALLY marked by your tutors
- Client and server must display meaningful messages at the terminal
- Extensive examples are available at the end of the spec
- You are NOT required to use the exact same text

Plagiarism

- **DO NOT DO IT**

- If caught

  - You will receive zero marks (a
    not your first offenc
  - Your name will be added to th

Seeking Help

- Assignment specific consults (for all 3 programming languages) from Weeks 7-10

  - Schedule to be announced in a few days

- Course message for

  - Read posts from other students before   question

- Read the spec – very often your ans   n there