

Whiteboard application

- Goal: Write a “whiteboard” application, where a community of users could interact and exchange messages.
- A server will wait for connections, on a INET socket, forking new processes to handle interactions with clients (e.g. there might be one process per client)
- Interacting with the application, users will:
 - Authenticate
 - List, subscribe and create "topics"
 - Append a message to one "topics"
 - See the status of sent messages (received/published)
 - Receive (and reply to) messages, by posting to topic
- Notes:
 - For the implementation, use processes (not threads) and SYSV IPCs (not POSIX semaphores, etc).
 - Once a message is published to a "topic" all subscribed users will be able to get a copy of it
 - Messages cannot be edited or deleted once sent/published
 - Users are added/deleted to the system by an external administrator which manages credentials
- Client commands (1):
 - authenticate
 - Reads and sends user name and password to server
 - list [messages|topics]
 - Gets list of messages, read or unread, ordered by topics
 - Lists available|subscribed topics
 - get [message#]
 - Receives and display message on user console
 - status [message#]
 - Displays the status of a specific message
 -
- Client commands (2):
 - reply [message#]
 - appends a new message to a thread (in a topic)

- create [topic]
 - creates a new topic (user will be the owner)
- append [topic] [thread]
 - appends a (new) message to a new thread in a topic
- subscribe [topic]
 - inserts the user in the list of recipients for this topic
- delete [topic]
 - only if owner of the topic, deletes the topic and all messages

Evaluation Scorecard

- The code works (reliably)
- Race conditions
- Robustness under unexpected situations
 - Misbehaved clients
 - Communication errors
 - Reboots/crashes
- Implementation of new commands
- ...
- Source Code, including for each function/global variable/data structure:
 - Purpose
 - Parameters, side effects and return value (for functions)

DOCUMENTATION describing:

- Design Choices (and the reasoning behind them)
- Macro modules and their interaction
- Test cases
- Release notes, including limitations, known errors, etc

Suggestions

- Be creative:
 - Evaluate possible alternatives, but (as a starting point):
- The whiteboard data structure should reside in shared memory and protected with SYSV semaphores
 - Apply the KISS principle (keep it simple s.. :^)
 - Start prototyping early, use scaffoldings, (e.g. functions returning just a plausible value, etc..)
- Few days before deadline, send a draft of code and documentation for review/interaction.