

Post Struct

This is a struct that represents posts and is housed in the server file. It has the following parameters:

- User
- Content,
- Post_time_string
- Post_time

It's used to represent posts in a user's timeline. All parameters are strings except for post_time, which is of type time_t

Client

connect_to():

1. Create a stub using NewStub() and CreateChannel(). Store stub as member variable in client class
2. Create container variables to pass to the Login RPC
3. Make a call to Login()
4. Return appropriate value based on RPC status

process_command():

1. Create an IReply
2. Parse input string and store as command and args
3. Create container variables for RPC calls based on command and args
4. Make correct RPC
5. Store RPC return values in IReply accordingly
6. Return IReply

process_timeline():

Server

User class:

- Member Variables
 - o Name
 - o following
 - o followers
 - o timeline
- addUser()
- Setters
- Getters
- Follow()
 - o Adds a user's name to following list
- Unfollow()
 - o Adds a users name to followers list

- Add_follower()/remove_follower()
 - o Both modify the member vectors

Server Class:

- A class to represent the server. There's a single global instance of this class
- Member variables
 - o users(vector of Users)
 - o numUsers
- the server functions are almost identical to the user functions, but act on a higher level of abstraction

RPC implementations:

All the RPC's have similar implementation as follows:

1. grab values from request
2. call server methods to implement the RPC functionality, passing request arguments
3. store the return values of the server method calls in the reply
4. return status of RPC function