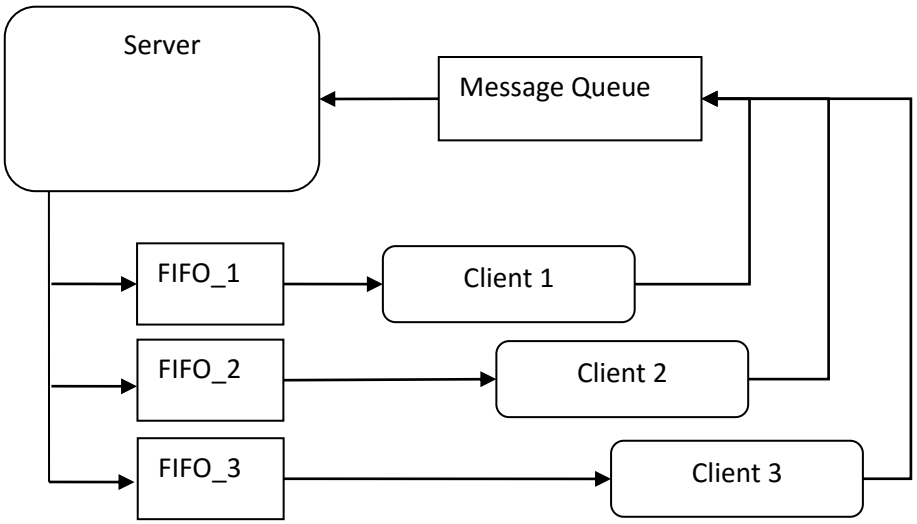
ISE 302–Operating Systems
2017–2018, Homework 3
23.11.2017

Introduction

You will develop a chat server and a client application using inter-process communication methods. The server should be able to handle any number of clients. The clients can log in, log out and send messages. The server must use a linux message queue for listening messages from clients and FIFO structures for relaying messages to clients. An example flow for a server and three clients can be seen in the following figure.



Server

- Any number of clients can be connected using unique usernames. No need to apply security measures. You can assume that each client will use unique usernames.
- A linux message queue should be generated on start-up. You can use your student id as a key for the message queue.
- Server should listen the linux message queue and according to the message:
    - Upon receiving a login request, it should add the user to the receiver list.
    - The incoming messages from the clients should be relayed to the other users that are in the receiver list with the username information of the sender. The FIFO structure of the users will be used for sending messages to the users.
    - Upon receiving a logout request, it should remove the user from the receivers list and deallocate the FIFO structure of the user.
- When “q” character is read from the terminal, server should remove the allocated resources (linux message queue and FIFO structures).

Client

- Client should read the username information from the terminal, create a FIFO structure for the user and send a log in request to the server over the linux message queue.
    - Each client should create a unique FIFO structure with the username. Ex: if the username is “kocca”, the FIFO structure should be named as “FIFO_kocca”.
- A thread should be created for reading the FIFO structure for incoming messages. It should print the received messages to the terminal.
- Each line read from the terminal should be sent to the server.
    - When “q” character is read from the terminal, client should send a log out request to the server over the linux message queue, and stop working.
- Upon the removal of the related FIFO structure, the client should stop working.

Messages

The following table lists the messages that should be sent over the linux message queue.

Description	Message content	Example (username=”kocca”)
Log in	login:username	login:kocca
Send message	username:message	kocca:Hi All!
Log out	logout:username	logout:kocca

Messages and usernames are limited to 180 and 10 characters, respectively.

Output

Possible program outputs for a server and clients are given below.

Server

Message Queue is generated.
‘kocca’ is connected.
‘kocca1’ is connected.
‘kocca2’ is connected.
‘kocca2’ is disconnected. Removing ‘FIFO_kocca2’
>q
Removing ‘FIFO_kocca’
Removing ‘FIFO_kocca1’
Message Queue is removed.
Server connection closed. #stopped working

Client 1

Enter a username:
>kocca
‘FIFO_kocca’ created.
Log in request sent.
>Hi All!
Message sent.
kocca1: Hi
Server connection closed. #stopped working

Client 2

Enter a username:
>kocca1
‘FIFO_kocca1’ created.
Log in request sent.
kocca: Hi All!
>Hi
Message sent.
Server connection closed. #stopped working

Client 3

Enter a username:
>kocca
‘FIFO_kocca2’ created.
Log in request sent.
kocca: Hi All!
kocca1: Hi
>q
Log out request sent. #stopped working

Notes

- You should submit 2 separate source codes, for a client and a server.
- You are required to use FIFO structure and linux message queue for client-server communication.
- You will lose points if your program **does not**
    - work for arbitrary number of clients.
    - deallocate resources.
    - avoid busy waiting.
    - stop working under the given conditions.
- Include development environment information, compilation and running commands as a comment in your code.

EXAMPLE:

```
// Development environment: Lubuntu 16.04 or ITU SSH servers
// To compile: g++ -c Homework3.cpp –pthread
// To run: ./a.out input.txt
```