Operating Systems Lab
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320232
Date: 2016-10-13
Deadline: 2016-10-19

# Problem Sheet #7

**Problem 7.1:** *posix message queues* (2+2+1+2+3 = 10 points)

Extend the program written in the previous assignment into a chat system.

a) The server program opens a second inbound message queue (with the well-known name `/mqchatsc`) over which clients can send control messages to the server. The server uses the message queue notification API to process incoming messages in a separate thread. Do not create your own thread, let instead the message queue implementation do the work.

b) The client program opens an inbound message queue (with the name `/mqchat-pid` where pid is replaced by the process identifier). The client uses the message queue notification API to process incoming messages in a separate thread. Received messages are printed to the standard output.

c) Define `subscribe` and `unsubscribe` messages that the client can send to the server's control queue. The subscribe message should include the client's process identifier and the name of the client's incoming message queue.

d) Upon startup, the client should send a `subscribe` message to the server and the server should then connect to the clients inbound message queue.

e) Modify the server such that it forwards any messages received via the server's inbound message queue to the message queues of all subscribed clients.

Test your client and server by sending a larger messages through them (e.g., redirect the clients standard input to read from a file), try to make sure the server does not block, e.g., when clients stop to process their incoming queue.