Operating Systems Lab
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320232
Date: 2016-10-06
Deadline: 2016-10-12

Problem Sheet #6

Problem 6.1: *posix message queues* (3+2+1+1+1+2 = 10 points)

Write a server and client program using POSIX message queues that can exchange simple messages.

a) The server program opens an inbound message queue (with the well-known name `/mqchatsq`) over which clients can send 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. The server simply prints received messages to the standard output. You may use `pause(2)` to suspend your main thread.

b) The client program connects to the server's inbound message queue and then goes into a loop reading lines from standard input and writing them as messages to the server. The client program terminates when the end of the standard input has been reached.

c) What happens when a client writes to the server's inbound message queue when the server is suspended (i.e., the server does not process any messages)?

d) Can the inbound message queue of the server exist even if the server who created it initially has terminated?

e) What happens if the server restarts and there are already messages in the server's inbound message queue?

f) Rewrite your programs such that clients never block on the server's inbound message queue and that the server handles any potentially queued messages.

Note that we will extend this program into a fully functioning chat server and client using POSIX message queues. So make sure that your code is well structured and extensible.