

Lab Requirement 5 - Message Queues

Requirement

Write TWO C programs that simulate a server process and several client processes that can communicate together. The server accepts a string from a client, converts the upper case letters to lower case letters and vice-versa, then it sends it back to the client.

Read the instructions carefully as well as the notes in the end before starting to write any code!

Approach: Message Queues

The client reads a message from the standard input and sends it to the server via the *Up* queue. Then, it waits for the server's answer on the *Down* queue. When the client receives the message from the server, it prints it out. You may assume the maximum size of any message is 256 bytes.

Multiple clients must be able to connect to the *Up* and *Down* queues. However, what happens if one client puts a message to convert on the system and another client is waiting for its response from the queue? There are different ways to handle this, but you should handle this using message types (tag). Each client has a particular client number based on the last 4 digits of its **Process ID**. Use this 4-digit number as the client identifier and make it the message type so that each client can always be sure to receive the message that it is waiting on to be converted.

Important Notes

The server and clients are completely different programs with different main functions. That is, **NO FORKING** is required!

The server and client should be running forever using a while loop. However, they can be terminated by using Ctrl+C (SIGINT).

The server creates the shared resources and the clients use (retrieve/get) them. When the server is terminated (by SIGINT), all the shared resources must be deleted as well. (Assume the server always exits after all the clients!)

You can use the following code for conversion.

```
#include <ctype.h>
/* convert upper case to lower case or vice-versa */
void conv(char * msg, int size)
{
    int i;
    for (i = 0; i < size; ++i)
        if (islower(msg[i]))
            msg[i] = toupper(msg[i]);
        else if (isupper(msg[i]))
            msg[i] = tolower(msg[i]);
}
```