Zewail City of Science and Technology
University of Science and Technology
Communication and Information Engineering Program
Operating Systems (CIE 302)

Lab Requirement 6- Shared Memory and Semaphores

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: Shared Memory and Semaphores (+ Signals)

The client reads a message from the standard input and writes it to the shared memory. Then it sends a signal to the server asking it to process the data it wrote to the shared memory. Then it sends a signal to the client telling it that it finished processing the data. When the client receives the message from the server, it prints it out. You may assume the maximum size of any message is 256bytes.

The server must be able to handle multiple clients simultaneously. However, what happens if one client puts a message to convert on the system and another client is waiting for its response from the server? You should handle this through semaphores.

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]);
}
```