

Para la realización de la práctica he decidido crear globalmente en el fichero del servidor: un contador de clientes conectados en el servidor, y dos booleanos para saber si un cliente o el servidor está abiertos. A la hora de aceptar un nuevo cliente, se comprueba si el servidor está cerrado o no. Si el servidor está en modo abierto, el número de clientes aumenta en uno y se lanza su correspondiente thread. Para ahorrarnos los "joins" de los threads he dado uso de la función detach() para en caso de un cierre inesperado que se liberen correctamente. Dentro de la función que realiza cada thread también he añadido dos funciones las cuales se ejecutan cuando detectan una señal (ctrl+c / ctrl+z) desde el servidor o desde el cliente respectivamente. He creado un monitor "controlAvion" para hacer uso de los datos compartidos de una forma correcta entre los clientes.

Las dificultades que he tenido han sido sobre todo a la hora de qué debía realizar el programa cuando detectaba las diferentes señales.

Durante las ejecuciones del programa, se observa que al realizar ctrl+c o ctrl+z desde el cliente, ambos, cliente y servidor se cerraban. Y si estos comandos eran introducidos desde el servidor, solo este se cerraba inmediatamente dejando a los clientes colgando.

Ficheros:

- Socket.cpp / Socket.hpp
- ControlAvion.cpp / ControlAvion.hpp
- ServidorMulticliente.cpp
- Makefile_p5_SERVER
- Ejecuta_p5_SERVER.bash