Lab08 – Timer & Signal

Sun, Wen-Lin

In lab5, we used multi-tasking server design to implement two different concurrent echo servers with processes and thread, respectively. Each client would send multiple commands. Both of the echo servers should receive and reply every commands from the client until the client close the connection. All the clients were served simultaneously. Now, in this lab, we have to strengthen the server that is able to clean out the idle clients. **The server should be implemented with process, timer and signal.**

The server should print out some messages:

1. When it accept a client:
   Accept Client [no. of the client]
2. When it receives a message from a client:
   Received "[received message]" from Client [no. of the client]
3. If a client does not send any message in 10 seconds, send the message below to client, close the connection and kill the child process.
   Timeout: Client [no. of the client]

Usage of the testing script:

./run_clients_no.sh [server_ip] [server_port] [#of normal clients] [#of idle clients]

Hint:

You may find that some kinds of timers will interrupt your I/O operations, and some kinds of timers will be blocked by the I/O operations. Try to figure out their relationship and modify your program to meet the requirements.