**All-in-One Server**

A server process S gets input data from processes P1, P2, P3 and from keyboard(standard input). P1 is connected through pipe and P2 is connected through FIFO(named pipe) to S.

Process P3 sends its standard output to S.

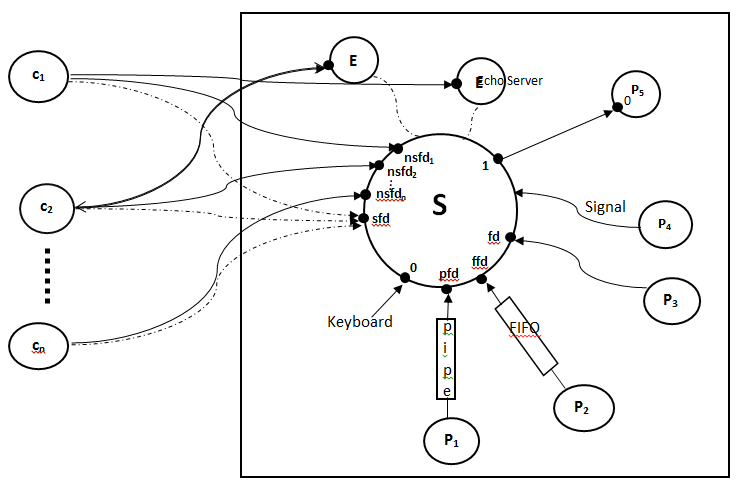
The server S also waits for chat connection requests from clients on a famous FIFO sfd and it accepts any such requests and also creates individual chat(echo) FIFOs as nsfdi.

Process P5 is connected to server S, in such a way that the standard output(fd value 1) of S will be the standard input(fd value 0) of P5.

If S gets data from keyboard or P1 or P2 or P3, it sends the same data to all connected chat FIFO descriptors (nsfds) and also to standard output by cout<<.. statement to process P5.

If S gets a signal from process P4, then it handovers an already chat connected client connection on first-cum-first basis to a separate newly created echo sever process E. From then onwards the client will be served by E only ( like server sends something and client displays). Likewise, whenever S gets a signal from P4, a separate echo sever process E will be created to serve a chat connected client. And if any echo server process E exits, then that E server’s client will again be receiving data (getting served) by server process S.

Implement a program (full code) for the server process S and for P1, P2, P3, P4, P5, Echo Server, and client. The program should **not use threads**.



**“Motivation is the art of**

**getting people to do**

**what you want them to do**

**because they want to do it. “** [Dwight D. Eisenhower](https://www.brainyquote.com/authors/dwight-d-eisenhower-quotes)