Birla Institute of Technology & Science, Pilani
Pilani Campus

II SEMESTER 2021-2022
Assignment-1

Course No.:	**IS F462**	**Course Title:**	**Network Programming**
Deadline:	**As on Canvas**	**Maximum Marks:**	**60M (15%)**

Note:

- Maximum of two students per group. Upload code in Canvas.
- Name your file idno1_idno2_assignment1.tar .

P1. You are required to build a bash-like shell for the following requirements. Your program should not use temporary files, popen(), system() library calls. It should only use system-call wrappers from the library. It should not use *sh* or *bash* shells to execute a command.

a) Shell should wait for the user to enter a command. User can enter a command with multiple arguments. Program should parse these arguments and pass them to execv() call. For every command, shell should search for the file in PATH and print any error. Shell should also print the pid, status of the process before asking for another command.

b) shell should create a new process group for every command. When a command is run with & at end, it is counted as background process group. Otherwise it should be run as fore-ground process group (look at tcsetpgrp()). That means any signal generated in the terminal should go only to the command running, not to the shell process. `fg` command should bring the background job to fore ground. `bg` command starts the stopped job in the background.

c) shell should support any number of commands in the pipeline. e.g. `ls|wc|wc|wc`. Print details such as pipe fds, process pids and the steps. Redirection operators can be used in combination with pipes.

d) Shell should support `#` operator. The meaning of this: it carries same semantics as pipe but use message queue instead of pipe. The operator `##` works in this way: `ls ## wc , sort`. output of `ls` should be replicated to both `wc` and `sort` using message queues

e) Shell should support `S` operator. The meaning of this: it carries same semantics as pipe but use shared memory instead of pipe. The operator SS works in this way: Using example, `ls SS wc, sort`. Output of `ls` should be replicated to both `wc` and `sort` using shared memory

f) Shell should support a command `daemonize` which takes the form `daemonize <program>` and converts the program into a daemon process.

g) shell should support <, >, and >> redirection operators. Print details such as fd of the file, remapped fd.

Deliverables:

- Brief Design Document (.pdf)
- shell.c

[30 M]

P2. In this problem let us extend Message Queues network wide for the following characteristics.

- One who writes a message is called a publisher and one who reads is called as subscriber. A publisher tags a message with a topic. Anyone who subscribed to that topic can read that message. There can be many subscribers and publishers for a topic but there can only be one publisher for a given message.
- Publisher program should provide an interface for the user to (i) create a topic. Publisher also provides commands for (ii) sending a message, (iii) taking a file and send it as a series of messages. When sending a message, topic must be specified. Each message can be up to 512 bytes.
- Publisher program takes address of a Broker server as CLA. There can be several broker servers on separate machines or on a single machine. The role of a broker server is to receive messages from a publisher and store them on disk and send messages to a subscriber when requested,
- Publishers and subscribers may be connected to different brokers. The messages should reach the right subscriber.
- Subscriber program takes the address of a broker server as CLA at the startup. It allows a user to (i) subscribe to a topic (ii) retrieve next message (iii) retrieve continuously all messages. Subscriber should print the message id, and the message.
- All brokers are connected in a circular topology. For message routing, the broker connected to a subscriber, queries its neighbor brokers and they query further and so on. Each query retrieves a bulk of messages limited by BULK_LIMIT (default=10).
- Brokers store messages for a period of MESSAGE_TIME_LIMIT (default=1minute)
- This system doesn't guarantee FIFO order of messages. Think and propose any mechanism that can guarantee FIFO order.

Deliverables:

- Publisher.c, Subscriber.c, Broker.c
- PDF file explaining design decisions and documentation

[30M]

--&--