# BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJASTHAN)
# I SEMESTER 2019-2020
## ASSIGNMENT-2

|  |  |  |  |
|---|---|---|---|
| **Course No.:** | **IS F462** | **Course Title:** | **Network Prog.** |
| **Deadline:** | **15-Nov-2019** | **Maximum Marks:** | **48M (12%)** |

**Note:**

- Maximum of two students per group.
- Upload code in http://nalanda.bits-pilani.ac.in Name your file idno1_idno2_assignment2.tar .

**P1.** You are required to build a custom-made file system "bigfs" as described below. The purpose of this file system is to store large files on multiple systems.

- Any file larger than 1 MB is split into blocks of 1MB. Each block is stored on a separate physical machine using local file system available on that m/c.
- A client provides commands to the user to create, copy, or move a file just like a local file system provides. Commands such as `cat`, `cp`, `mv`, `ls`, `rm` should be provided in client interface for the user.
- A client should also be able to copy from local file system to bigfs and vice-versa.
- A server named "FileNameServer" running on one of the systems is responsible for maintaining the file system hierarchy. A server named "FileDataServer" running on all m/cs is responsible for reading/writing the data blocks stored on that machine.
- A client contacts FileNameServer first for executing all commands. Commands such as ls, mv will execute completely on FileNameServer without a need to contact FileDataServers. Commands such as `cat, cp, rm` require the client to contact both FileNameServer and FileDataServers.
- When a command such as `cat` or `cp` is issued, client should read a file in parallel from multiple FileDataServers. Similarly, when writing a file using `cp`, file should be written to multiple FileDataServers in parallel.
- All servers should be TCP based and implemented using event-driven architecture (I/O Multiplexing with non-blocking I/O) and threads or processes can be used to delegate "block reads/writes".
- Bigfs should be run in at least 3 physical machines or VMs. It will be tested by copying a directory (containing large files) in local file system to bigfs and using other commands.

Deliverables:

- Design Document (.pdf)
- Client.c, fileNameServer.c, FileDataServer.c

[24 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

[24M]

**--&--**