

National University of Computer and Emerging Sciences



**Operating Systems Lab
Lab Manual 5**

**Muhammad Hassan Raza
Fall 2024**

**Department of Software Engineering
FAST-NU, Lahore, Pakistan**

FIFO Pipes in C

Task 1: Message Queue Between Two Unrelated Processes

Scenario: You are building a chat system that allows communication between two unrelated processes. One process is responsible for sending messages, and the other is responsible for receiving them.

1. Create two programs, `sender.c` and `receiver.c`.
2. The sender program should continuously take input from the user (a string) and write it into a FIFO pipe.
3. The receiver program should read the messages from the FIFO pipe and display them on the screen.
4. Ensure proper synchronization between the sender and receiver programs using the named pipe.

Task 2: Real-Time Logging System

Scenario: Your system requires a real-time logging system where a process continuously logs system status to a FIFO pipe, and another process reads the log and stores it in a log file (`log.txt`).

1. The logger process should generate random system status messages every 5 seconds and write them into the FIFO.
2. The log reader process should read the logs from the FIFO and append them to `log.txt`.
3. Make sure the FIFO is read and written to continuously without any data loss.

Task 3: Client-Server Calculation System

Scenario: You are tasked with implementing a simple client-server system using FIFO pipes. The client will send mathematical expressions to the server, and the server will evaluate the expression and return the result to the client.

1. The client process should take a simple mathematical expression as input (e.g., `5 + 3`) and send it through a FIFO pipe to the server.

2. The server process should read the expression, evaluate it, and return the result through another FIFO pipe to the client.
3. The client should display the result once it is received.
4. Ensure both processes handle multiple requests and responses properly.

Task 4: Temperature Monitoring System with Multiple Sensors

Scenario: You are implementing a temperature monitoring system where three independent sensor processes report temperature data to a central monitor process via FIFO pipes.

1. Create three sensor programs (`'sensor1.c'`, `'sensor2.c'`, `'sensor3.c'`). Each sensor should simulate reading a temperature value (a random number between 20°C and 30°C) and write this value into its own FIFO pipe every 3 seconds.
2. The central monitor process should read the values from all three FIFO pipes and display the temperatures from each sensor in real-time.
3. Ensure that the monitor can handle the incoming data simultaneously from all sensors.

Note: For all tasks, remember to:

- Create the FIFO pipes using the `'mkfifo()'` function.
- Use appropriate read and write functions (`'read()'`, `'write()'`) to transfer data between the processes.
- Handle pipe creation, opening, and closing carefully to avoid deadlocks or resource leaks.
- Use `'unlink()'` to remove the FIFO pipes after completing each program.