

NETWORK PROGRAMMING

(Assignment 1)

Duration: 2 Weeks

Total Marks: 20

1. Implement a small **client-server** program using named pipes.

SERVER: Creates two fifos in /tmp directory if they do not already exist. It waits for a client request on one of the fifos and when it receives a request, it opens a log file and writes the time and userid of the client process into it. Then it reads an **activities** file, searches for the activity mentioned in the file for the time specified by the client and writes the entry into the second fifo. After writing into the fifo, it goes back to waiting for client requests.

CLIENT: Waits for user to enter date and time. Then, opens the fifo marked for writing to the server and writes the date and time entered by the user along with the current time and user-id into it. Then it reads the second fifo and displays the information read from the fifo on the screen. After the information is displayed, it exits.

DETAILS: You should create an activities file in the following format: Each line should be <Date>%<Time>%Activity Arrange the lines in chronological order so that searching is simplified. When a specific date and time are received from the client, look for the nearest time and if the time difference is less than 1 hour, retrieve the line. If the time difference is more than 1 hour, return a message saying that “No activity is scheduled.”

The client request should be in the following format:

<Current date>%<Current Time>%<user-id>%<Date>%<Time>

2. Explore the behaviour of signals across an `exec()` call.

Write a program that sets a signal handler for `SIGINT` signal. After the signal handler is set, the program forks and the parent process executes `sleep(60)`. The child process prints its PID and then executes a `sleep(60)`. Send a `SIGINT` signal to both the parent and child repeatedly.

In the second case, the child prints its PID, and then `execs` a program that executes `sleep(60)`. Again, send `SIGINT` signal repeatedly to both parent and child.