Mini Project Description

(To be submitted by 10th Apr (Sun))

Mouli Sankaran

Mini Project: Focus

- Lab Mini-Project Description: Graded
  - Edu Server and Client Application
  - To be submitted by 10th Apr (Sun)

Lab Project Description

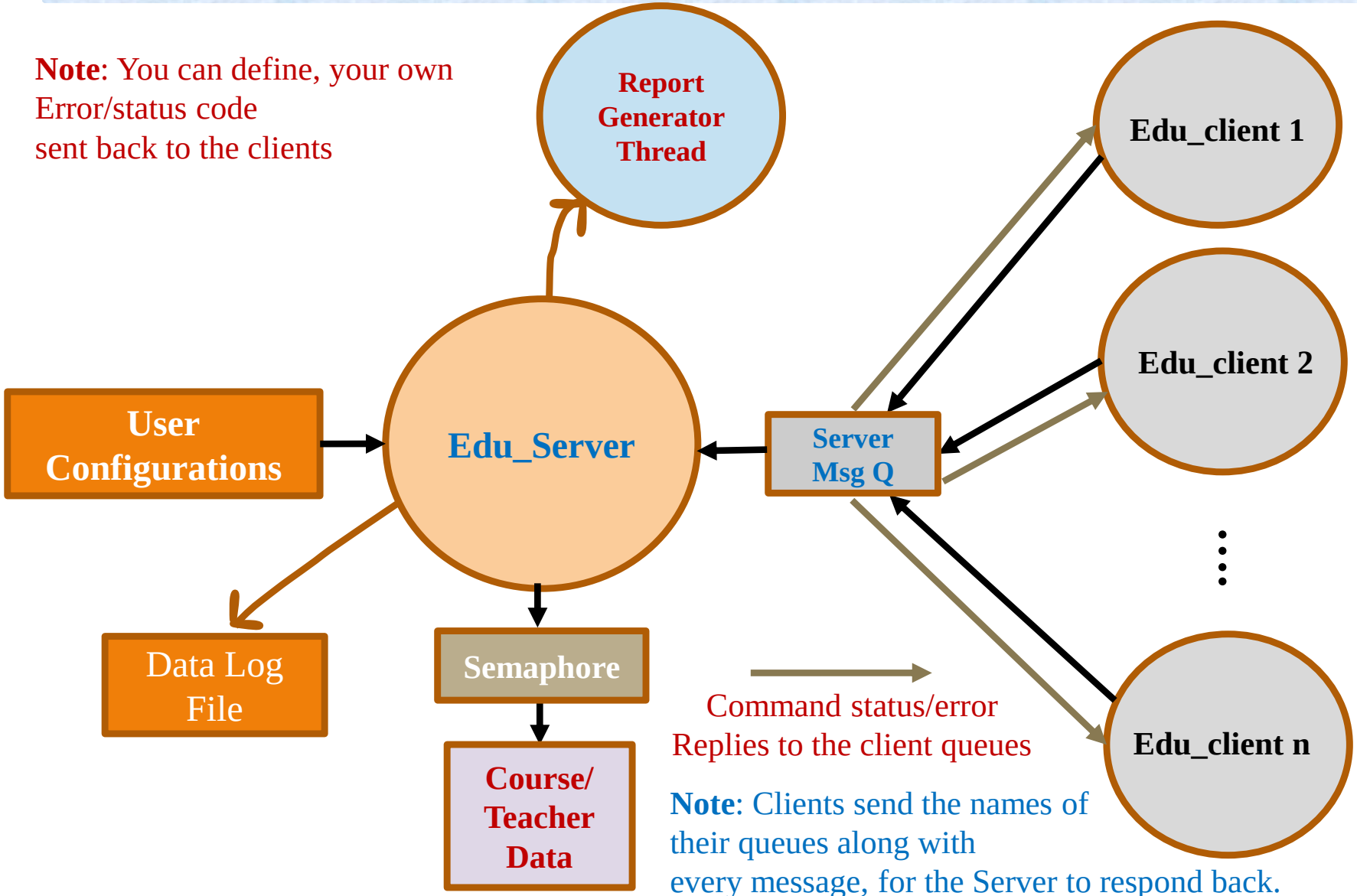
To be submitted by 10th Apr (Sun)

Note: Demo of the project and viva will be planned from the 3rd week of Apr.

1. Lab Mini Project: Edu_Server_Client

(To be submitted by 10th Apr (Sun))

Note: You can define, your own Error/status code sent back to the clients



Note: Clients send the names of their queues along with every message, for the Server to respond back.

2. Lab Mini Project: Edu_Server_Client

(To be submitted by 10th Apr (Sun))

1. Implement a multi-threaded Edu-Server and Client application in ,C using POSIX APIs such that they can be easily ported to other POSIX compliant operating systems.
2. It will accept the details from the users through the Client interface, on the Courses that are offered and the Teachers who could handle them.
3. It uses an internal logic to allocate the teachers to the courses currently being offered.
4. The Server will be maintaining the course/teachers data in the system.
5. Server will also have a periodic thread (every 10 seconds) that generates a summary of the current courses and the teachers assigned to them.
6. Semaphore will be used to protect the data while they are being accessed by the Server thread and the periodic report generator thread.
   - No GUI interface is expected to be implemented, which is optional.

3. Lab Mini Project: Edu_Server_Client

6. The Client processes can be instantiated from different terminals simultaneously, which await for the commands from the users to update the data maintained by the Server.
7. The Clients communicate with the Server through the message Q provided by the Server.
8. The Server continuously waits on the the Msg Q, for messages from various Client instances running simultaneously.
9. The User commands to the Client instances are:
   a) ADD_COURSE C1, C2, etc.
   b) DEL_TEACHER T1, T2, etc.
   c) DEL_TEACHER T1, T2, etc.
   d) DEL_TEACHER T1, T2, etc.

Note: You are encouraged to add additional commands as per your choice and need.

10. The above commands will be handled according to the available data on the Server.
11. Duplicate courses and Teachers will be ignored and error is sent back.
    - Course-Teacher allocations will be adjusted when a course or a teacher is deleted.

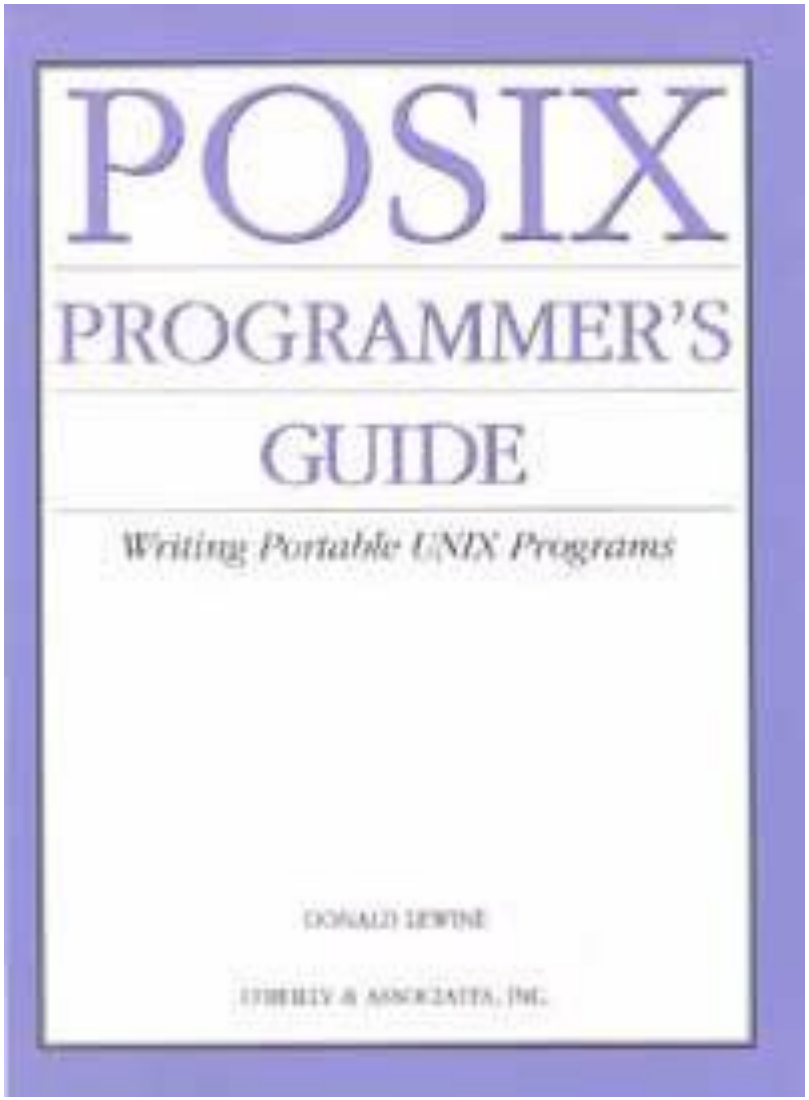
4. Lab Mini Project: Edu_Server_Client

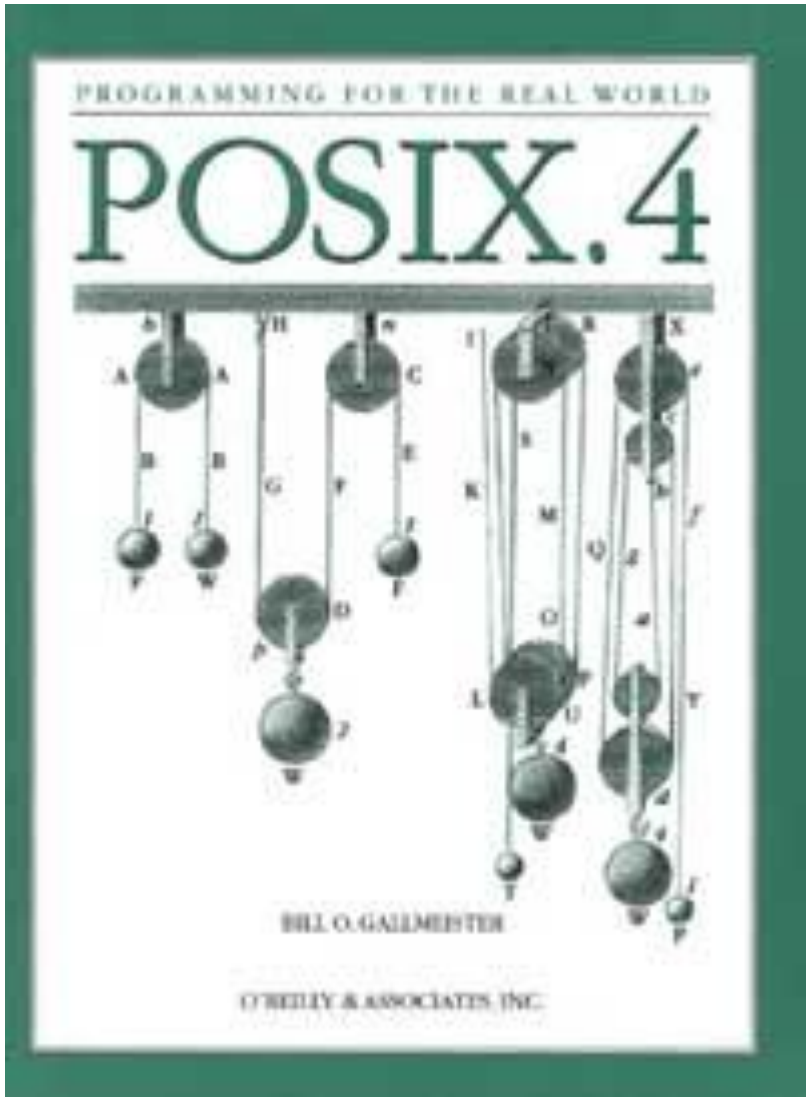
12. The initial configuration of the Edu Server with respect to the minimum and maximum values are configurable.
    A. MIN_COURSES 10
    B. MAX_COURSES 15
    C. MIN_TEACHERS 5
    D. MAX_TEACHERS 10
13. On Edu Server invocations, these above values can be given as configurable parameters, if the given values do not fall in the above limits, the above values will be chosen by the Server.
14. Clients will be populating the data once the Server is invoked with the initial configuration.
15. When the Server exits, it would save the current data into a text file.
    - Optionally, the Server could come up reading this data file when it is invoked without any configuration parameters.
16. You are encouraged to come up with suitable a data structure and additional features, if any, apart from what is mentioned here.

Mini Project: Summary

- Lab Mini-Project Description: Graded
  - Edu Server and Client Application
  - To be submitted by 10th Apr (Sun)

References


POSIX
PROGRAMMER'S
GUIDE
Writing Portable UNIX Programs
DONALD LEWINE
O'REILLY & ASSOCIATES, INC.


PROGRAMMING FOR THE REAL WORLD
POSIX.4
BILL O. GALLMEISTER
O'REILLY & ASSOCIATES INC.