

Exercise 6

- create a directory labs/lab06 and add it to your repository.
- Make sure all code you create for this lab/project is added and checked into this directory of your repository.
- Your project must be built to compile and run on our nisl Linux machine. Extra credit if you demonstrate using make to do some automated building, testing or execution of options of your project.

Exercise 6 - Project

- Implement a simple version of a networked program. You may use the socket library/ language of your choice (for example, C, Python, Java are all acceptable).
- There are several possibilities:
 1. Implement a concurrent server. You can start from the concurrent server code we used and discussed in class. Implement one of the simple client/server applications that usually requires concurrence from the Project Proposal ideas. These would include:
 1. a basic database allowing multiple clients to create tables, query tables and drop tables
 2. A small multi-player game that allows 2 or more players to compete. Examples might be a card game, or even tic-tac-toe.
 3. A simple file transfer server that allows multiple clients to connect and to upload and download files to some file transfer area
 4. others are also possible.
 2. For those interested in other types of distributed applications. Consider using the simple xmlrpcserver library from python to implement a small application using remote procedure calls. These could even be one of the applications listed in #1 above.
 3. Implement a basic http web server. You can start with the Python HttpServer library, which gets you pretty much 90% of the way to implementing a web server. You should add code so that different pages can be requested from you server, etc.