Midterm: Dec. 17th

Goal: Write a “Scoreboard Keeping” client-server application.

- Server maintains a <Cell-Value> data structure (e.g. an array)
- Clients connect and update cells, using the following message set:
  - **Lock** <Cell>, returns (OK, NOK)
  - **Unlock** <Cell>, returns (OK, NOK)
  - **Get** <Cell>, returns <Current Value of Cell>
  - **Set** <Cell> <NewValue>, returns (<OldValue>, NOK)

Server Flow:

- Creates a shared memory segment for the of array (size N defined in include file)
- Creates semaphores
- Initializes (cell values, semaphores, ..)
- Creates a socket,bind(port number defined in include file),listen
- Loops on accept()
    - Forks a new child to handle the connection
- close()/exit()

Child Flow:

- Loop
    - Receives a request from client (read from socket)
    - Performs the operation (lock/get/set/unlock)
    - Returns result to client (write to socket)
- close()/exit()

Client Flow (to test the server):

- Gets #iterations and server name as command line arguments
- Connects to server/port (port defined in include file)
- Loop #iterations
    - Creates a random number <J> in range [0..N), (N size of array, read from include file)
    - Increments the value of Cell#<J> using combination of [lock/get/set/unlock]
    - Checks for errors (e.g. value returned from a “set” is not what expected)
- shutdown()/close()/exit()