Homework 1

Policies:

- The homework is designed to be done by a group of two students.
- You are free to discuss the homework and how to divvy up the work between you and your partner, however, for maximum benefit, both partners need to be aware and participate in solving *every* problem.
- Inter-group discussions are encouraged, but each group must write its own solution. Copying solutions between groups is considered cheating and will be dealt with according to the policy announced in the first lecture of the class.
- Using material from the Internet including text and code are permitted provided that the right attribution is made, and that the student understands thoroughly what is being used.
- Project due by 5PM on September 18, 2023.

Advice

Start early!!

Questions:

1. Consider the famous qsort() algorithm for a very large dataset. What would you consider the bottleneck in the system performance to be from a qualitative viewpoint?

2. The GUPS benchmark can be summarized in the following simplified code:

   ```
   i = random(1, 1<<24);    // Generate a random
                            // integer between 0 and
                            // 1<<24
   hist[i]++;               // i and hist are of
                            // type integer
   ```

   The L1 cache is 32KB, and the L2 cache is 2MB. How do you expect the code to perform? Would it benefit from adding another layer of caching of 16MB size?

3. Using the function random() (man -s 3 random()), write a uniform number generator between 1 and *n* (you can use C or C++).

4. Write a program to generate a Poisson distribution of a mean value of *l.*

5. Consider a Web server that receives requests with an inter-arrival time defined by a Poisson distribution with an average of five milliseconds. Ninety per cent of all requests can be served within a time interval that is defined by a uniform distribution between three to twenty milliseconds, whereas the remaining ten percent of requests can be served within a time interval defined by a uniform distribution between two hundreds to one second.

    a. Draw a simple queuing model for this problem.
    b. Use an event-driven simulation to implement the queueing model in (a). Use the simulation to estimate the average and standard deviation of the response time of the overall requests. Segregate the measurements for the two types of requests. Estimate the average utilization of the server.
    c. If we would like to have the average response time to be thirty milliseconds and a standard deviation of +/- 10%, estimate the number of necessary servers. Use a single queue/multiple server paradigm.
    d. Repeat (c) but use multiple queues with one server per queue. Use a shortest-queue algorithm for assigning requests (measured by the number of requests).
    e. Repeat (d) but use a round-robin queueing algorithm.
    f. Repeat (c) with two queues, one for the short requests and one for the large requests. We would like the short request response time to be 15 milliseconds on average, whereas the long requests to be 800 milliseconds on average. Estimate the number of server and estimate the average utilization of servers per each category.
    g. Compare the results from (b) through (f). Comment.

    For this problem, you need to repeat the simulation several times with different random seeds to gain confidence in your numbers.