Homework 2 for CS152 (Fall 2018)

Due Tuesday 10/30, 2018

1. What is a thread? What does a thread contain?

2. The function array_sum below computes a sum of an array. Please write a version that uses multiple threads to compute the sum in parallel. (Hint: use pthread_create and pthread_join in the pthread library).

```
double array_sum(double *arr, int len) {
  double sum = 0;
  for(int i=0; i<len; i++)
    sum += arr[i];
  return sum;
}
```

3. Two players play table tennis. The first player serves, the second player hits the ball backs to first player, and first player hits back, and with a 20% probability, one player may drop the ball. Write a program to simulate this situation. Each player runs in one thread. (hint: use pthread_create and sem_init, sem_wait, sem_post. Use a random number generator to determine if a player drops the ball.) A possible output looks like this:

```
Player 1 serves.
Player 2 hits it back to Player 1.
Player 1 hits it back to Player 2.
Player 1 drops the ball.
```

4. A group of 5 people go to a restaurant. They wait until the last person arrives before they start ordering. Implement this scenario using threads and semaphores. (hint: use pthread_create, sem_init, sem_wait, sem_post.)

5. In lec04.pdf, there is an example of web server using multi-threading. It creates a new thread to serve every request. Suppose you like to limit the resource consumption by allowing no more than 100 active threads simultaneously, how do you modify the code to realize this limit? (Hint: use semaphore(s). pseudo code is enough.)

```
web_server() {
  while (1) {
    int sock = accept();
    thread_create(handle_request, sock);
  }
}

handle_request(int sock) {
  Process request
  close(sock);
}
```