COMP 2004 - Fall 2021
Instructor: Dr Vinicius Prado da Fonseca
Assignment 3

1. (10%) Using Amdahl’s Law (see Computer Architecture and Amdahl’s Law pdf in the course contents), calculate the speedup gain for the following applications:
   a. 40 percent parallel with
      i. eight processing cores
      ii. sixteen processing cores
   b. 67 percent parallel with
      i. two processing cores
      ii. four processing cores
   c. 90 percent parallel with
      i. four processing cores
      ii. eight processing cores

2. (40%) Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers:

   ```
   ./a3q2 90 81 78 95 79 72 85
   ```

   The program will report:

   ```
   The average value is 82
   The minimum value is 72
   The maximum value is 95
   ```

   The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the parent thread will output the values once the workers have exited. (We could obviously expand this program by creating additional threads that determine other statistical values, such as median and standard deviation.)

3. (50%) Programming problem 4.28 from the 10th edition of the text with modifications.

   Modify programming problem Exercise 3.20 from Chapter 3 (Assignment 2 Question 2). You will create a number of threads—for example, 100—and each thread will request a pid, sleep for a random period of time, and then release the pid. (Sleeping for a random period of time approximates the typical pid usage in which a pid is assigned to a new process, the process executes and then terminates, and the pid is released on the process's termination.) On UNIX and

Linux systems, sleeping is accomplished through the sleep() function, which is passed an integer value representing the number of seconds to sleep. This problem will be modified in Chapter 7.

Modifications: The number of threads to create as well as the maximum sleep time will be passed on the command line as:

```
<program name> <number threads> <maximum sleep>
```

For example:

```
./a3q3 100 15
```

This represents 100 threads, each with a random sleep time of up to 15 seconds.