Zagazig University
Faculty of Engineering
Computer and Systems Dept.

Spring 2020
Third Year
CSE325: Operating Systems

Discussion #4

Covers: Chapter 4

Questions:

1. Using Amdahl's Law, calculate the speedup gain of an application that has a 60 percent parallel component for
    a. two processing cores
    b. four processing cores
2. (Choose) The speedup gain of an application that has … percent parallel component and … processing cores is 2.5.
    a. 0.25, 4
    b. 0.75, 15
    c. 0.25, 5
    d. 0.75, 5
3. What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?
4. Which of the following components of program state are shared across threads in a multithreaded process?
    a. Register values
    b. Heap memory
    c. Global variables
    d. Stack memory
5. Is it possible to have concurrency but not parallelism? Explain.
6. Consider the following code segment:

```
int main()
{
   pid t pid;
   pid = fork();
   if (pid == 0) { /* child process */
         fork();
         thread create(. . .);
   }
   fork();
}
```

    a. How many unique processes are created?
    b. How many unique threads are created?
7. The program shown Below uses the Pthreads API. What would be the output from the program at LINE C and LINE P?

```
#include <pthread.h>
#include <stdio.h>
int value = 0;
void* runner(void* param); /* the thread */
```

```
int main(int argc, char* argv[])
{
   pid t pid;
   pthread t tid;
   pthread attr t attr;
   pid = fork();
   if (pid == 0) { /* child process */
        pthread attr init(&attr);
        pthread create(&tid, &attr, runner, NULL);
        pthread join(tid, NULL);
        printf("CHILD: value = %d", value); /* LINE C */
   }
   else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d", value); /* LINE P */
   }
}
void* runner(void* param) {
   value = 5;
   pthread exit(0);
}
```

Programming Questions:

1. 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

   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.