CSCI 3453: Operating System Concepts
HW Assignment # 3
Due Date: February 21, 2021 @ 11:55 PM

1) Is it possible to have concurrency but not parallelism? Explain.
   Concurrent: Dealing with lots of things at once. Execution of threads will be woven with time
   Parallelism: Doing lots of things at once

   Possible to be concurrent without parallelism (usually on single core processors) If there are lots of threads happening on a single core, they can be scheduled in succession. The threads cannot run at the same time. However if there are multiple cores, two different threads can run at the same time. Assuming that threads don't rely on each other for data

2) Using Amdahl's Law, calculate the speedup gain for the following applications:
   1) 40 percent parallel with (a) eight processing cores and (b) sixteen processing cores = 60% serial
      - a) 1.538
      - b) 1.6
   2) 67 percent parallel with (a) two processing cores and (b) four processing core =33 serial
      - a) 1.5037
      - b) 2.010
   3) 90 percent parallel with (a) four processing cores and (b) eight processing cores
      - a) 3.07692
      - b) 4.70588

3) Using the program shown below, explain what the output will be at LINE A.

```
#include<sys/types.h>
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>

 int value = 5;
int main()
{
   pid_t pid;
   pid =fork();
   if (pid == 0) { /* child process */
      value += 15;
      return 0;
   }
   else if (pid > 0) { /* parent process */
      wait(NULL);
      printf("PARENT: value = %d",value); /* LINE A */
      return 0;
   }
}
```

Line A = PARENT: value = 5

When fork() is called, the child and parent share the same value at first and then goes through the if/else statement where their values will change. Child will now have a value of 20, Parent has a value of 5, but since we are only printing the value of the parent, Like A = 5.

4) Including the initial parent process, how many processes are created by the program below:

```
#include <stdio.h>
#include <unistd.h>
int main()
{
   /* fork a child process */
   fork();
   /* fork another child process */
   fork();
   /* and fork another*/
   fork();
   return 0;
}
```

Would be 8, since each fork creates a copy of the child and a fork is 2^n

5) Describe the actions taken by a kernel to context-switch between processes.

When an interrupt occurs, it causes the OS to change its CPU core from its current task to run a kernel routine. System needs to save it's current context in the PCB so that it can be restored when the process is done. When it is done, it performs a state restore to bring back the process to be continued. (pg 114)

6) Provide three programming examples in which multi threading provides better performance than a single-threaded solution.
   - Weather App on smart phone (Thread for updating display and another for downloading data)
   - Microsoft word with multiple threads within itself (a thread for displaying graphics, another thread for responding to keystrokes from the user, and a third thread for performing spelling and grammar checking in the background.)
   - Google chrome with 20+ tabs open running efficiently (one thread display images or text while another thread retrieves data from the network.) pg160

7) Which of the following components of program state are shared across threads in a multi threaded process?
   1) Register values - no
   2) Heap memory - yes
   3) Global variables - yes

    4) Stack memory - no

8) Determine if the following problems exhibit task or data parallelism:
    1) Using a separate thread to generate a thumbnail for each photo in a collection
        - Data parallelism
    2) Transposing a matrix in parallel
        - Data parallelism
    3) A networked application where one thread reads from the network and another writes to the network
        - Task parallelism
    4) The Grand Central Dispatch system
        - Data parallelism

9) Consider the following code segment:

```
pid_t pid;
pid =fork();
if (pid == 0) { /* child process */
 fork();
 thread_create( . . .);
}
fork();
```

1) How many unique processes are created? 5
2) How many unique threads are created? 2

Create the text file using Microsoft Word, Open Office Writer, or a text editor. Name the document as follows:
FirstName_LastName_HW3

Rubric for Grading:

Description	**Points**
Question 1	10
Question 2	30
Question 3	10
Question 4	10
Question 5	10
Question 6	10
Question 7	20
Question 8	20
Question 9	20
Total Points	**140**