COSC1112/1114: Operating Systems Principles

Tutorial 03 (week 04)

1. Provide two programming examples in which multithreading does not provide better performance than a single-threaded Solution.

2. Is it possible to have concurrency but not parallelism? Explain.

3. Using Amdahl's Law, calculate the speedup gain of an application that has a 60 precents parallel component for
   (a) two processing cores
   (b) four processing cores.

4. Which of the following components of program state are shared across threads in a multithreaded process?
   - Register values
   - Heap memory
   - Global variables
   - Stack memory

5. Linux does not distinguish between processes and threads. Instead, Linux treats both in the same way, allowing a task to be more akin to a process or a thread depending on the set of flags passed to the clone() system call. However, many operating systems— such as Windows or Solaris—treat processes and threads differently. Typically, such systems use a notion wherein the data structure for a process contains pointers to the separate threads belonging to the process. Contrast these two approaches for modelling processes and threads within the kernel.

6. The program shown below uses the Pthreads API. What would be the output from the program at LINE C and LINE P?

```
#include <stdio.h>
#include <unistd . h>
#include <stdlib.h>
#include <pthread . h>

int value = 0;

void* runner ( void* param) {
        value = 5;
        pthread_exit ( 0 ) ;
}

int main ( ) {
        int pid ;
        pthread_t t id ;
        pthread_attr_t at t r ;

        pid = fork ( ) ;
        i f ( pid == 0) {
                pthread_attr_ini t (&at t r ) ;
                pthread_create(&tid ,&attr , runner ,NULL) ;
                pthread_join (tid ,NULL) ;
                print f ( "CHILD: v = %d" , value ) ; /* LINE C */
        } else if (pid >0) {
                wait (NULL) ;
                printf ( "PARENT: v = %d" , value ) ; /* LINE P */
        }
        return EXIT_SUCCESS;
}
```

7. Google's Chrome browser is designed to open each new website in a separate process. Would the same benefits have been achieved if instead Chrome had been designed to open each new website in a separate thread? Explain.

8. Consider a multiprocessor system and a multithreaded program written using the many-to-many threading model. Let the number of user-level threads in the program be more than the number of processors in the system. Discuss the performance implications of the following scenarios.
   - The number of kernel threads allocated to the program is less than the number of processors.
   - The number of kernel threads allocated to the program is equal to the number of processors.
   - The number of kernel threads allocated to the program is greater than the number of processors but less than the number of user-level threads.