Operating Systems	November 24, 2012
CSC 227	Fall 2012/2013
Homework 4 (A Sample Solution)	Points: 20
Due: December 01, 2012: 2:00 pm	

Note: A handwritten solution is unacceptable.

1. [4.0 points] What is the difference in the CPU's structure in the following two cases: Multiprogramming processes, Multiprogramming threads.

Answer:

When a thread is stopped, it has values in the registers. They must be saved, just as when the process is stopped the registers must be saved. Multiprogramming threads is no different than multiprogramming processes, so each thread needs its own register save area.

2. [4.0 points] Provide two programming examples in which multithreading provides better performance than a single-threaded solution?

Answer:

1. A web server that services each request in a separate thread.
2. A web browser with separate threads playing sound, downloading a file, collecting user input, etc
3. A word processor with a thread to save, at regular intervals, the file that is currently being executed, and another thread a spell-checker, etc
4. An application such as matrix multiplication where each row of the matrix product is evaluated, in parallel, by a different thread.

3. [2.0 points] What's the difference between many-to-many and two-level models?

Answer:

The two level model allows more flexibility by embedding two models into one: many-to many and one-to-one models.

4. [2.0 points] Describe two ways for implementing a thread library?

Answer:

- The library involves ensuring that all code and data structures for the library reside in user space with no kernel support.

- Implementation a kernel-level library supported directly by the operating system so that the code and data structures exist in kernel space

5. [4.0 points] What is the difference between a process and a thread? Which one consumes more resources?

Answer:

- A process defines the address space, text, resources, etc. A thread defines a single sequential execution stream within a process.
- Threads are bound to a single process. Each process may have multiple threads of control within it.
- It's much easier to communicate between threads than between processes.
- It's easy for threads to inadvertently disrupt each other since they share the entire address space.
- Process consumes more resources

6. [4.0 points] What is a task in Linux OS? Can a thread creation be done using fork() system call in Linux?

Answer:

In Linux, a task is a process or thread.

No, thread creation is done using clone() system call. The child and the parent tasks share the parent address space.