

CSCI 315 Operating Systems Design
Fall 2014 - Prof. Felipe Perrone
Activity 12

Work on this activity with one, or at most, two partners.

1) Create a scenario to illustrate the concept of *concurrency*.

2) Create a scenario to illustrate the concept of *parallelism*.

3) You have two threads that share an integer variable *a*, which has been initialized with value 0. Consider the two programs below, which promise to guarantee that two threads that access and increment this variable with different increments produce a *deterministic* result of 23. Compare the two programs and discuss which one is *better* (be sure to explain your criteria for “better”).

PROGRAM A

mutex m;
init(&m, 0);

Thread 1	Thread 2
a = a + 6; signal(&m);	wait(&m); a = a + 17;

PROGRAM B

mutex m;
init(&m, 1);

Thread 1	Thread 2
wait(&m); a = a + 6; signal(&m);	wait(&m); a = a + 17; signal(&m);

4) Compare the concurrency constructs *process* and *thread* from the perspective of run-time performance and memory utilization. Is the use of one clearly better than the other? Does context switching happen in only one or in both?

5) Which process scheduling policies can lead to *starvation*? What aspects of these policies can cause this type of problem?

6) When we talk about solutions to the *critical section* problem, we understand they need to satisfy *mutual exclusion*, *progress*, and *bounded-waiting*. Explain what you understand by progress and by bounded-waiting (give examples, if you like).