Aisha Jackson

November 20, 2014

Concurrency Questions

1. Concurrency can occur at the instruction level, statement level, unit level, and program level. The instruction level executes two or more machine instructions simultaneously. The statement level executes two or more high level language statements simultaneously. The unit level executes two or more subprogram units simultaneously and the program level executes two or more programs simultaneously.
2. Concurrency should be used because multiprocessor computers capable of physical concurrency are now widely used. Also, a program written to use concurrent execution can be faster than the same program written for non-concurrent execution. Lastly, many program applications are now spread over multiple machines.
3. Physical concurrency is the most natural category of concurrency and several program units from the same program literally execute simultaneously. Logical concurrency is when the actual execution of programs is taking place in interleaved fashion on a single processor.
4. Co-routines have a single thread of control which is the sequence of program points reached as control flows through the program.
5. At the subprogram level, we may have a task or process or thread.
6. Lightweight tasks all run in the same address space while heavyweight tasks execute in its own address space.
7. Cooperation synchronization is when a task must wait for another task to complete some specific activity before the task can continue its execution. Competition synchronization is when two or more task must use some resource that cannot be simultaneously used.
8. The possible task execution states are new which is when a task is in the new state and has been created but has not begun its execution. A ready task is ready to run but is not currently running. A running task is one that is currently executing. A blocked task is a task that is blocked but has been running and the execution was interrupted by one of several different events. A dead task is no longer active.
9. Liveness is a characteristic that a program unit may or may not have. It means the unit will eventually complete its execution. If all tasks in a concurrent environment lose their liveness, it is called deadlock.
10. Three methods for synchronization are semaphores, monitors, and message passing.
11. Dijkstra devised semaphores to provide competition synchronization through mutually exclusive access to shared data structures.
12. A third semaphore called access is used to control access in competition synchronization. However, wait and release must be atomic.
13. Monitors are used to encapsulate the shared data and its operations to restrict access. A monitor is an abstract data type for shared data.
14. Message passing is a general model for concurrency. Message passing can be either synchronous or asynchronous. Synchronous message passing is when tasks are busy and they cannot be interrupted by other units.
15. The execution of a task is completed if control has reached the end of its code body. This may occur if an exception has been raised for which there is no handler.
16. Java’s threads are the process in which the run methods can execute. The thread class has five constructors and collection of methods. Semaphores in Java have a class that has two methods, acquire and release which corresponds to the wait and release operations.
17. High performance FORTRAN is a collection of extensions that are meant to allow programmers to specify information to the compiler to help it optimize the execution of programs on multiprocessor computers. HPF includes ne specification statements and intrinsic or built in subprograms.