

31/7/2020

Multithreading

Basics of Java Threading

↳ Creating, Managing & Monitoring threads - 1

Concurrency → running several things at the same time.

Segmentation: processes can't access memory & resources.

Threads → share memory & resources.

Time slicing allow threads to run simultaneously.

Simultaneously.

Memory accesses → slow b/c CPU & memory bus.

Standard threads runs

Exception: daemon thread.

Sleeping → Use loop to wait for period to expire.

↳ doesn't block.

platform dependent

- sleep(0) may (or) may not sleep

- yield(0) may (or) may not yield.

Default Thread Group Handler:

↳ check if default Uncaught Exception handler has been set.

Handling group of threads.

→ Setting the priorities of threads

Inter Thread Communication:

↳ Threads need to share their data more involved than the variables

Code Reordering → Compiler, JVM, Preprocessor

Hesenberg

Synchronizing data → volatile data.

↳ Any thread can make variables as volatile.

Memory Fence:

↳ Type of Barrier instruction that causes a CPU / compiler to enforce ordering constraint on memory operations issued before & after Barrier instruction.

Mutual Exclusion → Allows one thread to access critical section at once

→ Built in Synchronized gives mutual Exclusion.

→ Prevent Concurrent access to resources

→ One (or) more parts of code which may not be accessed by more than one thread at a time.

Mutex → Short for mutual Exclusion object.

Mutual Exclusion → Synchronised keyword.

↳ to use project Invariants

Three liveness issues:

Deadlock → Threads cannot make progress.
Livelock → No thread acquires all mutexes it requires.
Starvation → Deadlocks don't have to involve just mutexes.

Livelock → similar to deadlock, except that states of processes involved is livelock.

Starvation → when threads are not getting enough execution time to carry out their tasks

Waiting requires a condition \rightarrow Wait Set

IllegalMonitorState Exception

↳ Forgetting to Synchronize

↳ Synchronizing on wrong object.

notifyAll() \rightarrow Causes Thread to request reacquire monitor.

\rightarrow Passing Tasks to Thread Pools

Task \rightarrow Runnable starts at run()

Adapting Callable to Runnable.

CallerRunsPolicy

↳ If not shutdown, Caller Thread runs instead.