Spurious Wake up of a Thread:

A spurious wakeup is a phenomenon that can occur in concurrent programming, particularly in multithreaded environments, where a thread wakes up from waiting even though there was no explicit signal or event that caused it to do so.

In some cases, the waiting thread may be awakened by the underlying system or hardware for reasons other than the expected event or signal. This can happen due to various factors such as implementation details, timing issues, or hardware glitches.

While spurious wakeups are relatively rare, they can cause issues in concurrent programs if they are not handled properly. For example, if a thread is waiting for a specific condition to be met and is awakened spuriously, it may incorrectly assume that the condition has been met and proceed with its execution, leading to unexpected behavior or bugs.

To mitigate the risk of spurious wakeups, it is common to use a loop around the wait condition to check if the condition is really met before proceeding. This is sometimes called a "looping wait" or a "guarded wait". By using a looping wait, the thread can recheck the condition and continue waiting if the wakeup was spurious, ensuring that it does not proceed with its execution prematurely.

In summary, a spurious wakeup is an event where a thread wakes up from a wait state even though no explicit signal or event occurred. It is a rare occurrence but can cause issues in concurrent programs if not handled correctly.