Concurrency Bugs

Mutual Exclusion Violations

Race Condition

Two or more threads access shared data without proper synchronization, and the result depends on timing.

Analogy

Two workers fight over each other to count up on a board. But they often overwrite their own answers.

Atomicity Violation

A multi-step operation is interrupted by another thread.

Example

Thread A checks a flag before then using it, but Thread B changes the flag while being used.

Analogy
A worker signals to his colleagues that the single-person bathroom is being used and locked the door.

Another worker comes in and opens the lock thinking that its *his personal lock*.

Other workers are forced to see the man in horror and he was interrupted doing his dirty business.

Liveliness Violations

Check Synchronization primitives first to get the analogies here.

Deadlock

Two or more threads wait forever for resources locked by each other.

Analogy

A debate club wants to coordinate who will talk first, so they asked the chicken to have coordination between them.

> "Whoever has me shall talk. Else, they'll wait for their turn at had. Once you are done talking, pass it over to the next one who wants to speak."
>
> - The Chicken.

Many debates went over. The chicken is passed on to the next worker and to the next. Until one time, a student passed out.

The thing is, they never gave away the chicken. The students are steadfast obeying the rule to not speak, so no one can suggest to pass it along to next.

The students are forced to sit in silence forever until recess comes.

The chicken here is the **mutex**, the workers are **threads**.

Livelock

Threads are *not blocked*, but keep responding to each other in a way that prevents progress (like two people in a hallway both stepping aside endlessly).

Analogy

Once someone got the chicken back from the passed out student, they begun to speak. However, two people pleasers are next in line.

The first one said:

> "Okay, you can have the chicken. I think you're more important to talk."

The second said:

> "No, you can have it! I'm not as important as you are"

They keep debating who should have the chicken, stuck in an endless loop.

Starvation

A thread never gets CPU time or access to a resource because others always take priority.

Analogy

While the two bicker over who gets the rightful owner of the chicken, the other workers are starved to speak. They get super irritated by the two not giving the chicken to those of lower order.

They know they'll never get to taste speaking, when two have hands over the chicken.

Ordering Violations

Order Violation

An operation happens earlier/later than intended.

Analogy

A chef cook meal before the customer even starts ordering it.

Memory Consistency Errors

Even with the use of Synchronization primitives such as Mutexes, it only guarantees that one threads gets exclusive rights to execute a critical section of code.

However, it does not guarantee that memory can be consistent across caches.

You need **Atomics and memory barriers** to enforce consistent memory across caches.

Analogy

A Record keeper can keep track of minutes of a meeting of each office.

However, the record keepers also need to coordinate with each other to see what decision should get transmitted and affect the other offices' minutes of the meeting.

There, a random sheep provided a solution: a walkie talkie that relays that new information to write.

The **walkie talkie** of the sheep is the use of atomic variables.

Performance Violations

Excessive Lock Contention

When many workers try to acauire a lock at the same time, they are forced to wait and take turns. Having too many locks and workers may force many threads to sleep.

Analogy

There's a bathroom that many workers used, there's a queue that forms as one waits for the worker inside to finish their business.

They wait so long that going to the comfort room takes more time than working.

Priority Inversion

A low-priority thread excessively holds a lock needed by a high-priority thread, while a medium-priority thread keeps running.

Analogy

When you run a restaurant, the servers often gives priority to listen to a random lowly duck than to a five-star Michelin restaurant critic.

They still sever other mediocre customers their food, but the duck has exclusive rights to get their orders written more often than not.