

# Starvation

---

Starvation occurs when one thread is unable to obtain the resource it needs, to execute.

This is usually caused by other concurrent threads being greedy.

Some threads are able to make progress, but others can't.

This means your application may still keep running, and some of the work is getting done, but not all of it.

# Fair Locks

---

A fair lock guarantees that all threads waiting to acquire the lock will be given an equal chance of acquiring it.

This is in contrast to an unfair lock, which doesn't make any guarantees.

Remember, the monitor lock is unfair.

A Reentrant lock can be fair or unfair.

# How Fair Locks Work

---

When a thread requests access to a fair lock, it gets added to a FIFO queue.

The lock is then granted to the thread at the head of the queue, or the first in.

# Should you use a fair lock?

---

## Benefits:

- Fair locks can help to prevent thread starvation.
- They may improve the overall performance of a system, by ensuring that all threads get a chance of accessing resources.
- They can make a system more predictable and easier to debug.

## Drawbacks:

- Fair locks might have a negative impact on performance, especially in systems where threads are frequently competing for locks.
- Fair locks can be more difficult to implement.