

Aging (scheduling)

In Operating systems, **aging** (US English) or **ageing** is a scheduling technique used to avoid starvation. Fixed priority scheduling is a scheduling discipline, in which tasks queued for utilizing a system resource are assigned a priority each. A task with a high priority is allowed to access a specific system resource before a task with a lower priority is allowed to do the same. A disadvantage of this approach is that tasks assigned with a lower priority may be starved when a large number of high priority tasks are queued. Aging is used to gradually increase the priority of a task, based on its waiting time in the ready queue.

Contents

Problem

Application

Example

References

Problem

In priority-based scheduling algorithms, a major problem is indefinite block, or starvation. A process that is ready to run but waiting for the CPU can be considered blocked. A priority scheduling algorithm can leave some low-priority processes waiting indefinitely. A steady stream of higher-priority processes can prevent a low-priority process from ever getting the CPU.^[1]

Application

Aging is used to ensure that jobs with lower priority will eventually complete their execution. This technique can be used to reduce starvation of low priority tasks.^[2] There are many ways to implement aging, but all have the same principle that the priority of a process should increase as it waits in the ready queue. The increase in priority may or may not be equal to the waiting time of the process.

Example

Suppose a system with priority range of 0-512. In this system, 0 means highest priority. Consider a process with priority 127. If we increase its priority by 1 every 15 minutes, then in more than 32 hours the process will age to 0 priority and get executed.

References

1. Silberschatz, Galvin, Gagne *Operating System Principles*, 7th ed., p.159
2. Processor Scheduling - Notes - Operating Systems - Computer Science Now (<http://www.comsci.us/os/notes/ch08.html>)