WIKIPEDIA

Multilevel feedback queue

In computer science, a **multilevel feedback queue** is a scheduling algorithm. *Scheduling algorithms* are designed to have some process running at all times to keep the central processing unit (CPU) busy.[1] The *multilevel feedback queue* extends standard algorithms with the following design requirements:

1. Separate processes into multiple *ready queues* based on their need for the processor.
2. Give preference to processes with short CPU bursts.
3. Give preference to processes with high I/O bursts. (I/O bound processes will sleep in the *wait queue* to give other processes CPU time.)

The *multilevel feedback queue* was first developed by Fernando J. Corbató (1962). For this accomplishment, the Association for Computing Machinery awarded Corbató the Turing Award.[2]

Process scheduling

Whereas the multilevel queue algorithm keeps processes permanently assigned to their initial queue assignments, the *multilevel feedback queue* shifts processes between queues.[3] The shift is dependent upon the CPU bursts of prior time-slices.[4]

- If a process uses too much CPU time, it will be moved to a lower-priority queue.
- If a process is I/O-bound or an interactive process, it will be moved to a higher-priority queue.
- If a process is waiting too long in a low-priority queue and starving, it will be aged to a higher-priority queue.

Algorithm

Multiple FIFO queues are used and the operation is as follows:

1. A new process is inserted at the end (tail) of the top-level FIFO queue.
2. At some stage the process reaches the head of the queue and is assigned the CPU.
3. If the process is completed within the time-slice of the given queue, it leaves the system.
4. If the process voluntarily relinquishes control of the CPU, it leaves the queuing network, and when the process becomes ready again it is inserted at the tail of the same queue which it relinquished earlier.
5. If the process uses all the quantum time, it is pre-empted and inserted at the end of the next lower level queue. This next lower level queue will have a time quantum which is more than that of the previous higher level queue.
6. This scheme will continue until the process completes or it reaches the base level queue.
   - At the base level queue the processes circulate in round robin fashion until they complete and leave the system. Processes in the base level queue can also be scheduled on a first come first served basis.[5]
   - Optionally, if a process blocks for I/O, it is 'promoted' one level, and placed at the end of the next-higher queue. This allows I/O bound processes to be favored by the scheduler and allows processes to 'escape' the base level queue.

For scheduling, the scheduler always starts picking up processes from the head of the highest level queue. Only if the highest level queue has become empty will the scheduler take up a process from the next lower level queue. The same policy is implemented for picking up in the subsequent lower level queues. Meanwhile, if a process comes into any of the higher level queues, it will preempt a process in the lower level queue.

Also, a new process is always inserted at the tail of the top level queue with the assumption that it will complete in a short amount of time. Long processes will automatically sink to lower level queues based on their time consumption and interactivity level. In the multilevel feedback queue a process is given just one chance to complete at a given queue level before it is forced down to a lower level queue.

Scheduling parameters

In general, a multilevel feedback queue scheduler is defined by the following parameters:[5]

- The number of queues.
- The scheduling algorithm for each queue which can be different from FIFO.
- The method used to determine when to promote a process to a higher priority queue.
- The method used to determine when to demote a process to a lower priority queue.
- The method used to determine which queue a process will enter when that process needs service.

External links

- Multilevel Feedback Queue Schedulers — Solaris 2.6 Time-Sharing (https://pages.cs.wisc.edu/~remzi/solaris-notes.pdf)
- Processor Sharing Queueing Models of Mixed Scheduling Disciplines for Time Shared System (https://www.lk.cs.ucla.edu/data/files/Kleinrock/Processor%20Sharing%20Queueing%20Models%20of%20Mixed.pdf)

See also

- Lottery scheduling
- Fair-share scheduling
- Round-robin scheduling

References

1. *Silberschatz, Abraham (1994). Operating System Concepts, Fourth Edition. Addison-Wesley. p. 131. ISBN 978-0-201-50480-4.*

2. *Arpaci-Dusseau, Remzi H.; Arpaci-Dusseau, Andrea C. (2014). "Multi-level Feedback Queue". Operating Systems: Three Easy Pieces (https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-sched-mlfq.pdf) (PDF). Arpaci-Dusseau Books.*

3. *Silberschatz, Abraham (1994). Operating System Concepts, Fourth Edition. Addison-Wesley. p. 147.* *ISBN 978-0-201-50480-4*.
4. *Silberschatz, Abraham (1994). Operating System Concepts, Fourth Edition. Addison-Wesley. p. 148.* *ISBN 978-0-201-50480-4*.
5. *Silberschatz, Abraham; Galvin, Peter Baer; Gagne, Greg (2008). Operating system concepts (8th ed.). Hoboken, N.J.: Wiley. p. 198.* *ISBN 978-0470128725*.