

Exploiting MLFQ:

The MLFQ algorithm could be exploited by a process so that the process can maintain a higher priority, making it execute faster. The process would need to voluntarily relinquish CPU control from time to time. This way, it will ensure it remains in a higher priority queue and be executed faster.

Comparison between different scheduling algorithms:

Approximate total times for benchmark code (used the command "time benchmark"):

- RR: 25000 ticks
- FCFS: 27000 ticks
- PBS: 20000 ticks
- MLFQ: 26000 ticks

Result:

FCFS is the least efficient, followed by MLFQ, followed by RR, followed by PBS. Therefore, PBS is the most optimized of the scheduling algorithms tested.

The below graph was generated by running the given benchmark code with 4 processes. In this graph, the increase in queue number is caused by demotion of processes due to them exceeding the specified time slice, and the decrease in queue number is caused by aging (which increases the priority of a process that has been waiting for more than a specified time).



