

Aaron Althoff

Lab 4

The multilevel feedback queue time sharing scheduling achieves fair scheduling by giving processes that are I/O bound higher priority and smaller CPU time slices, while decreasing the priority of CPU bound processes but giving them more CPU time. This implementation achieves fairness because it allows the I/O bound processes that need less time to execute first, increasing responsiveness of interactive applications. It then allows CPU bound processes more time to run in the background when all of the I/O bound processes are finished. The implementation in this lab achieves this by increasing the priority of a process if it doesn't use its entire time slice, or decreasing it if it does. In my implementation, to handle the null process the lowest I allowed the priority of any process besides the null process to be was 1. That ensured that no process besides the null process would have priority of 0 and the null process would only run if it was the last process.