Jacob Pinksa
Jogen Pathak
4348.503
3/27/2023
JRP180005

Possibility of Starvation
FCFS
In FCFS, the possibility of starvation is low. Processes will be run in the order they become available. No process is able to cut in line. The only issue would be when one process takes extremely long. Shorter processes would not be able to cut in and take longer than one might expect.

Round Robin
Round Robin is impossible to starve. It will progress through all ready processes in an equal amount of time, resulting in fast processes finishing faster on average while longer processes take a bit more time. All processes will progress so starvation is avoided.

SRT
In SRT a big process will end up waiting until the end to be able to make progress. If there are many processes, a big process could starve for a long time.

HRRN
HRRN is all around better at avoiding starvation while not being too inefficient. It prioritizes faster processes but also prioritizes processes that have waited a long time. It also will not interrupt processes for other faster process so that the long processes can finish.

Effect of Context Switching on Throughput
Looking at each algorithm's throughput, Context switching did not affect the throughput. Each process had the same throughput.

Fairness
FCFS
FCFS treats the processes fairly by not letting smaller processes cut ahead of larger processes. Per minute it will suffer in throughput but no processes are favored over others
Round Robin
Round Robin gives all ready processes the same amount of time regardless of size or how long they have been running. It will suffer throughput in the middle due to how many processes it could have sharing the disk at once.
SRT
SRT is unfair, it prioritizes smaller process over larger one. It will suffer in throughput at the end.
HRRN

HRRN is more fair than SRT, but still prioritizes smaller processes. It prevents bigger processes from being starved

Instructions on how to run
Read the ipynb python notebook files to see output or run the .py files. The python files can be run in the command prompt as long as you have python. They need no input.

Discussion
FCFS and HRRN all around had the most consistent performance. With relatively equal throughput throughout. Round Robin is slow initially and completes most of its processes in the later half. SRT in contrast completes most of its processes early on by prioritizing the short ones. HRRN has all the benefits of each algorithm and is the most consistent with throughput per minute. I would recommend HRRN as the best out of all of them.

Proposal
All the algorithms take the same amount of time total to complete the processes. Increasing the speed would be based on other factors. For instance, swapping from process to process takes time, so algorithms that finish the current process will run slightly faster on an actual computer. In terms of fairness that won't leave certain processes starving, HRRN and FCFS are the best. HRRN finishes the current process, so it does not swap more than it needs to. It also treats processes more fairly. If a long process is waiting a long time it will get to rather than wait until there are no more processes smaller than it. HRRN is better than the others.