Runs

After compiling and testing my program, I am disappointed with the results I got. Utilization of the various components are low, with the CPU having highest utilization at 60 percent and the lowest utilization being DISK2 with 0 percent. I think this is because I didn't implement my idea properly. The way the program is currently working, each job goes through the cycle until it exits from CPU, and then the next job can enter. When you look at the logs of the program, you can see that it is not in order, and the the first couple hundred events manage to cycle through and leave from the CPU, but the events generated after them don't even make it to the disks or network. Furthermore, the utilization of DISK2 is 0 because the queues for DISK1 and DISK2 are always 0. If both the queues are the same size, the the event gets sent to DISK1 by default. My program did not properly implement multiprogramming. Instead, only one event gets worked on at a time. More proof of this is that the events that exit the CPU do so in a chronological order. Theoretically, this should've not happened because each event spends a different amount of time in the various components of the simulation, which means that a job that arrives later could finish before a job that arrived earlier, but that never happened in my program. There is a fundamental design issue in my code.