Diana Kocsis

Professor Sherif Khattab

CS1550: Introduction to Operating Systems

12 June 2020

# FIFO Queue vs. Priority Queue

In Operating Systems, one of the most important concepts is CPU Scheduling. CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated by the CPU. There are several different scheduling methods including ones that use a first in, first out (FIFO) queue within a semaphore and ones that use a priority queue within a semaphore. Both of these have many pros and cons.

In the FIFO queue, the oldest process in the queue is allocated to the CPU first. One of the advantages of using a FIFO queue is that it is very simple to write and understand. It also has very fast runtime, where enqueuing and dequeuing is O(1) runtime. However, the average waiting time is often very long, and a convoy effect can occur as short processes wait for one big process to get out of the CPU. This can result in lower CPU and device utilization. In addition, using a FIFO queue within a semaphore is non-preemptive, meaning once the CPU is allocated to a process, the process holds the CPU until it gets terminated or reaches a waiting state. This can be very bad in situations where users need the CPU at regular time intervals.

On the other hand, a priority queue is a queue where a priority is associated with each process, and the CPU is allocated to the process with the highest priority. There are many advantages and disadvantages to using a priority queue within a semaphore. One of the advantages is that processes with a higher priority can be handled faster and do not have to wait very long compared to other processes that are not as important. One of the major cons is that

indefinite blocking, or starvation, can occur if a low priority process keeps on getting blocked by a stream of higher priority processes. This means that the process will never run, and it will be very detrimental to the program. In addition, the runtime of enqueueing and dequeuing into a priority queue is longer than that of a priority queue. If using a heap, the runtime will be O(log N) instead of O(1) as in FIFO queues.