1. Fair vs. Priority scheduling.

The most fundamental difference between fair and priority scheduling is actually quite accurately described by their corresponding names. A fair scheduling algorithm gives every participating member an equal chance to ‘win’ the ‘lottery’. Every participant has an equal chance to be the result of the algorithm. However, in a priority scheduling algorithm, certain members are prioritized over others. For example: there could be several participants in the algorithm with different assigned time values. In the algorithm it is possible that participants with higher time values could be prioritized, so they have a higher chance to be chosen by the algorithm than those with lower time values.

2. Scheduling policy descriptions

Fair Scheduling

Round-robin scheduling

Round-robin scheduling employs time-sharing. This means that each job is given a certain amount of CPU time. If the job isn’t completed when the CPU time runs out, the job is interrupted. The algorithm thus in fact iterates between all jobs, assigning them all a certain amount of CPU time. So when a job is interrupted, eventually more CPU time will be assigned to the same job, the job will then be resumed. When a process is terminated, the algorithm will start handling the next task in line and allocate a full portion of CPU time to the next job. This sequence repeats until the job is finished and no more CPU time is required.

Work-conserving scheduling

A work conserving scheduler works kind of similarly to the round-robin scheduler. The main difference in this scheduler is that this scheduler automatically tries to keep scheduled resources busy. If there is a submitted job which is ready to be scheduled, the work-conserving scheduler will try to schedule those jobs, even while the job isn’t set to be scheduled yet. This makes sure all the work is always done.

Priority scheduling

Earliest deadline first scheduling

This is a way of scheduling most students are probably very familiar with. In this kind of scheduling the queue of jobs is searched for the process closest to its deadline. This is the process that will be executed the first.

Shortest job first scheduling

In shortest job first scheduling, the queue of jobs is searched for the job that takes the least CPU time to schedule. The job which shows to need the least time will then be scheduled. When the scheduling is finished, the queue is searched again to find the next shortest job. This sequence is repeated until all of the jobs are finished.

3. Systematic comparison

All of the described scheduling methods obviously have the same end goal; all jobs should be scheduled and the queue should be empty when the scheduling is finished. The way in which this is approached however, is fairly different from one case to another. The way in which the jobs are scheduled might not differ all that much, the way in which is determined which job to schedule first however, is. In the fair scheduling methods, this is still fairly equal. All jobs have the same chance to be scheduled first.