Complexity

1. SJF:

The total complexity is O(r log r + t \* (c + i + r + e + r log r)), where r means ready queue size, c means number of CPU, i means io list, e means existing, t means time elapse.

2. SRT:

The total complexity is O(r log r + t \* (c + i + r + e log e + e \* c + e)), where r means ready queue size, c means number of CPU, i means io list, e means existing, t means time elapse.

3. Round-Robin:

The total complexity is O(c + t \* (c + i + r + e)), where r means ready queue size, c means number of CPU, i means io list, e means existing, t means time elapse.

4. Preemptive Priority:

The total complexity is O(r log r + t \* (c + i + r + e log e + e \* c + e)), where r means ready queue size, c means number of CPU, i means io list, e means existing, t means time elapse.

SJF prefers to short time processes, in this case, IO bound process, and don’t prefer the CPU-bound process, so it spends more time to deal with interactive, and the real-time is short. And this algorithm may cause starvation in CPU-bound processes. This algorithm is especially appropriate for batch jobs.

SRT prefers to short time processes, in this case, IO bound process, and don’t prefer the CPU-bound process, so it will deal with the IO first, most of the time when an IO process arrive, the SRT will preemptive to deal with IO first and the real time will be a little higher because CPU-bound always been preemptive. This algorithms deal with batch processes well and the real- time is not high. And this algorithm may cause starvation in CPU-bound processes. SRT is a perfect algorithm to deal with interactive process.

Round Robin didn’t prefer to CPU-bound or IO-bound processes and treat different processes fairly. The interactive process is not working well in this algorithm because this interactive should require more frequencies to be processed. The real-time process is not been processed well because this algorithm treat different process fairly and real-time should requirement more attention. Batch process will be deal well in this algorithm.

Preemptive Priority is very good algorithm to deal with interactive and real-time processes, because these two types of processes will require more attentions and should be handled in a higher priority. This algorithm deal with processes by priority, so either IO-bound or CPU-bound with the same priority will be treat equally. This algorithm may cause the starving for the low priority process.