CS2106 Tutorial 4

AY 25/26 Sem 1 — github/omgeta

Q1. (a.) FIFO minimises average response times if jobs arrive in ready queue in order of increasing job lengths. This avoids short jobs later from awaiting longer for an earlier longer job.

(b.) RR behaves identically to FIFO when job lengths are shorter than the time quantum, since jobs will never be pre-empted.

(c.) When there are many jobs exceeding the time quantum, RR results in reduced throughput due to greater overhead from context-switches.

(d.) Yes: when time quantum is too small, there is high overhead from context switching reducing throughput and increasing average turnaround time.
No: reducing time quantum improves initial response time.

(e.) Yes: CPU processes can be given higher priority for I/O so they can return to waiting for the CPU, decreasing overall turnaround time at the expense of responsiveness of IO-bound processes.
No: maximise responsiveness of IO-bound processes.

Q2. (a.) $(nChild + 1) \times n$

(b.) For large values of n or $nChild$, we do not get the expected result. This is because of race conditions where processes are reading the same old value and writing back the same incremented value, losing increments. As more processes make more additions, there are more interleavings, giving higher error.

(c.) No, the issue persists with single core. Likely because of preempting between the load/add/write steps causing races.

Q3. (a.) shm is faster as processes do not have to wait their turn, since processes are not guaranteed to be scheduled in order, we will not a lot of waiting time.