

* Shortest Job First / Priority Scheduling / Round Robin

① Shortest Job First (SJF) [Non-preemptive]

- ① Process with least BT: Will be dispatched to CPU first.
- ② Must do estimation for BT for each process in ready queue beforehand, correct estimation of BT is merely an impossible task.
- ③ Criteria for SJF $\rightarrow (AT + BT)$.
- ④ This will suffer from Convoy effect as if the very first ~~process~~ process which came in ready state is having a larger BT. Starvation might happen.

② SJF [Preemptive]

- ① Less starvation & No Convoy effect.
- ② Gives average WT less for a given set of processes as scheduling short job before a long one decreases the WT of short job more than it increases the WT of the long process.

③ Priority Scheduling [Non-preemptive]

- ① Priority is assigned to a process when it is created.
- ② SJF is a special case of general priority scheduling with priority inversely proportional to BT.

④ Priority Scheduling [Preemptive]

- ① Current RUN State job will be preemptive

- if next job has higher priority comes.
- (b) May cause indefinite waiting (starvation) for lower priority jobs. Ageing is the solution, gradually increase priority of process that wait so long.

⑤ Round Robin Scheduling (RR)

- (a) Most popular, like FCFS but preemtive.
- (b) Designed for time sharing systems.
- (c) Criteria = $AT + \text{time quantum}(TQ)$, Doesn't depend on BT.
- (d) No process is going to wait forever, hence very low starvation [No Convoy effect].
- (e) Easy to implement, If TQ is small, more will be context switch (more overhead).

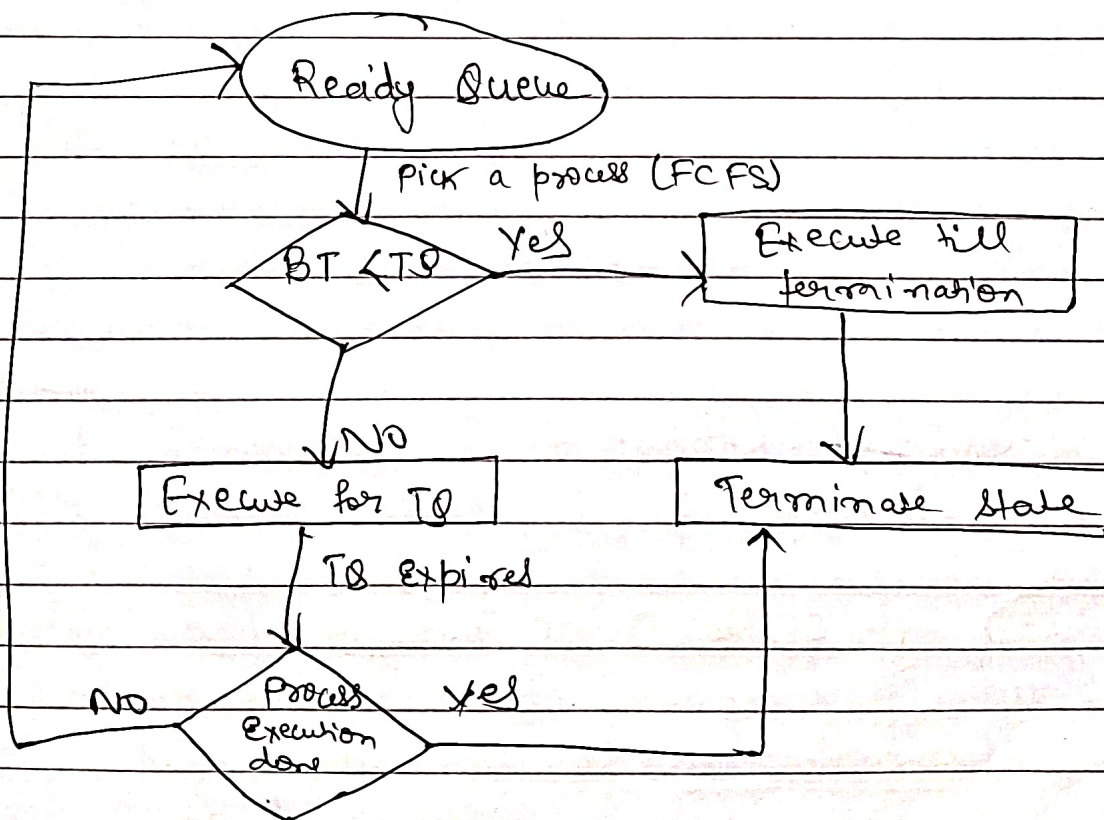


Diagram of RR.