Scheduling Policies

CPU allocation algorithms and policies used by the Processor Manager (i.e., job and process schedulers)

Algorithm	Policy Type	Disadvantages	Advantages
First-Come, First-Served	Non-preemptive	Unpredictable turnaround times; element of chance	Easy to implement
Shortest Job Next	Non-preemptive	Indefinite postponement of some jobs; requires execution times in advance	Minimises average waiting time
Priority Scheduling	Non-preemptive	Indefinite postponement of some jobs	Ensures fast completion of priority jobs
Shortest Remaining Time	Preemptive	Overhead incurred from context switching	Ensures fast completion of short jobs
Round Robin	Preemptive	Requires optimal time quantum selection	Reasonable response times for interactive users; fair CPU allocation
Multiple-Level Queues	Preemptive / Non-preemptive	Overhead incurred from monitoring queues	Flexible scheme; ageing or similar prevents indefinite postponement; fair for CPU-bound jobs
Earliest Deadline First	Preemptive	High overhead from monitoring dynamic deadlines	Facilitates timely job completion

---

Job and Process + Thread States

Job & Process States

1. **HOLD**: job scheduler moves from hold to ready
2. **READY**: process scheduler moves from ready to running
3. **RUNNING**: process scheduler moves from running to: waiting; ready; or finished
4. **WAITING**: process scheduler moves from waiting to ready
5. **FINISHED**: job or process scheduler moves from running to finished

Thread States

1. **READY**: process scheduler moves thread to running when assigned to a processor
2. **RUNNING**: thread moves from running to either finished or one of the following
3. **WAITING**: thread moved to waiting to wait for external event (e.g., awaiting user input)
4. **DELAYED**: thread moved to delayed by programmatic design (e.g., sleep between email retrieval)
5. **BLOCKED**: thread moved to blocked by IO request or page fault