Process Scheduler Program Guide

Brief instructions for using the Round-Robin scheduling simulator (Dynamic vs Static).

1. Entry Point

Run only `Main.java`.
You MUST set (or confirm) the input and output file paths inside the `main` method (constants in `Main`): - `INPUT_PATH` → your events file (e.g. `inputDRR2.txt`) - `OUTPUT_PATH` → results file (e.g. `output.txt`)

No command-line arguments are required unless you modify the code.

2. Scheduler Selection

The configuration line includes `SCHED` (added extension): - `SCHED=1` → Dynamic Round Robin (quantum computed in `PrManager`) - `SCHED=2` → Static Round Robin (fixed quantum = `10 + teamNumber` → with team 7: 17)

Switch snippet:

```
switch (sched) {
    case 1 -> scheduler = new DRoundRobinScheduler();         //
        Dynamic RR (quantum computed in PrManager)
    case 2 -> scheduler = new SRoundRobinScheduler(10 + 7);  //
        Static RR, team 7 → Q = 17
    default -> scheduler = new SRoundRobinScheduler(10 + 7);
}
```

3. Input Specification

External events are defined one per line:

Type	Format	Notes
Configuration	`C <time> M=<totalMemory> S=<totalDevices> SCHED=<1|2>`	Initializes system resources and scheduler
Arrival		Priority 1 may use Hold Queue

Type	Format	Notes
	`A <time> J=<id> M=<memReq> S=<devReq> R=<burst> P=<priority>`	1; others Hold Queue 2
Display	`D <time>`	A large time (e.g. 999999) used for final snapshot

Fields: - `M` = memory units required / total - `S` = devices required / total - `R` = CPU burst (execution cycles) - `P` = priority - `J` = job ID

4. Example Input File (`inputDRR1.txt`)

```
C 2 M=100 S=1 SCHED=1

A 0 J=1 M=6 S=0 R=5 P=1
A 0 J=2 M=6 S=0 R=5 P=1
A 0 J=3 M=6 S=0 R=5 P=1

A 5 J=4 M=6 S=0 R=5 P=1
A 5 J=5 M=241 S=0 R=5 P=1
A 5 J=6 M=24 S=0 R=5 P=1

D 999999
```

5. Output

The program writes a snapshot (at `D` times) to the output file: - System resources (total/available) - Ready / Hold queues - Finished jobs table (ArrivalTime, CompleteTime, TurnaroundTime, WaitingTime)

Turnaround = `completionTime - arrivalTime`
Waiting = `turnaround - originalBurst`

6. Notes

- Jobs exceeding total memory/devices are rejected.
- Dynamic RR quantum = rounded average of remaining bursts (running + Ready) or full remaining if alone.
- Static RR quantum fixed (17 with team 7).

That’s all—set paths in `Main.java`, prepare the input file, run `Main.java`.