CSCI 3753: Operating Systems Fall 2024

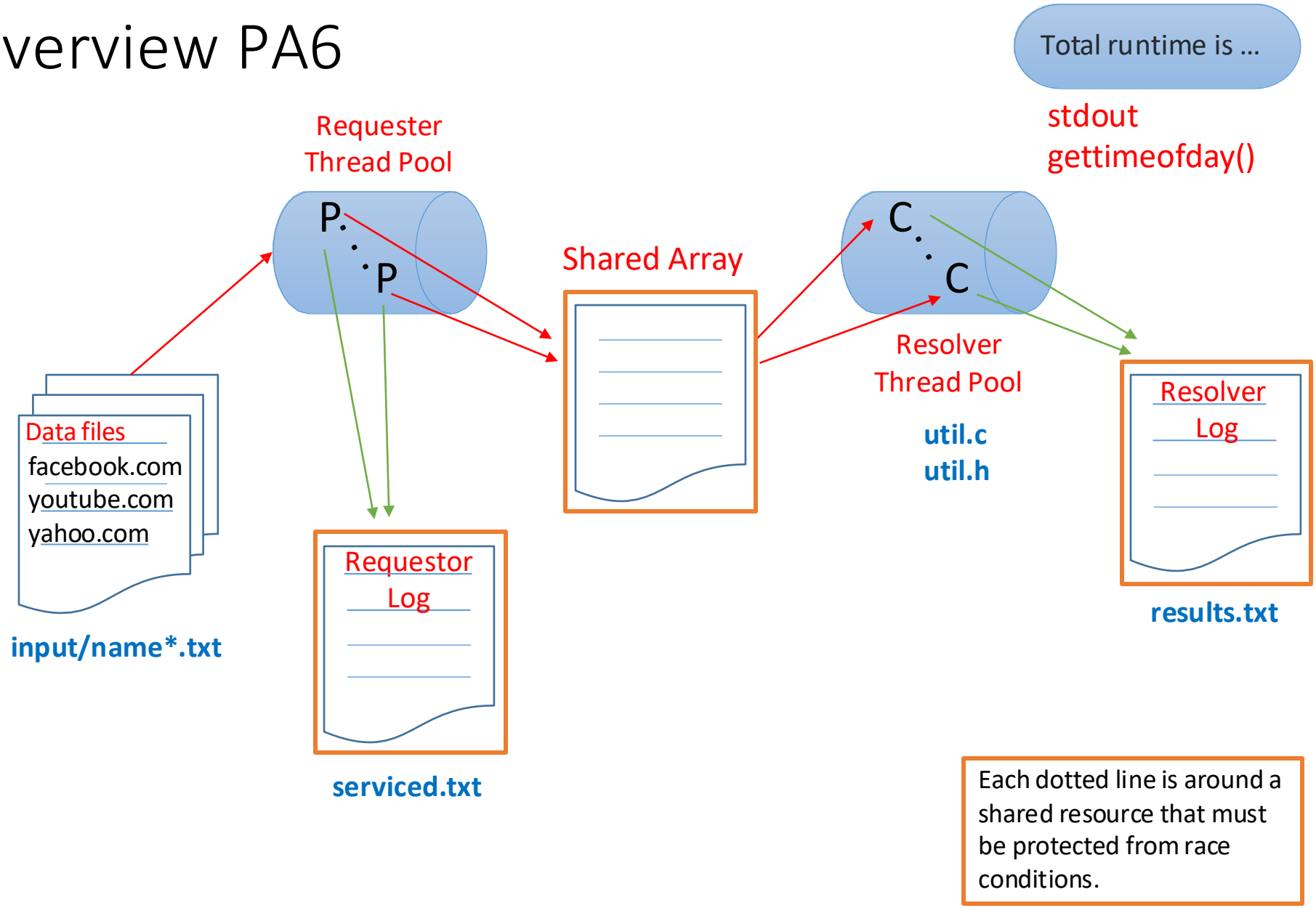
Dylan Sain

Department of Computer Science

University of Colorado Boulder

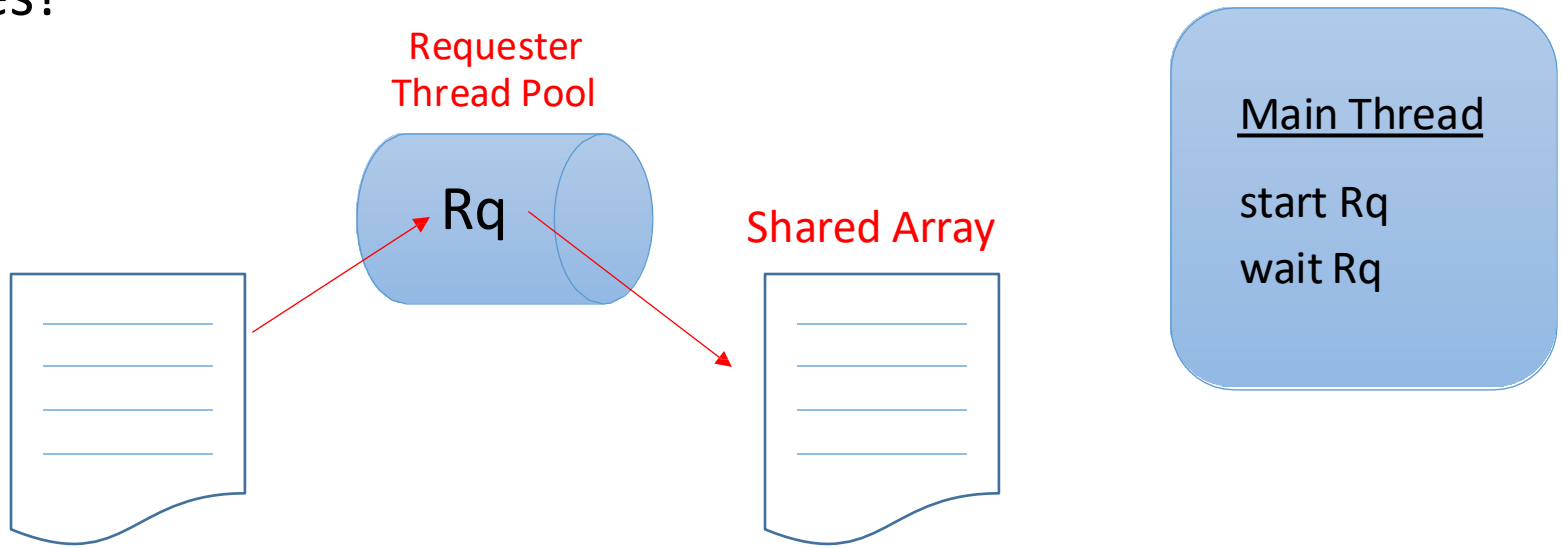
Week 10: PA6 and Process Scheduling

Overview PA6



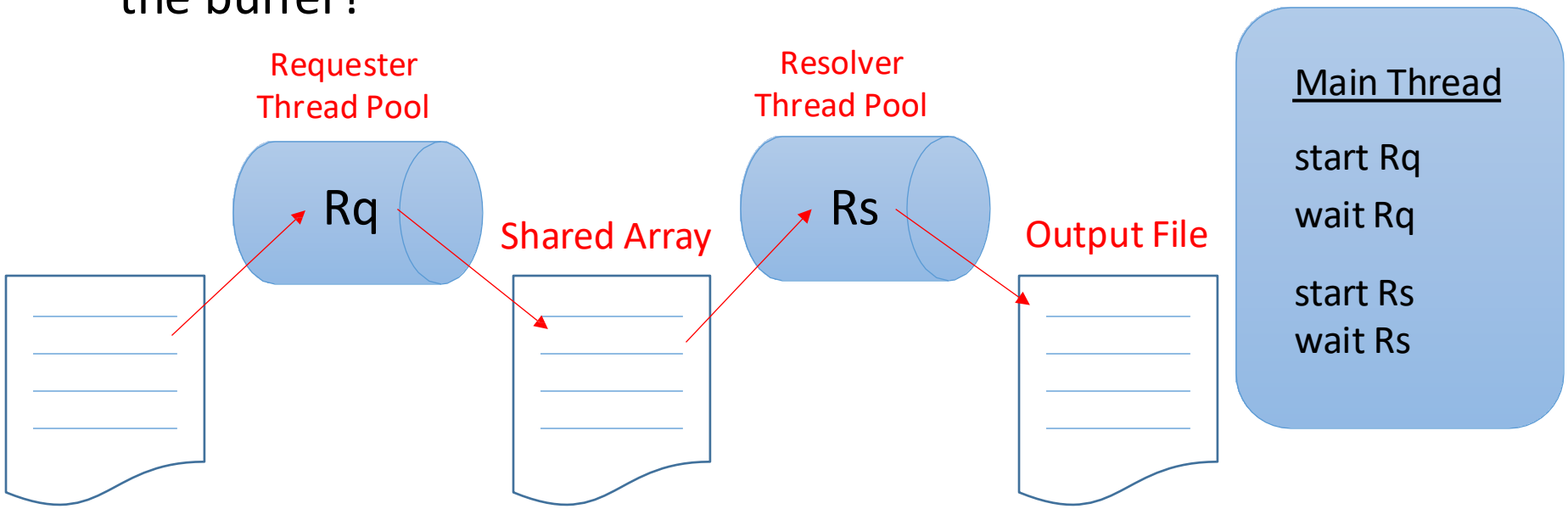
Implementation – Step 1

- **Action:** Create a simple program to create a requester thread that will repeatedly read a line from a given file and add an entry into the shared buffer
- **Validation:** Does the buffer have the correct number of entries?



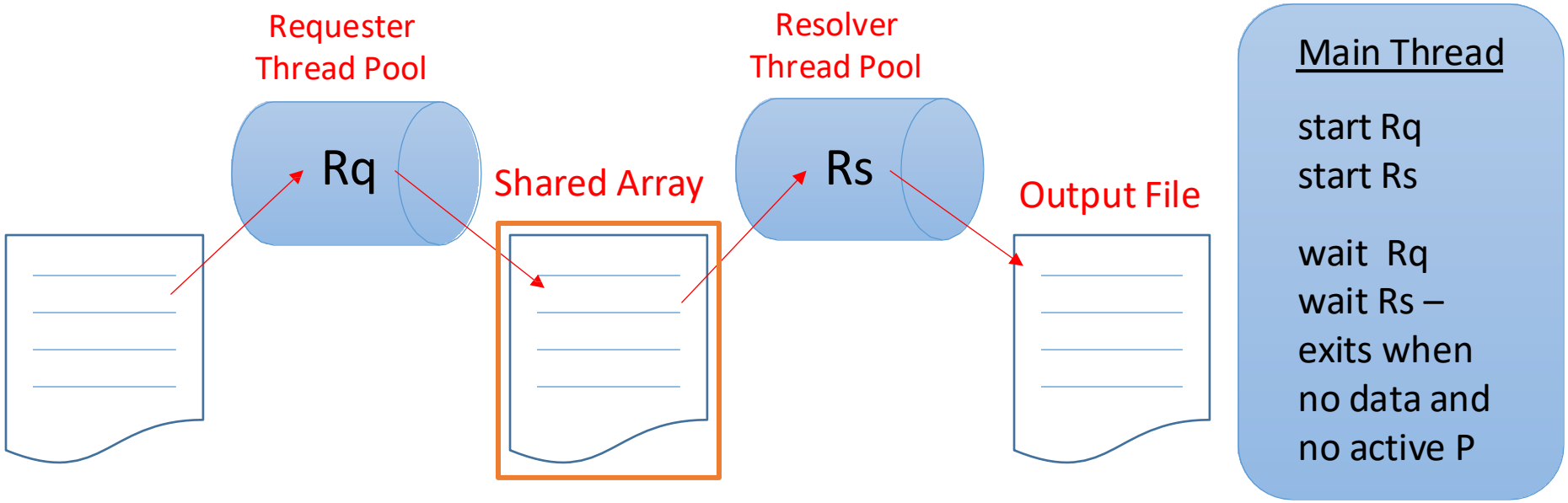
Implementation – Step 2

- **Action:** Use the result in step 1, once that process is complete, start a resolver thread to take items out of the buffer. Then, write the results to an output file.
- **Validation:** Does the output file contain the entries stored in the buffer?



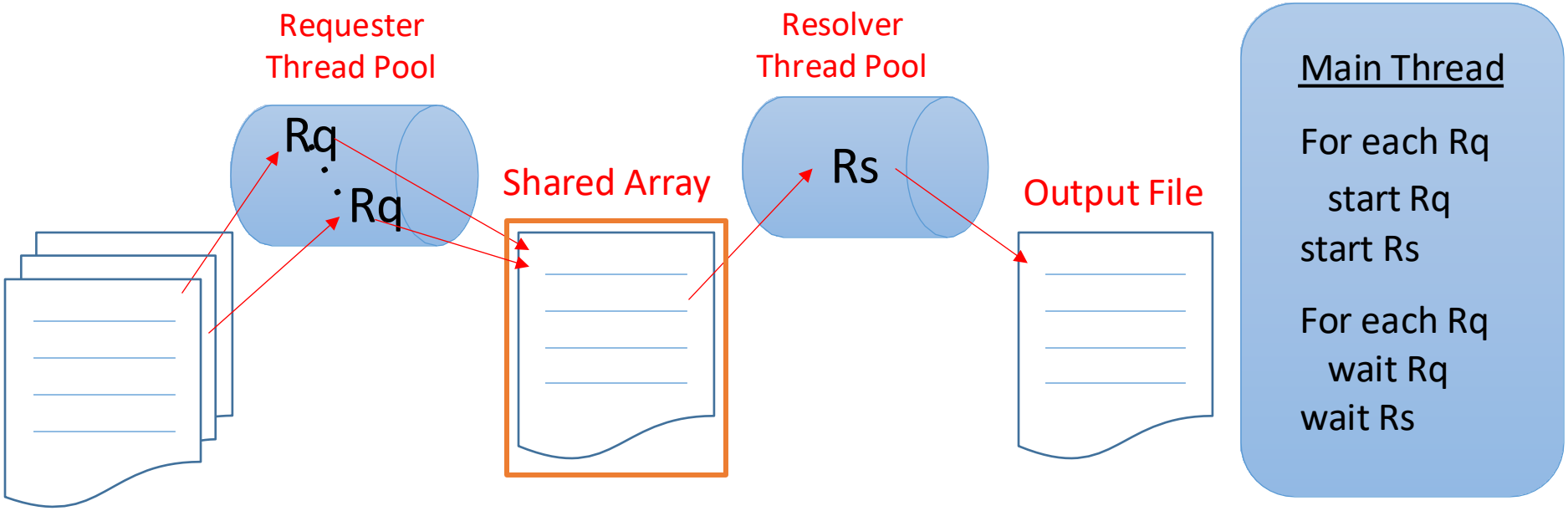
Implementation – Step 3

- **Action:** Once you are assured that your application can write and read to the buffer correctly (although serially), then try to make them run concurrently. Multiple processes accessing and modifying the same data can cause race conditions. You must protect the critical sections of each thread with a mutex.



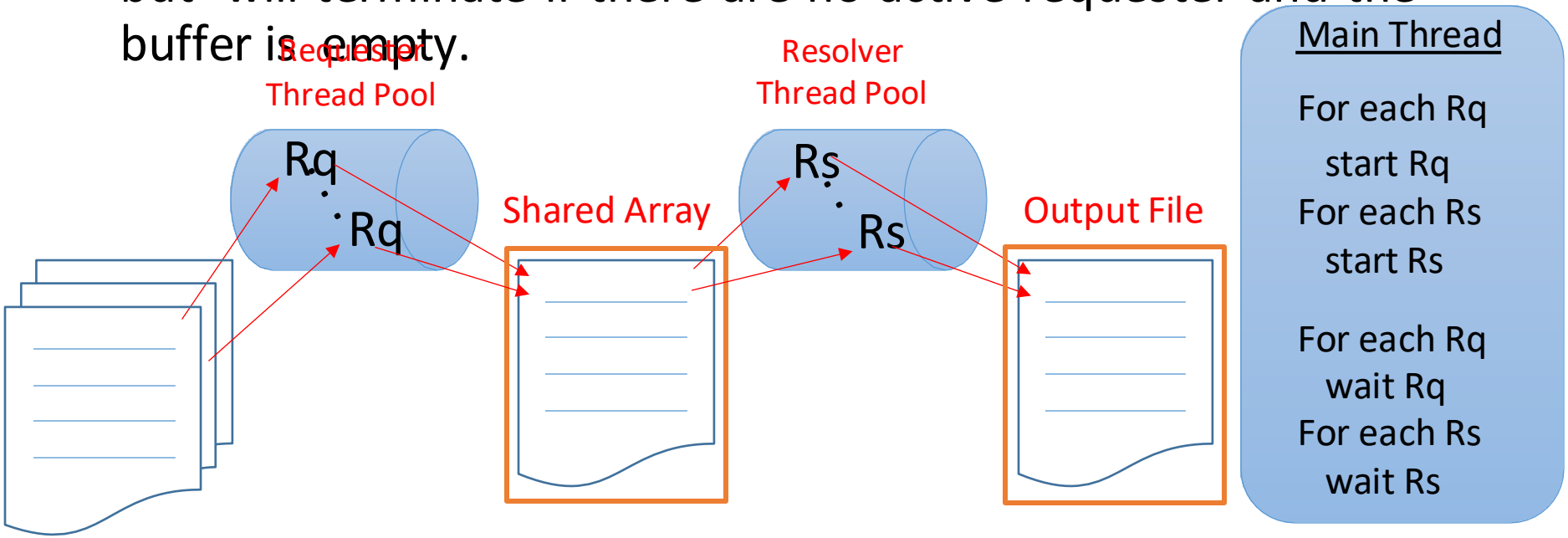
Implementation – Step 4

- **Action:** create multiple requester threads to read from multiple different files. Each requester can read single lines from a different file. The requester will terminate when all lines from the file have been processed.



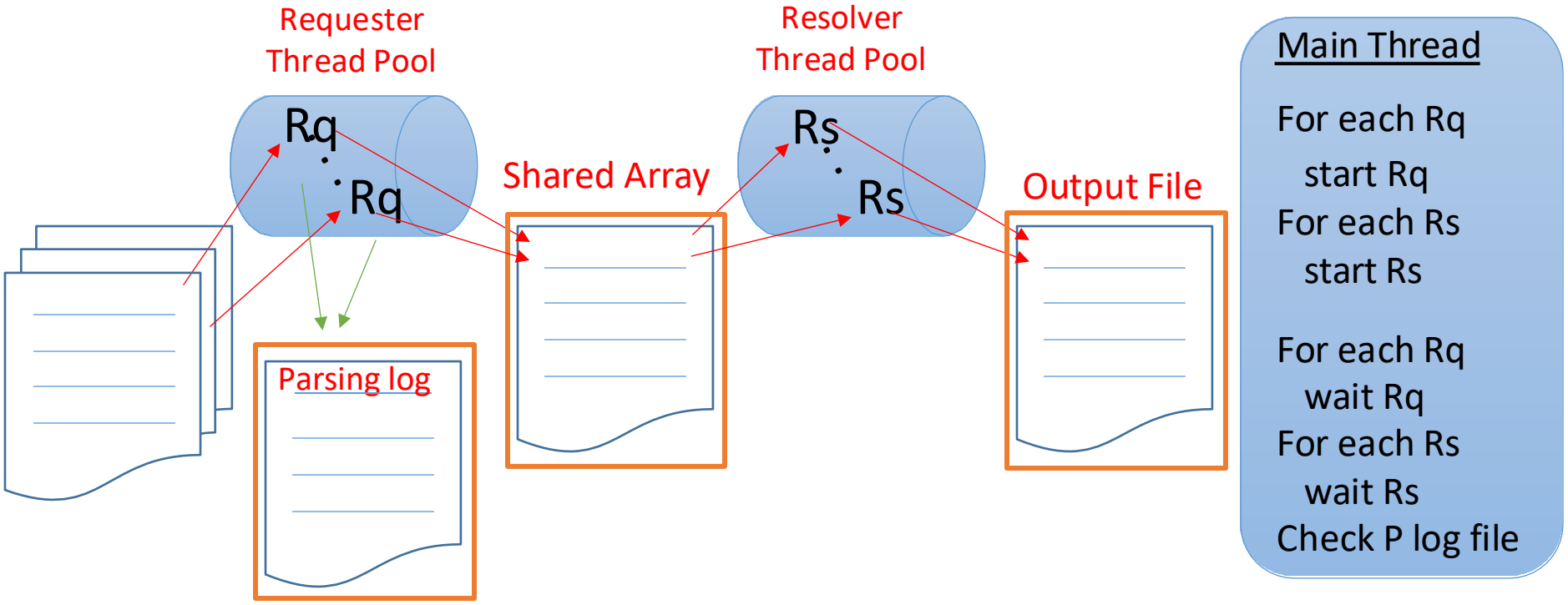
Implementation – Step 5

- Action: create multiple resolver threads to read from multiple requester threads via a single shared buffer.
  The resolver will wait for data (spin wait is acceptable) but will terminate if there are no active requester and the buffer is empty.



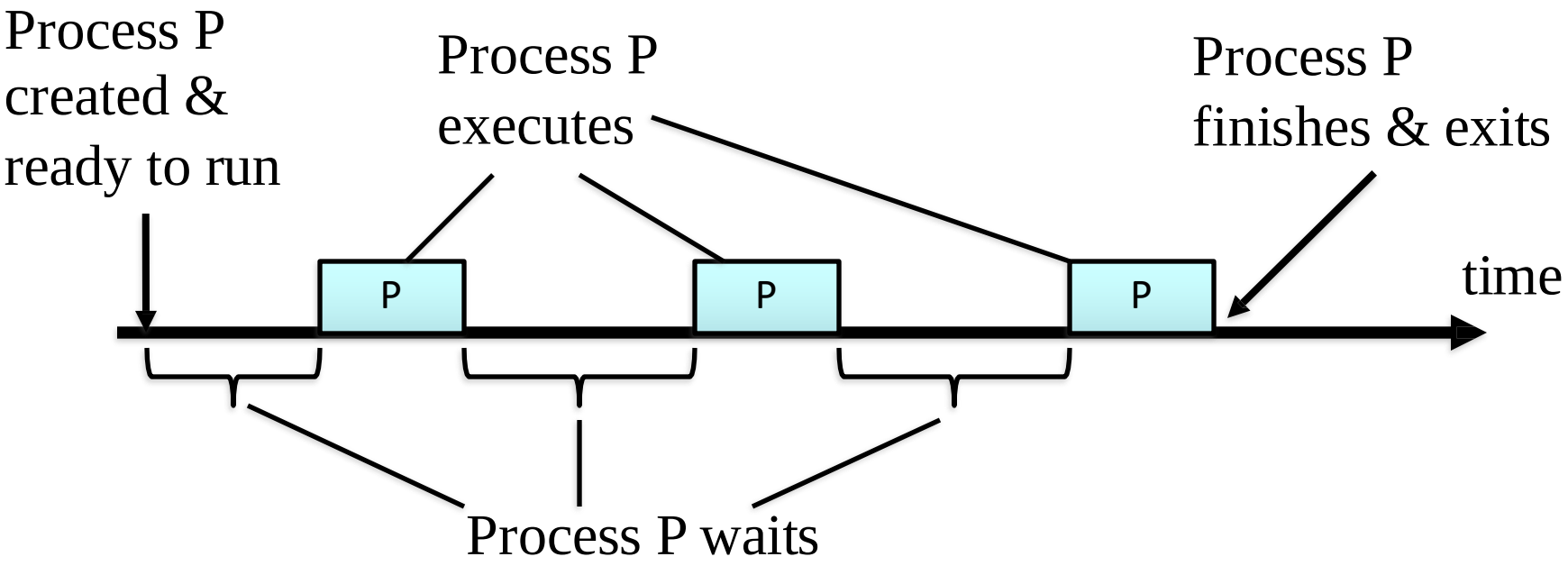
Implementation – Step 6

- **Action:** move back to the requester threads, each thread must record the data it has processed. Files are a shared resource and therefore must be protected from multiple processes accessing it.



Scheduling

Per Job or Task Metrics



- Execution time
- Wait time
- Turnaround time
- Response Time

Exercise: Test your skills!

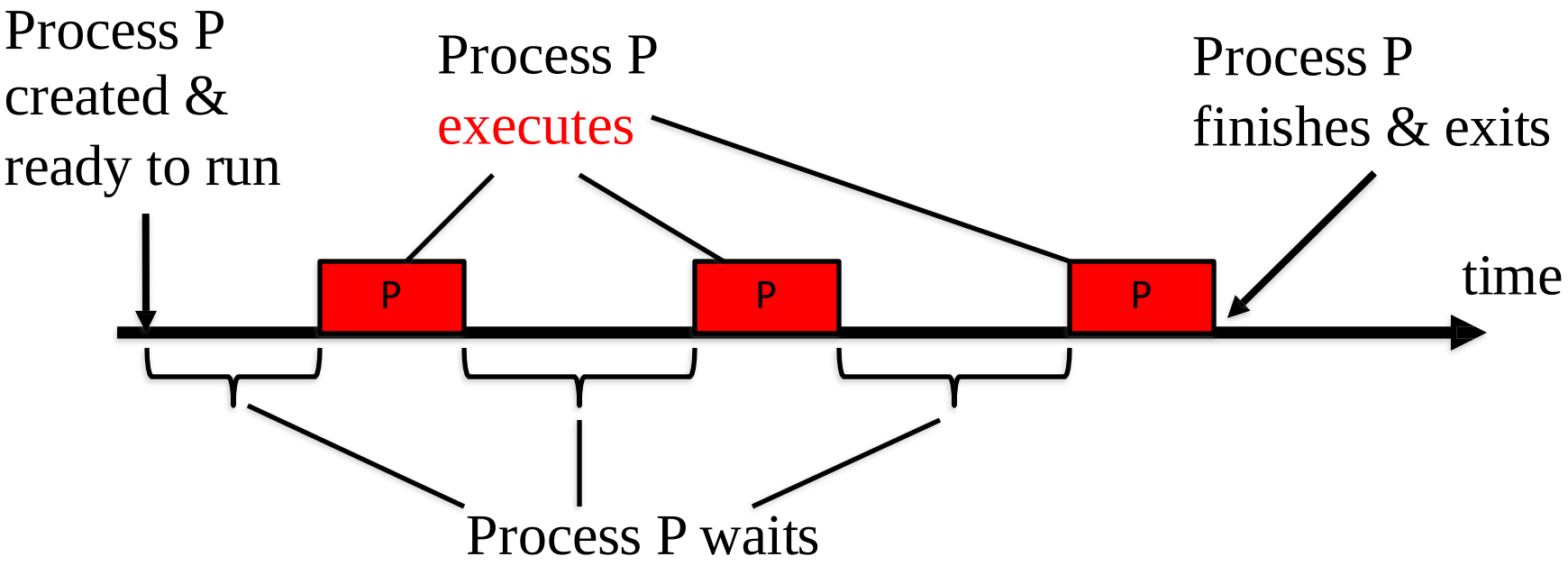
Go to Kahoot.it

Use the code

Add your name (please use your real name)

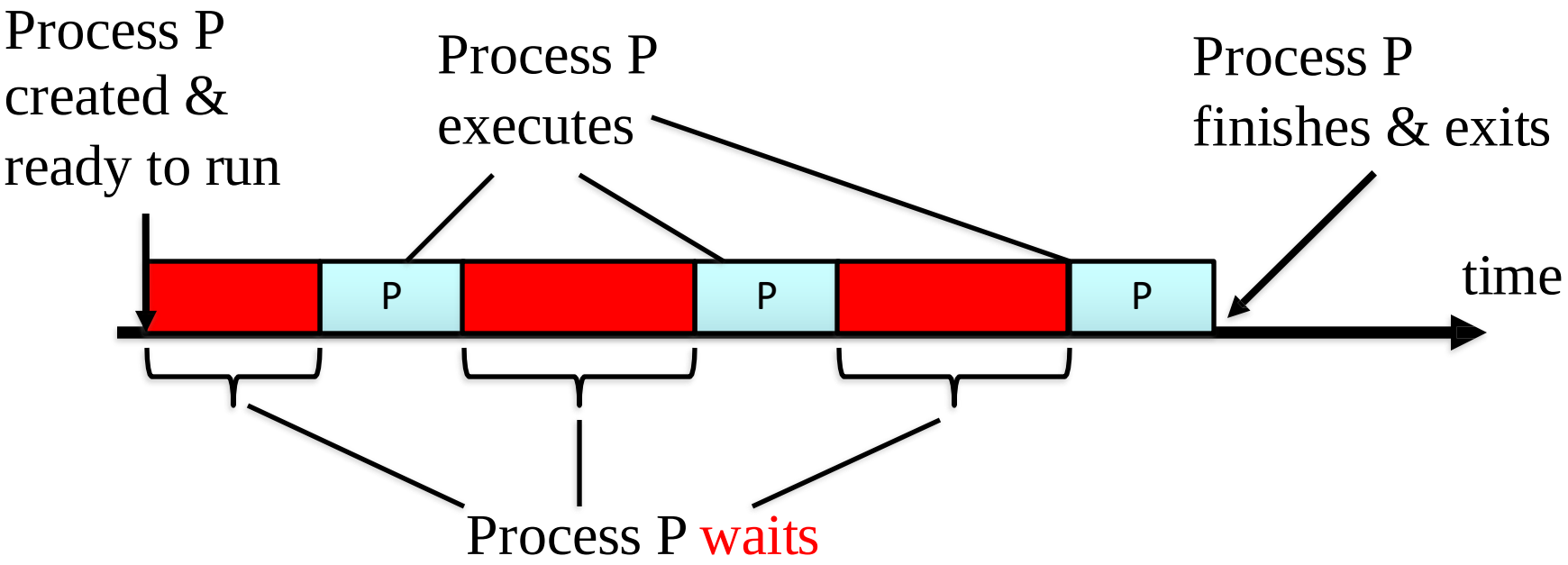
Choose the correct answer!

Per Job or Task Metrics



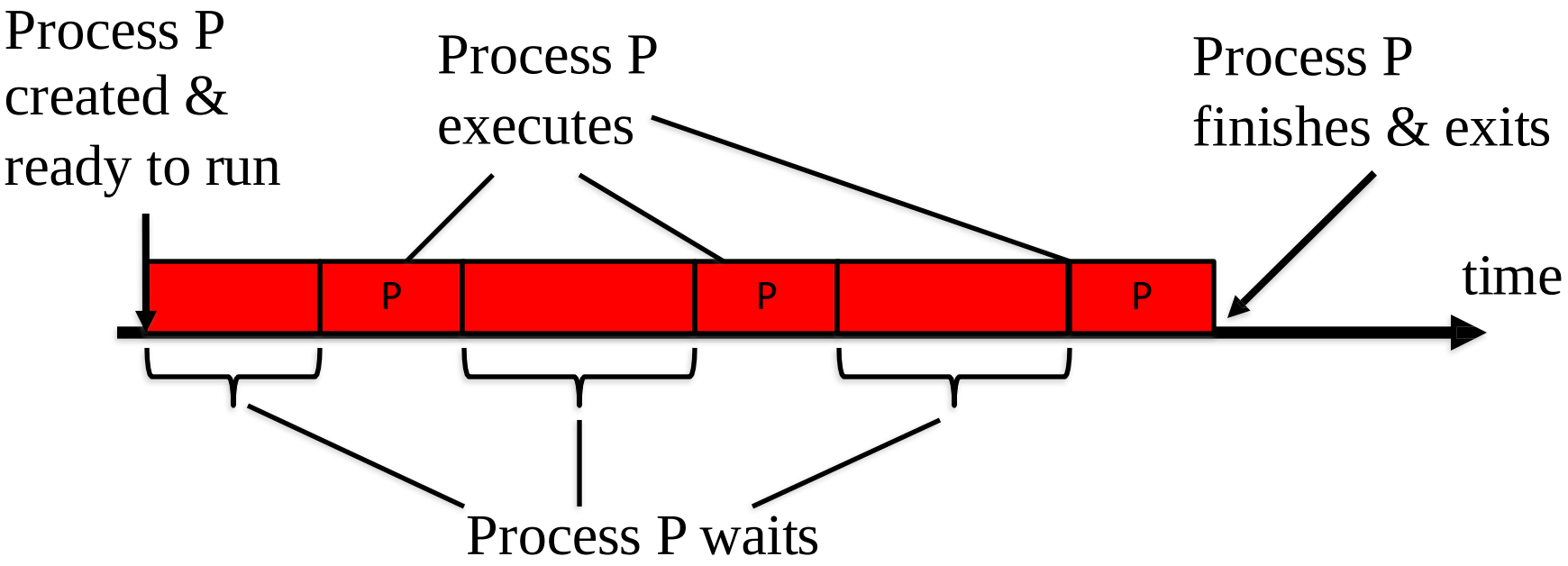
- ***Execution time*** E(P) = the time on the CPU required to fully execute process P
  - Sum up the time slices given to process P
  - Also called the "burst time" by textbook

Per Job or Task Metrics



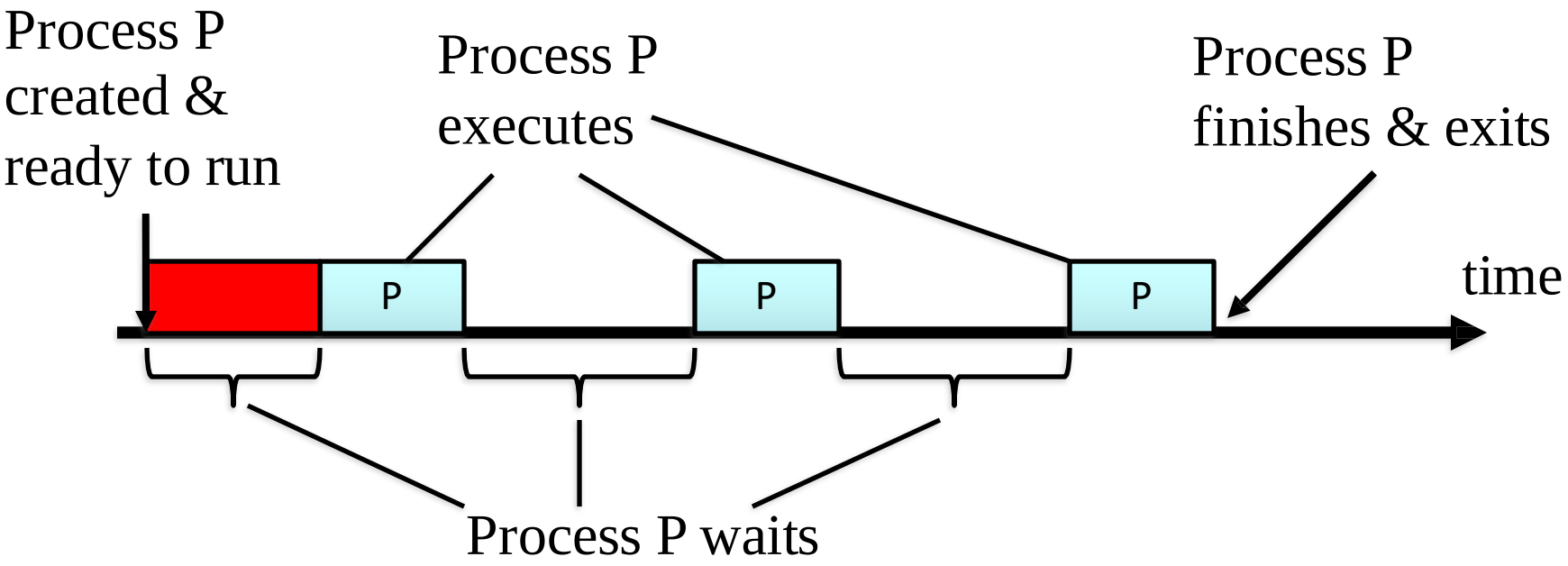
- ***Wait time*** W(P) = the time process P is in the ready state/queue waiting but not running
  - Sum up the gaps between time slices given to process *P*
  - But, does NOT include I/O waiting time

Per Job or Task Metrics



- ***Turnaround time*** T(P) the time from 1st entry of process P into the ready queue to its final exit from the system (exits last run state)
  - Does include time waiting and time for IO to complete

Per Job or Task Metrics



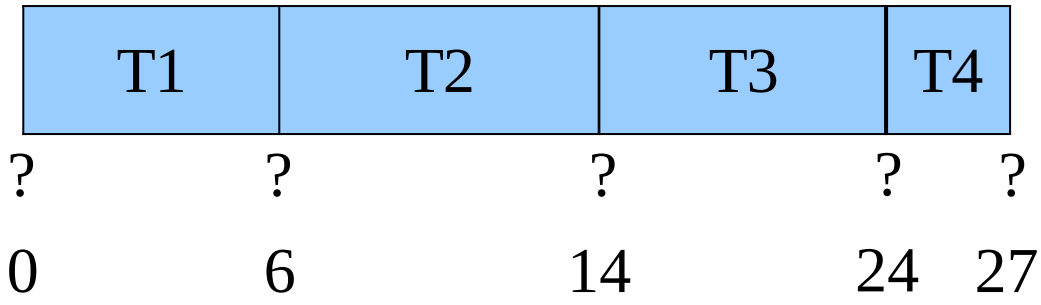
- ***Response time*** R(P) = the time from 1st entry of process P into the ready queue to its 1st scheduling on the CPU (1st occurrence in running state)
  - Useful for interactive tasks

Scheduling Policies

- First Come First Serve (FCFS) Scheduling
- Shortest Job First (SJF) Scheduling
- Round Robin Scheduling
  - Weighted RR
- Earliest Deadline First Scheduling
  - W/ or w/o preemption
  - Least slack (slack time = time until deadline – remaining execution time)
- Priority-based or Multilevel Queue Scheduling
- Multilevel Feedback Queue Scheduling
- Completely Fair Scheduler in Linux

Scheduling Policies

- First Come First Serve (FCFS) Scheduling
  - Average wait time?
    → (6+14+24)/4 = 11 ms
  - Average turnaround time?
    → (6+14+24+27)/4 = 17.75 ms

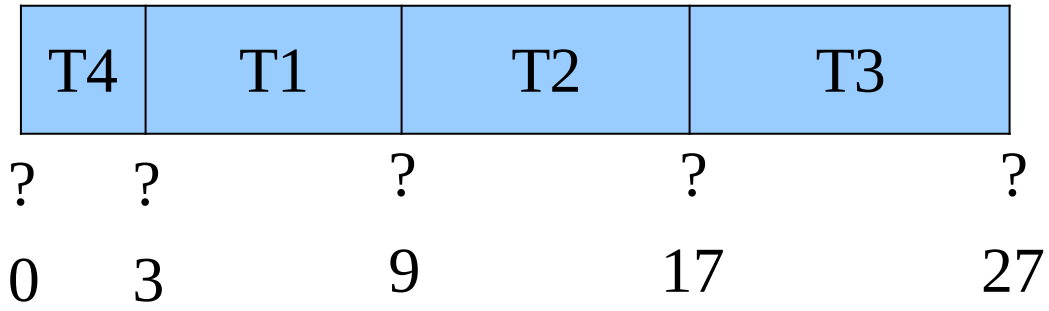


Task	CPU Execution Time (ms)
T1	6
T2	8
T3	10
T4	3

- All processes arrived just before time 0 in the order.

Scheduling Policies

- Shortest Job First (SJF) Scheduling
  - Average wait time?
    → (3+9+17)/4 = 7.25 ms
  - Average turnaround time?
    → (3+9+17+27)/4 = 14 ms

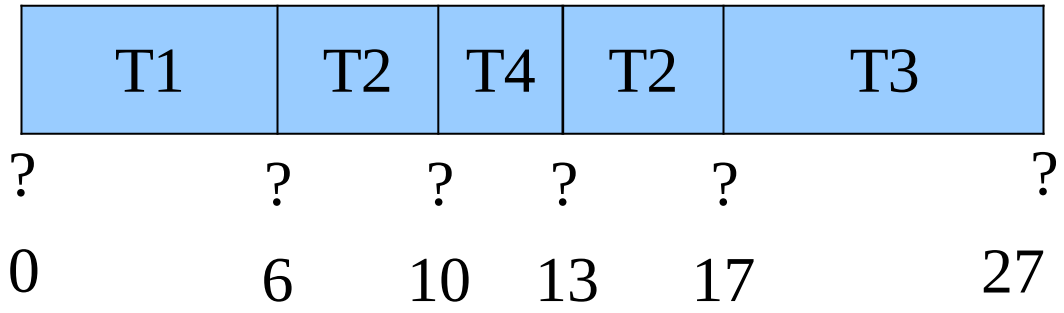


Task	CPU Execution Time (ms)
T1	6
T2	8
T3	10
T4	3

- All processes arrived just before time 0.

Scheduling Policies

- Shortest Job First (SJF) Scheduling w/ Preemption
  - Average wait time?
    → (6+3+17)/4 = 6.5 ms
  - Average turnaround time?
    → (6+6+3+8+3+17+10)/4 = 13.25 ms



Task	CPU Execution Time (ms)
T1	6
T2	8
T3	10
T4	3

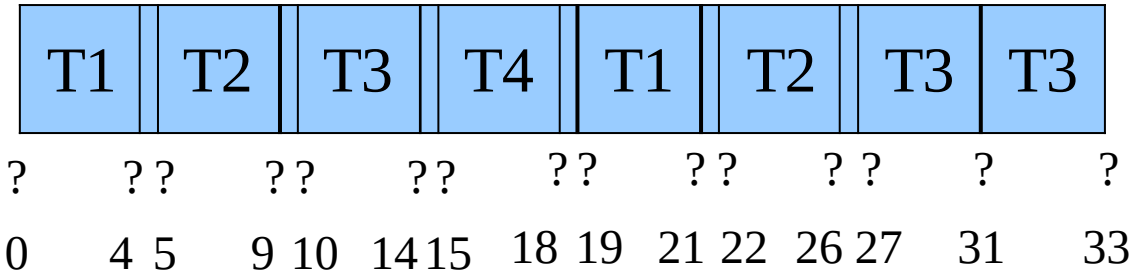
- T1, T2, and T3 arrived just before time 0. T4 arrived at 10ms.

Scheduling Policies

- Round Robin Scheduling
  - Let time slice = 4 ms
  - Assuming a 1ms switching time
  - Average wait time?
    → (15+5+13+10+13+15)/4 = 17.75 ms
  - Average response time?
    → (5+10+15)/4 = 7.5 ms

All processes arrived just before time 0 in the order.

Task	CPU Execution Time (ms)
T1	6
T2	8
T3	10
T4	3



Scheduling Policies

- Earliest Deadline First (EDF) w/ Preemption
  - Let time slice = 20 ticks
  - Assuming a 5 ticks of switching time

Task	CPU Execution Time	Arrival Time	Deadline
T1	30	0	100
T2	90	20	230
T3	40	55	145
T4	20	85	145