REAL TIME OPERATING SYSTEMS

Lesson-16:

Task Scheduling Cooperative Models

1. Common scheduling models

Common scheduling models

- Cooperative Scheduling of ready tasks in a circular queue. It closely relates to function queue scheduling.
- Cooperative Scheduling with Precedence Constraints
- Cyclic Scheduling of periodic tasks and Round Robin Time Slicing Scheduling of equal priority tasks
- Preemptive Scheduling
- Scheduling using 'Earliest Deadline First' (EDF) precedence.

Common scheduling models

- Rate Monotonic Scheduling using ‘higher rate of events occurrence First’ precedence
- Fixed Times Scheduling
- Scheduling of Periodic, sporadic and aperiodic Tasks
- Advanced scheduling algorithms using the probabilistic Timed Petri nets (Stochastic) or Multi Thread Graph for the multiprocessors and complex distributed systems.

2. Cooperative Scheduling in the cyclic order

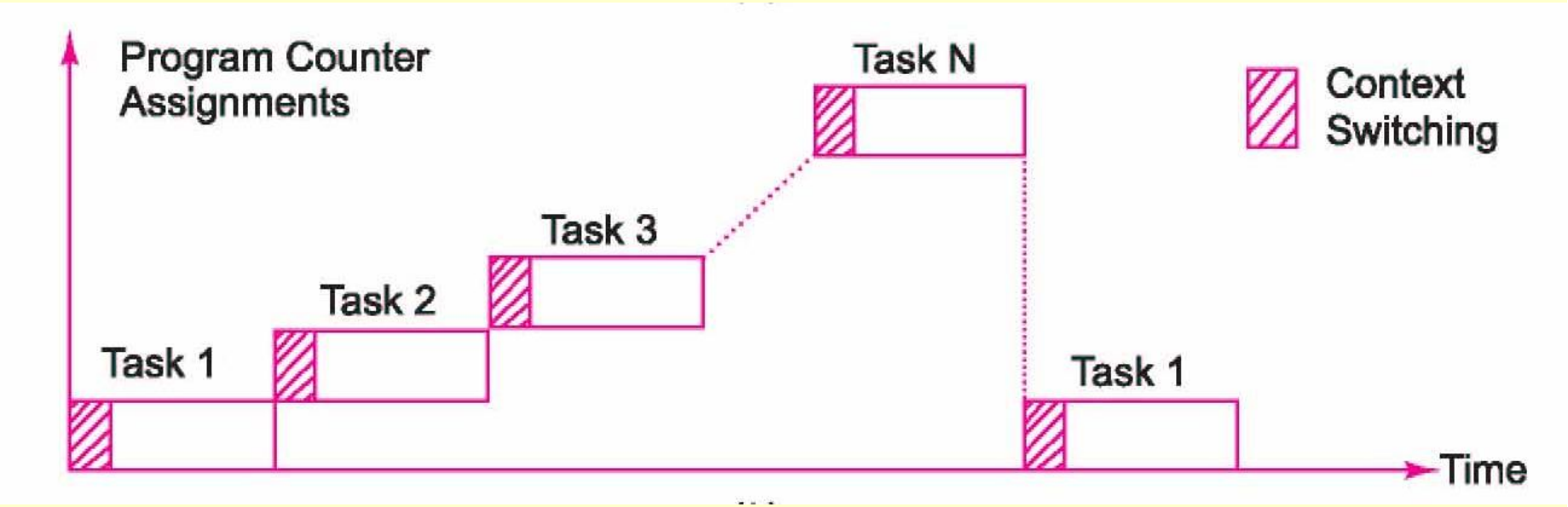
Cooperative Scheduling in the cyclic order

- Each task cooperate to let the running task finish
- Cooperative means that each task cooperates to let the a running one finish.
- None of the tasks does block in-between anywhere during the ready to finish states.
- The service is in the cyclic order

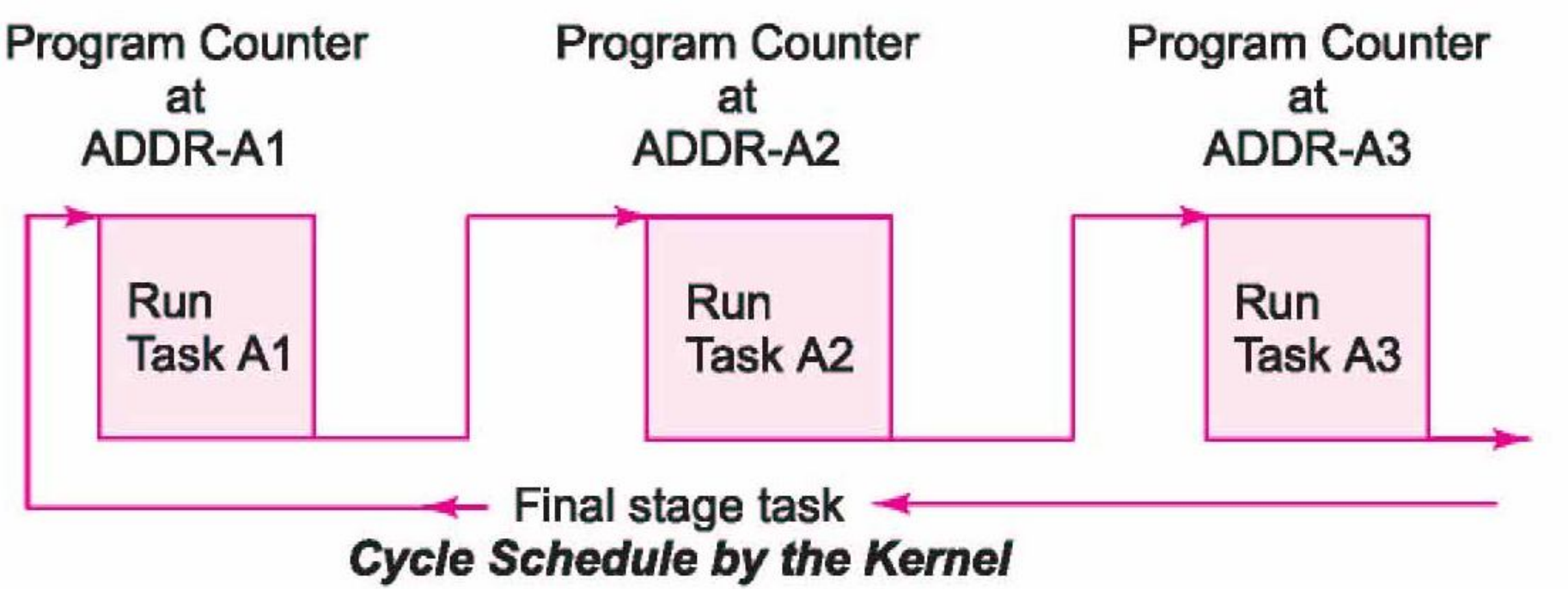
Worst-case latency

- Same for every task
- Tworst = $\{(st_i + et_i)_1 + (st_i + et_i)_2 + ... + (st_i + et_i)_{N-1} + (st_i + et_i)_N\} + t_{ISR}$.
- t_{ISR} is the sum of all execution times for the ISRs
- For an i-th task, switching time from one task to another be is st_i and task execution time be is et_i
- i = 1, 2, …, N –1 , N, when number of tasks = N

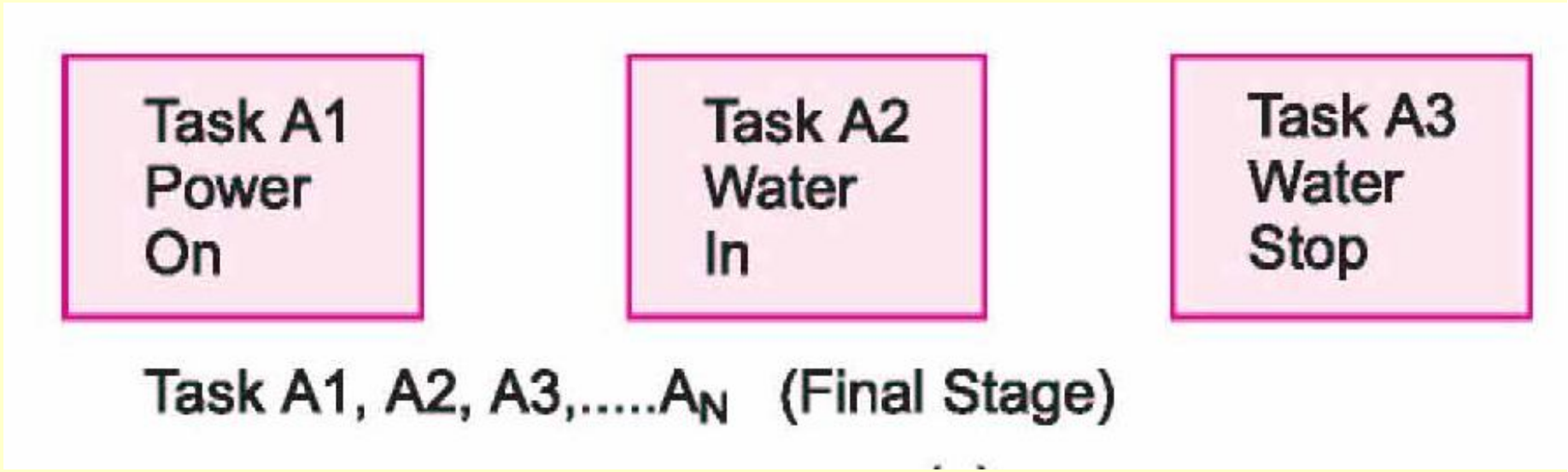
Program counter assignments (switch) at different times, when the on the scheduler calls the o tasks from the list one by one in the circular queue from the list.



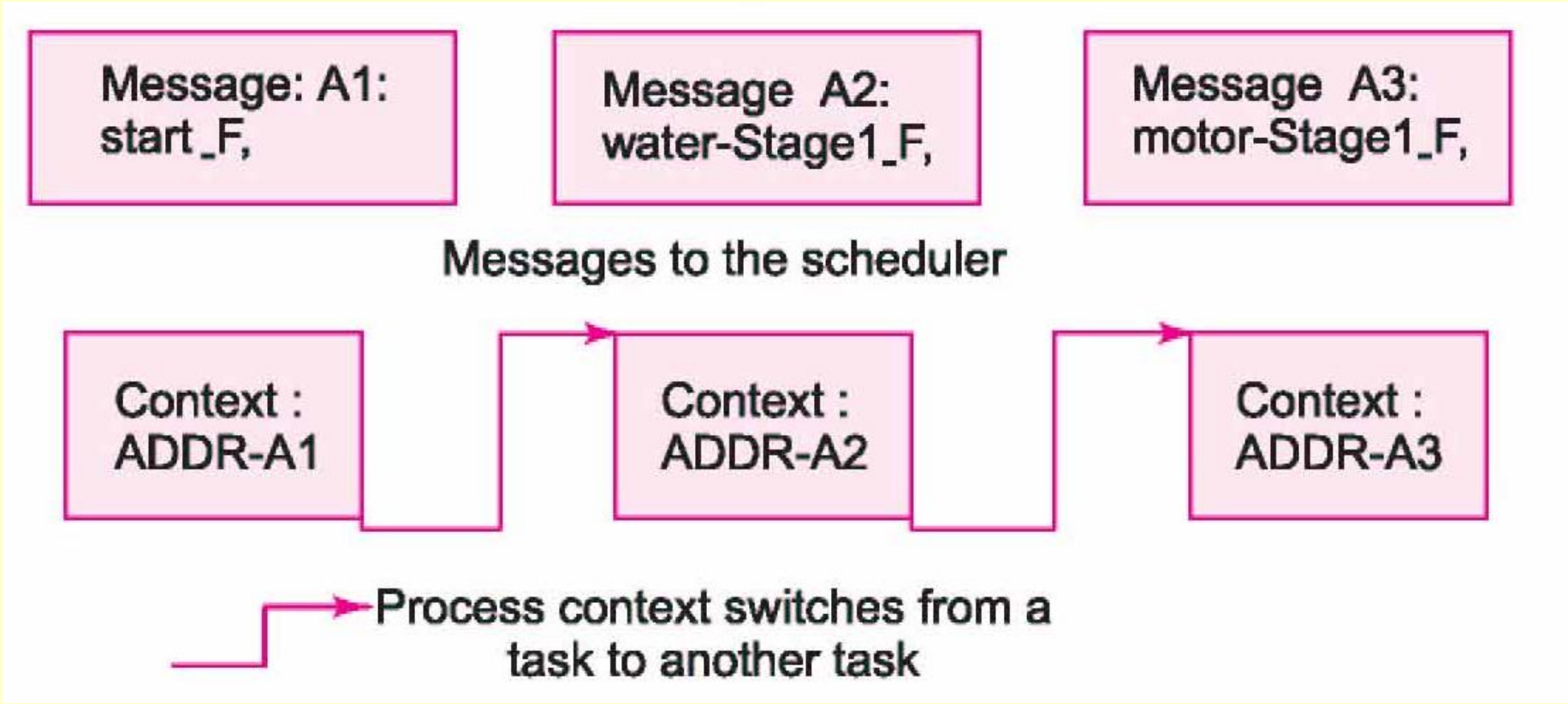
Cyclic scheduling model in tasks scheduling



First three tasks among N- tasks in washing machine tasks scheduling



Messages from the scheduler and task programs contexts at various instances in washing machine tasks scheduling



2. Cooperative Scheduling of Ready Tasks in List

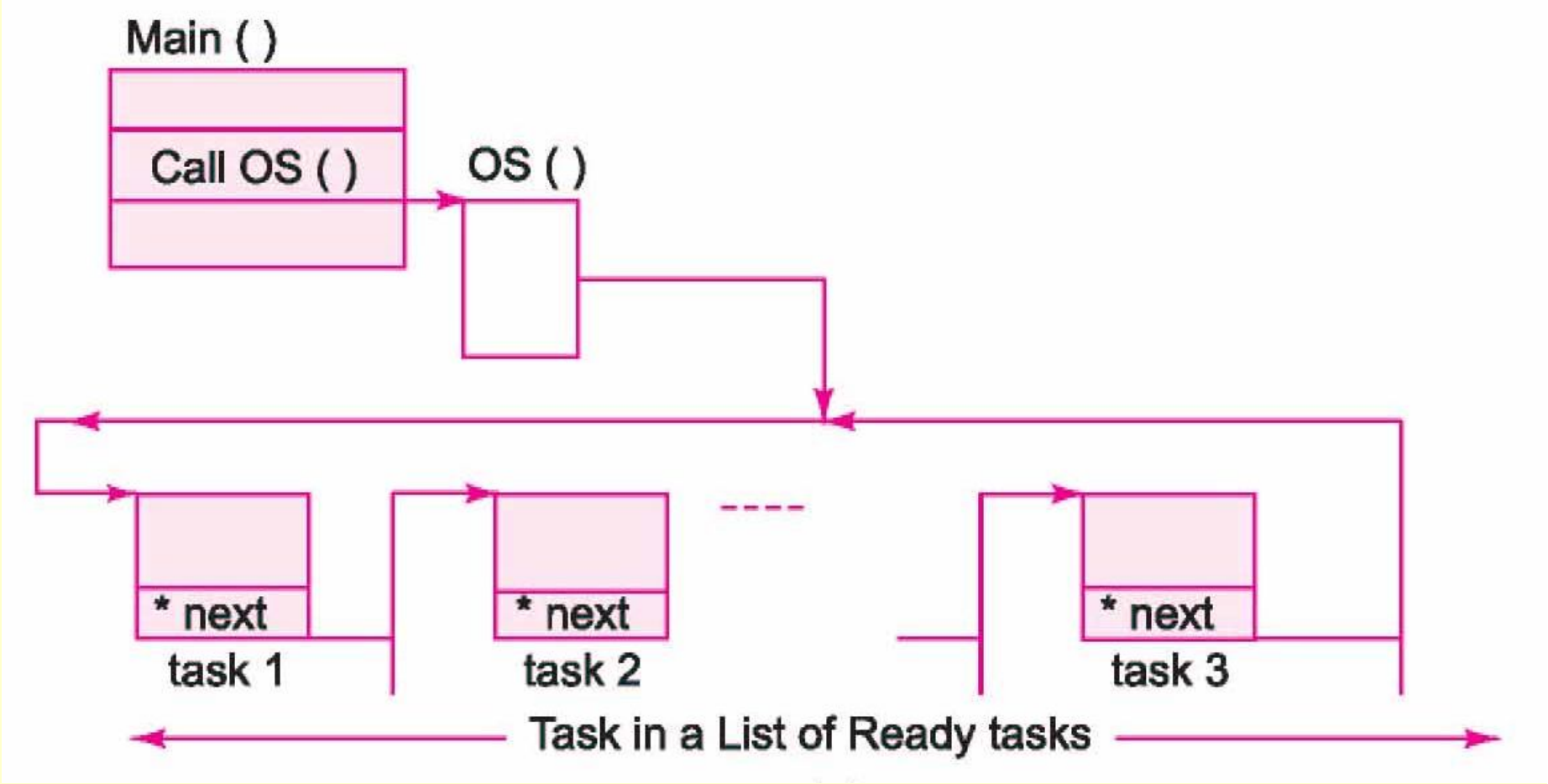
Cooperative Scheduling in the order in which a task is initiated on interrupt

- None of the tasks does block in-between anywhere during the ready to finish states.
- The service is in the order in which a task is initiated on interrupt.

Worst-case latency

- Same for every task in the ready list
- Tworst = $\{(dt_i + st_i + et_i)_1 + (dt_i + st_i + et_i)_2 + ... + (dt_i + st_i + et_i)_{n-1} + (dt_i + st_i + et_i)_n\} + t_{ISR}$.
- t_{ISR} is the sum of all execution times for the ISRs
- For an i-th task, let the event detection time with when an event is brought into a list be is dt_i , switching time from one task to another be is st_i and task execution time be is et_i
- i = 1, 2, …, n –1 , n

Scheduler in which the scheduler inserts into a list the ready tasks for a sequential execution in a cooperative mode



3. Cooperative Scheduling of Ready Tasks Using an Ordered List as per precedence Constraints

Cooperative Scheduling in the order of Ready Tasks using an Ordered List as per priority precedence

- Scheduler using a priority parameter, taskPriority does the ordering of list of the tasks– ordering according to the precedence of the interrupt sources and tasks.
- The scheduler first executes only the first task at the ordered list, and the total, equals the to period taken by the first task on at the list. It is deleted from the list after the first task is executed and the next task becomes the first.

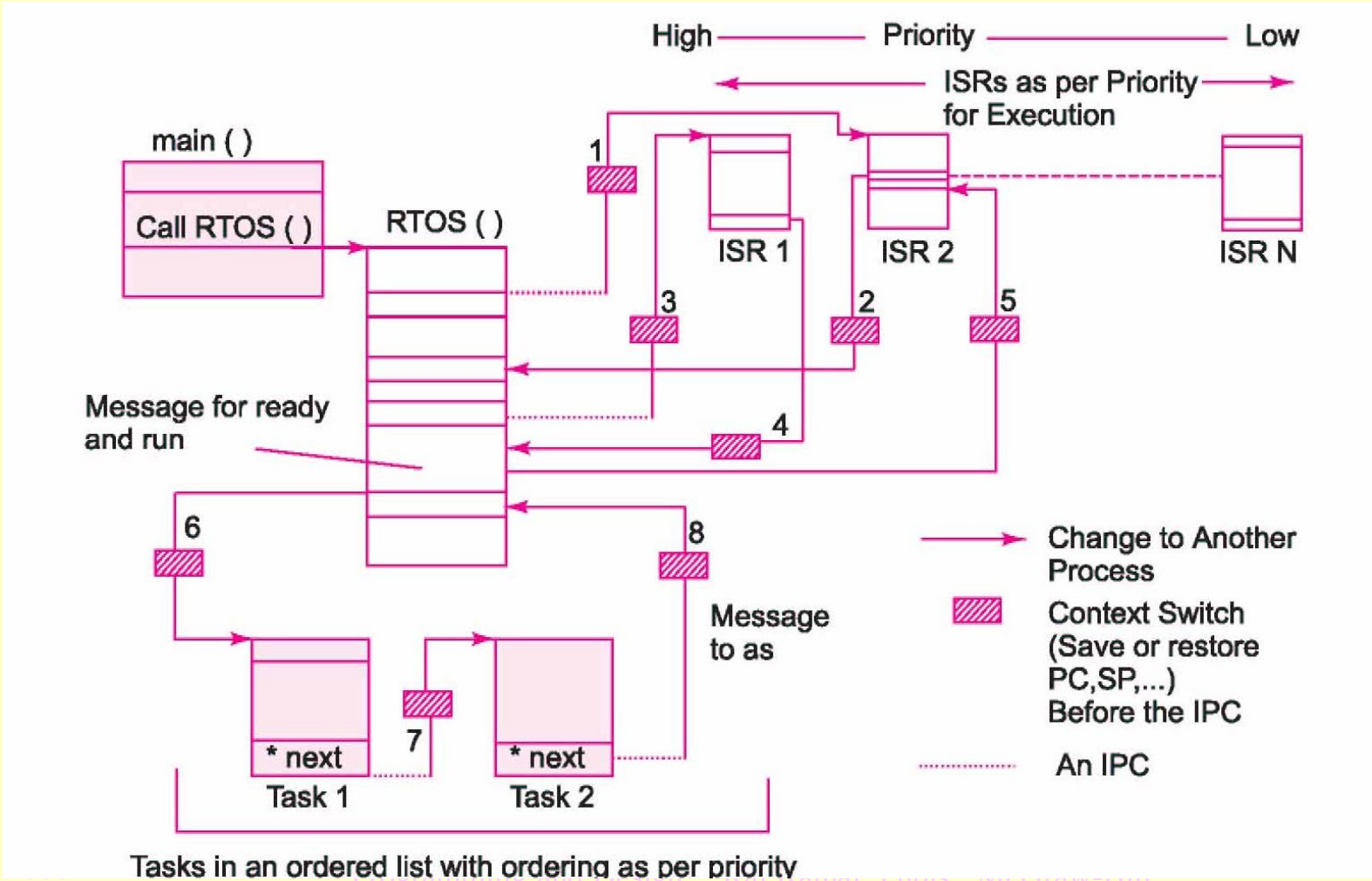
Cooperative Scheduling in the order of Ready Tasks using an Ordered List as per priority precedence

- The insertions and deletions for forming the ordered list are made only at the beginning of the cycle for each list.

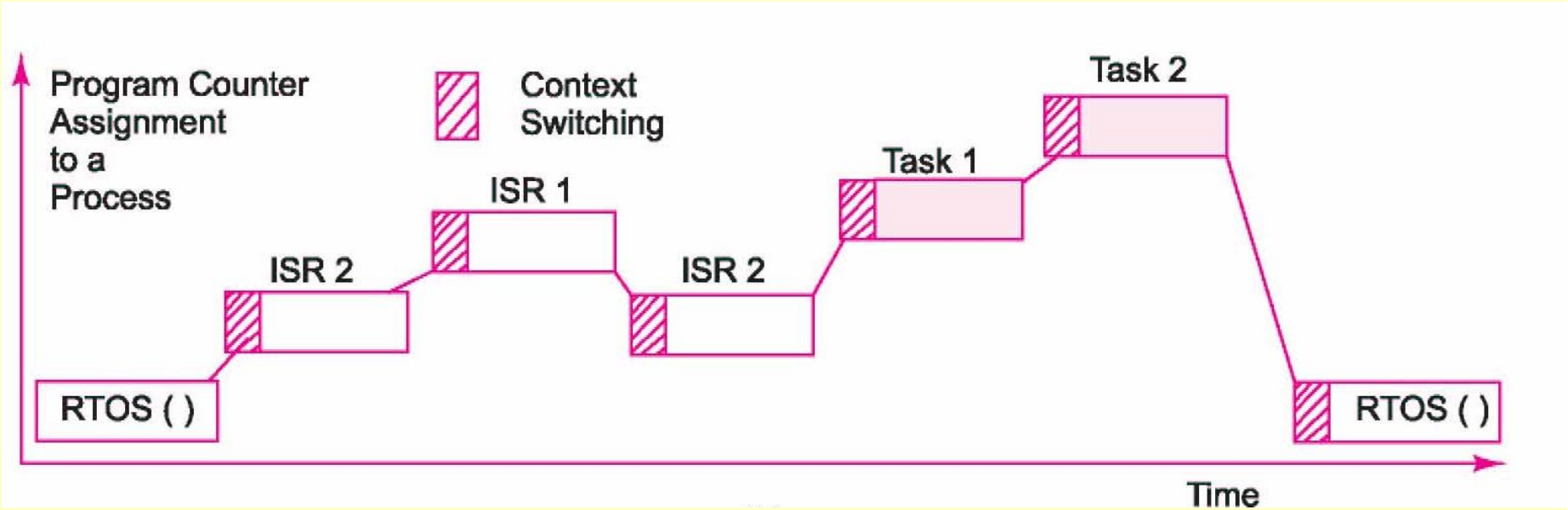
Worst-case latency

- Not Same for every task. Varies from $(dt_i + st_i + et_i)_{p(m)}\} + t_{ISR}$ to $\{(dt_i + st_i + et_i)_{p1} + (dt_i + st_i + et_i)_{p2} + ... + (dt_i + st_i + et_i)_{p(m-1)} + (dt_i + st_i + et_i)_{p(m)}\} + t_{ISR}$.
- t_{ISR} is the sum of all execution times for the ISRs
- For an i-th task, let the event detection time with when an event is brought into a list be is dt_i , switching time from one task to another be is st_i and task execution time be is et_i
- i = 1, 2, …, m –1 , m; m is number of ISRs and tasks in the list

Cooperative Priority based scheduling of the ISRs executing in the first layer and Priority based ready tasks at an ordered list executing in the second layer



Program counter assignments at different times on the scheduler calls to the ISRs and the corresponding tasks



Example of ACVM

- First the coins inserted by the user are read, then the chocolate delivers, and then display task displays ‘thank you, visit again’ message.
- Each task cooperates with other to finish.
- The precedence of tasks in the ready list─ reading coins is highest, then of chocolate delivery and display for the ordered list of the ready tasks.

Summary

We learnt

- Each task cooperate to let the running task finish in cooperative scheduling model
- Cyclic order
- Cyclic in list of initiated tasks into a ready list
- Cyclic in ordered list of initiated tasks into a ready list and list arranged in order of ISR and task priorities

End of Lesson 16 of Chapter 8