REAL TIME OPERATING SYSTEMS

Lesson-18:

Round Robin Time Slicing of tasks of equal priorities

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

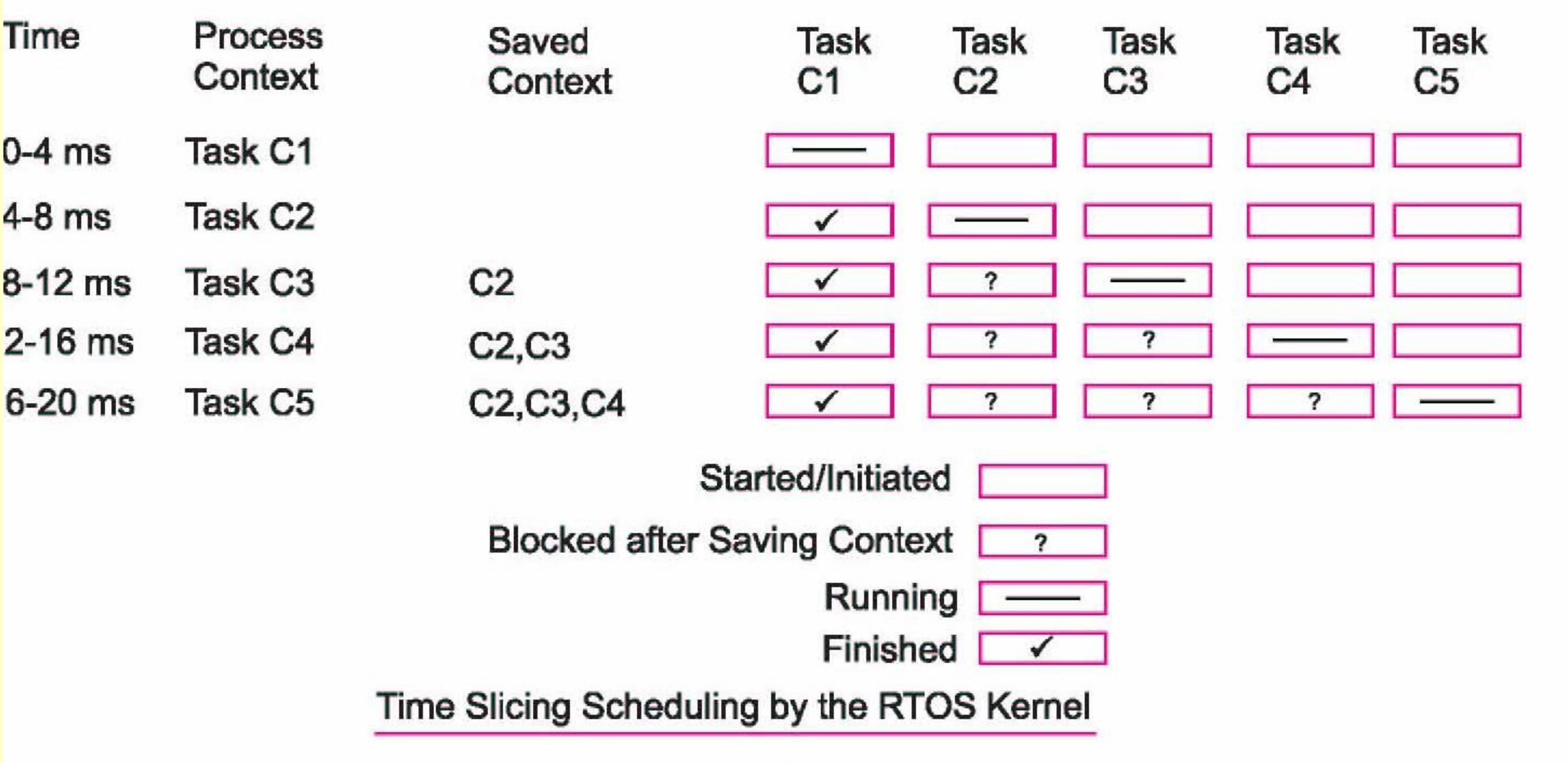
- Rate Monotonic Scheduling using ‘higher rate of events occurrence First’ precedence
- Fixed Times Scheduling
- Scheduling of Periodic, sporadic and aperiodic Tasks
- Advanced scheduling algorithms using the probabilistic Timed Petri nets (Stochastic) or Multi Thread Graph for the multiprocessors and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

2. Round Robin Time Slice Scheduling of Equal Priority Tasks

Equal Priority Tasks

- Round robin means that each ready task runs turn by in turn only in a cyclic queue for a limited time slice.
- Widely used model in traditional OS.
- Round robin is a hybrid model of clock-driven model (for example cyclic model) as well as event driven (for example, preemptive)
- A real time system responds to the event within a bound time limit and within an explicit time.

Tasks programs contexts at the five instances in the Time Scheduling Scheduler for C1 to C5



Time	Process Context	Saved Context	Task C1	Task C2	Task C3	Task C4	Task C5
0-4 ms	Task C1		—	□	□	□	□
4-8 ms	Task C2		✓	—	□	□	□
8-12 ms	Task C3	C2	✓	?	—	□	□
2-16 ms	Task C4	C2,C3	✓	?	?	—	□
6-20 ms	Task C5	C2,C3,C4	✓	?	?	?	—

Started/Initiated □

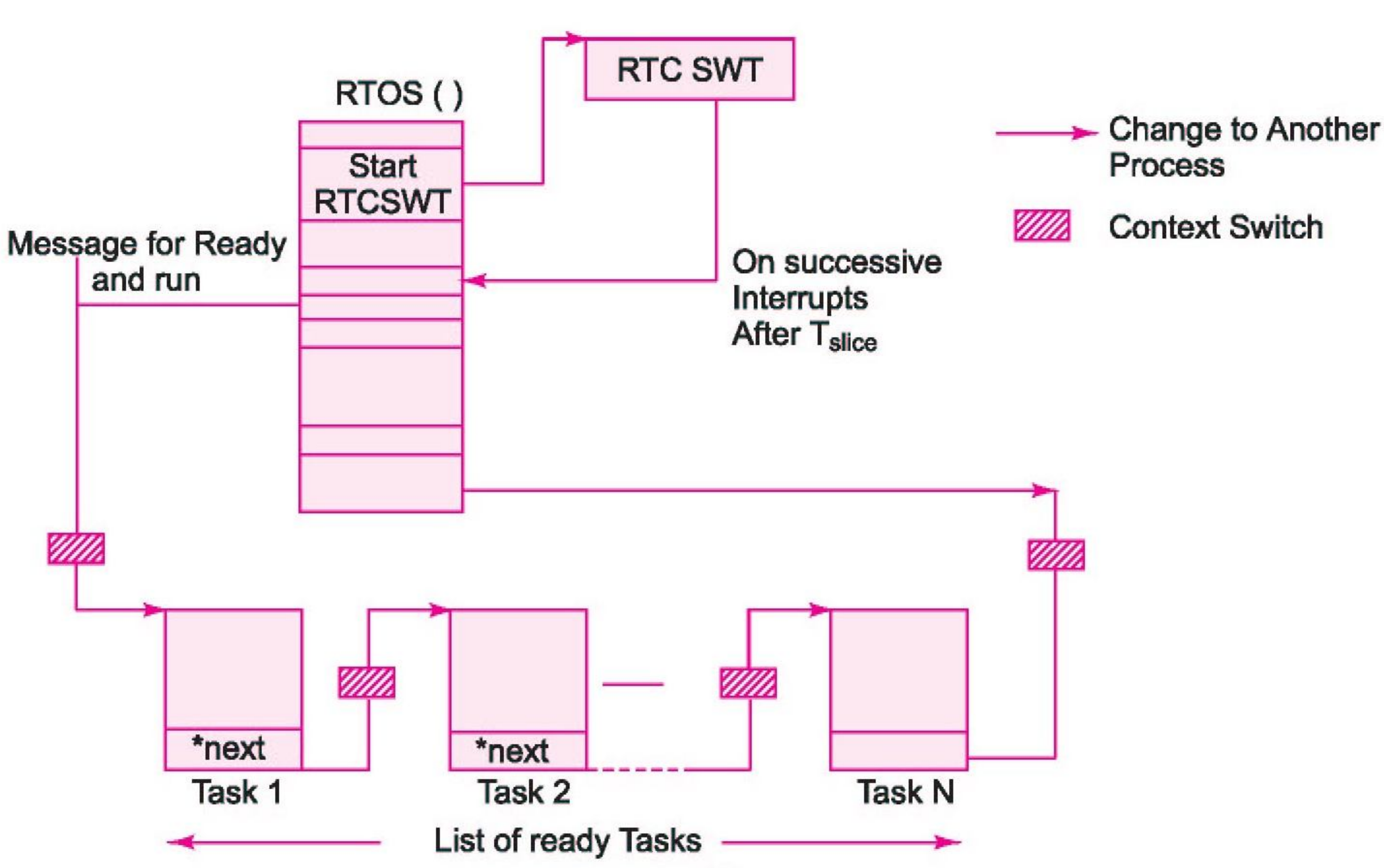
Blocked after Saving Context ?

Running —

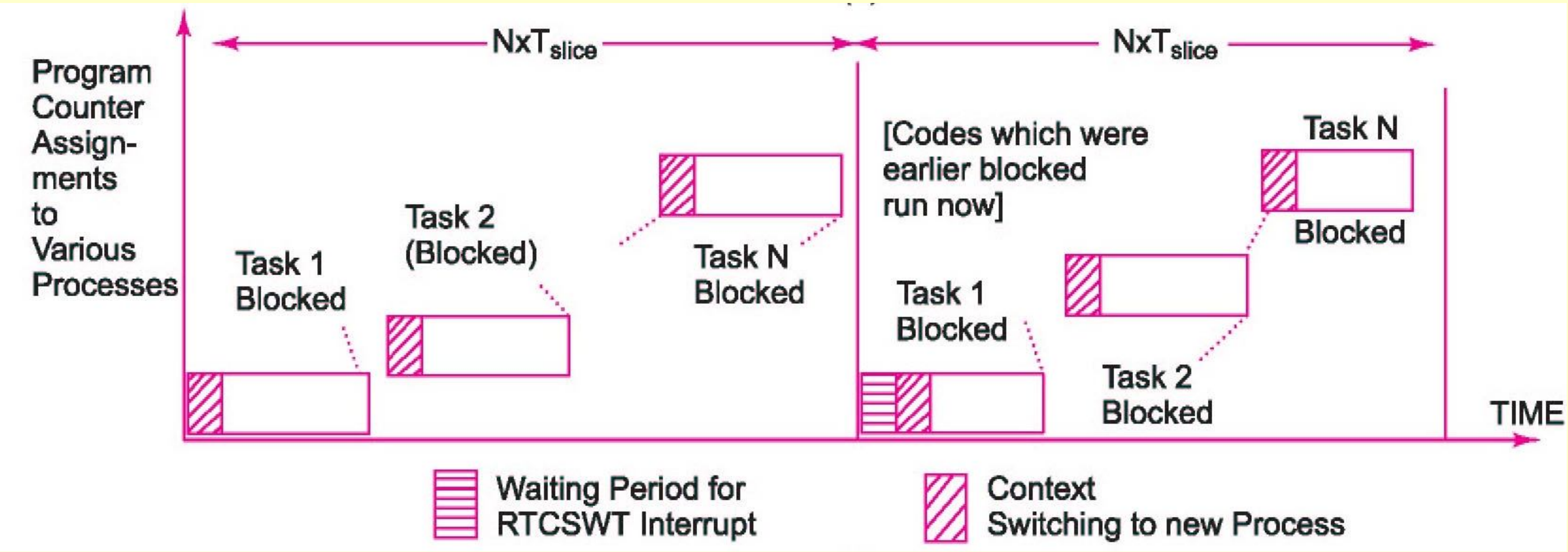
Finished ✓

Time Slicing Scheduling by the RTOS Kernel

Programming model for the Cooperative Time sliced scheduling of the tasks


RTOS ( )
RTC SWT
Start RTCSWT
Change to Another Process
▨ Context Switch
Message for Ready and run
On successive Interrupts After T_{slice}
*next
*next
Task 1
Task 2
Task N
List of ready Tasks

Program counter assignments on the scheduler call to tasks at two consecutive time slices. Each cycle takes time = N × t_{slice}


NxT_{slice}
NxT_{slice}
Program Counter Assign-ments to Various Processes
Task 1 Blocked
Task 2 (Blocked)
Task N Blocked
[Codes which were earlier blocked run now]
Task 1 Blocked
Task 2 Blocked
Task N
Blocked
TIME
Waiting Period for RTCSWT Interrupt
Context Switching to new Process

Case : $T_{cycle} = N \times T_{slice}$

- Same for every task = T_{cycle}
- $T_{cycle} = \{T_{slice}\ )\} \times N + t_{ISR}$.
- 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* and task execution time be is *et*
- Number of tasks = N

Worst-case latency

- Same for every task in the ready list
- Tworst = $\{N \times (T_{slice})\} + t_{ISR}$.
- t_{ISR} is the sum of all execution times for the ISRs
- i = 1, 2, …, N –1 , N

VoIP Tasks Example

- Assume a VoIP [Voice Over IP.] router.
- It routes the packets to N destinations from N sources.
- It has N calls to route.
- Each of N tasks is allotted from a time slice and is cyclically executed for routing packet from a source to its destination

Round Robin

- Case 1: Then each task is executed once and finishes in one cycle itself.
- When a task finishes the execution before the maximum time it can takes, there is a waiting period in-between period between two cycles.
- The worst-case latency for any task is then N × t_{slice}. A task may periodically need execution. A task The period for the its need of required repeat execution of a task is an integral multiple of t_{slice}.

Case 2: Alternative model strategy

- Case 2: Certain tasks are executed more than once and do not finish in one cycle
- *Decomposition* of a task that takes the abnormally long time to be executed.
- The decomposition is *into two or four or more tasks*.
- Then one set of tasks (or the odd numbered tasks) can run in one time slice, t'slice and the another set of tasks (or the even numbered tasks) in another time slice, t''slice.

Decomposition of the long time taking task into a number of sequential states

- *Decomposition of the long time taking task* into a number of sequential states or a number of node-places and transitions as in finite state machine. (FSM).
- Then its one of its states or transitions runs in the first cycle, the next state in the second cycle and so on.
- This task then reduces the response times of the remaining tasks that are executed after a state change.

Summary

We learnt

- Round robin time slice scheduling is for handling tasks with each task for a fixed time slice
- Longer tasks can be decomposed into more tasks to allocate more time slices to that

End of Lesson 18 of Chapter 8