Real Time Systems and Control Applications

Contents

Concepts of Real Time Scheduling

Real-Time Tasks

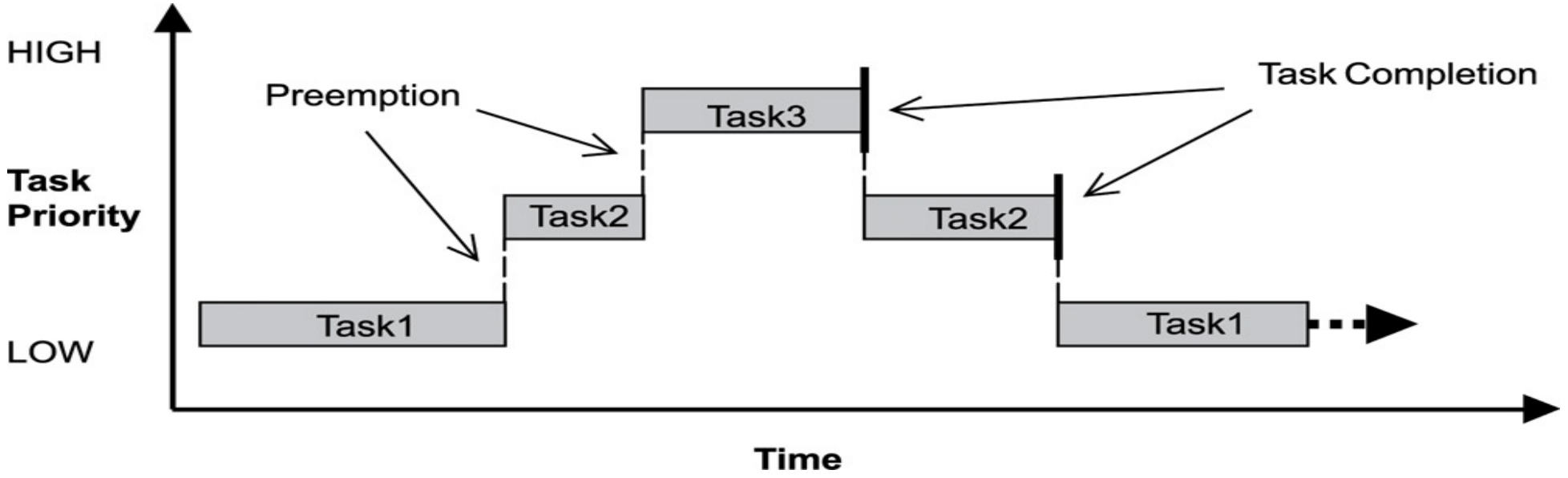
- In real-time systems, several tasks execute concurrently
- Each task has a real-time constraint

- Task Categories
  - Periodic Task --- inter-arrival time between two instances almost same
  - Sporadic Task --- inter-arrival time between consecutive instances differ widely. (Hard timing constraints)
  - Aperiodic Task --- inter-arrival time between consecutive instances differ widely. (Soft deadlines)

Preemptivity of Tasks

- A task is preemptable if its execution can be suspended any time to allow execution of other jobs.

- Priority Based Preemptive Scheduling



Temporal Parameters

- In order to meet the real time requirements of hard real-time tasks, it is assumed that many parameter of these tasks are known at all times. Some of these parameters are described below:
  - Number of Tasks (n)
  - Release Time or Arrival Time ($r_{i,j}$)
  - Absolute Deadline (d_i)
  - Relative Deadline (D_i) ➔ $d_i = r_{i,j} + D_i$
  - Execution Time (E_i)
  - Response Time (R_i)

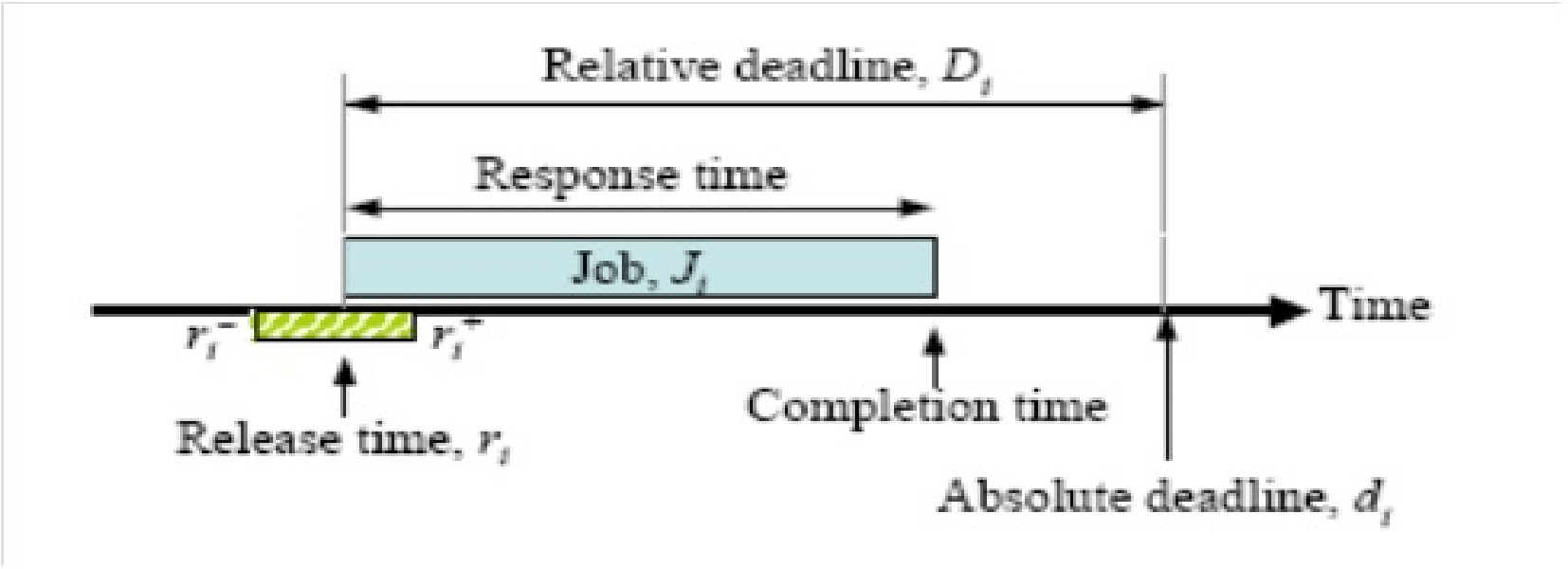
Number of Tasks

- In many embedded systems, the number of tasks is fixed as long as the system remains in an operation mode.

- In some systems the number of tasks may change as tasks are added or deleted while the system executes, still the number of tasks with hard timing constraints is known at all times.

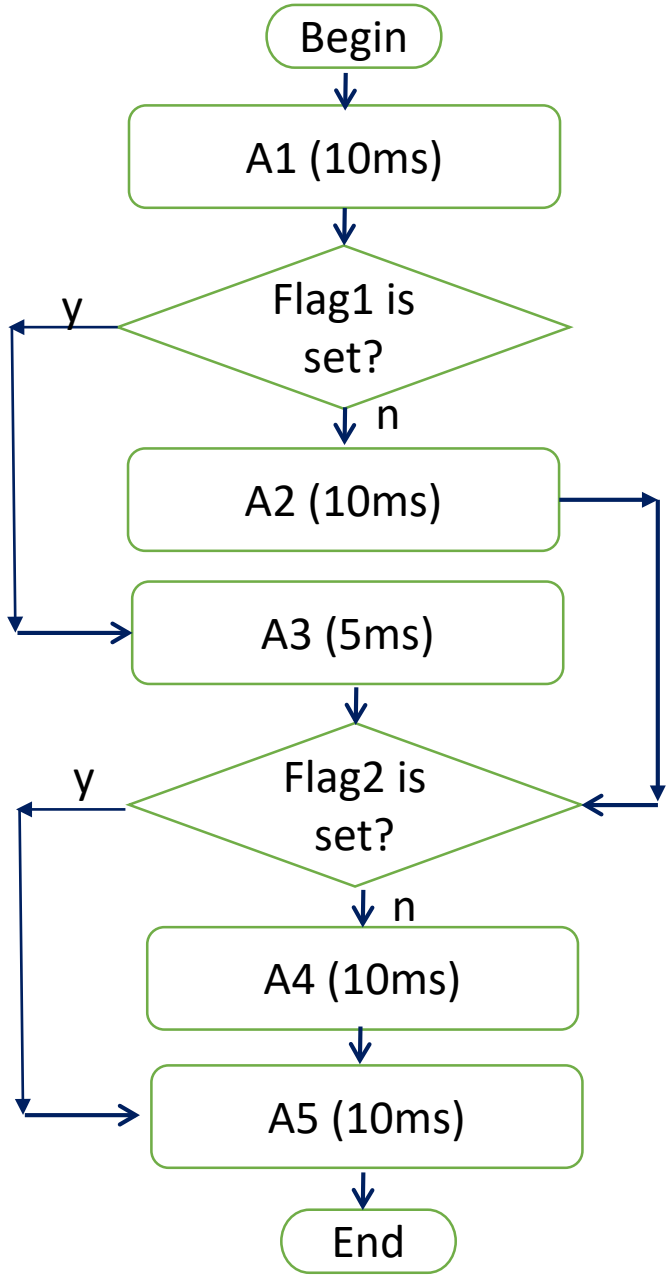
Relative Deadline v.s. Absolute Deadline

- **Relative deadline** is an interval when the **absolute deadline** is a moment in time at which the job must be completed.



Execution Time

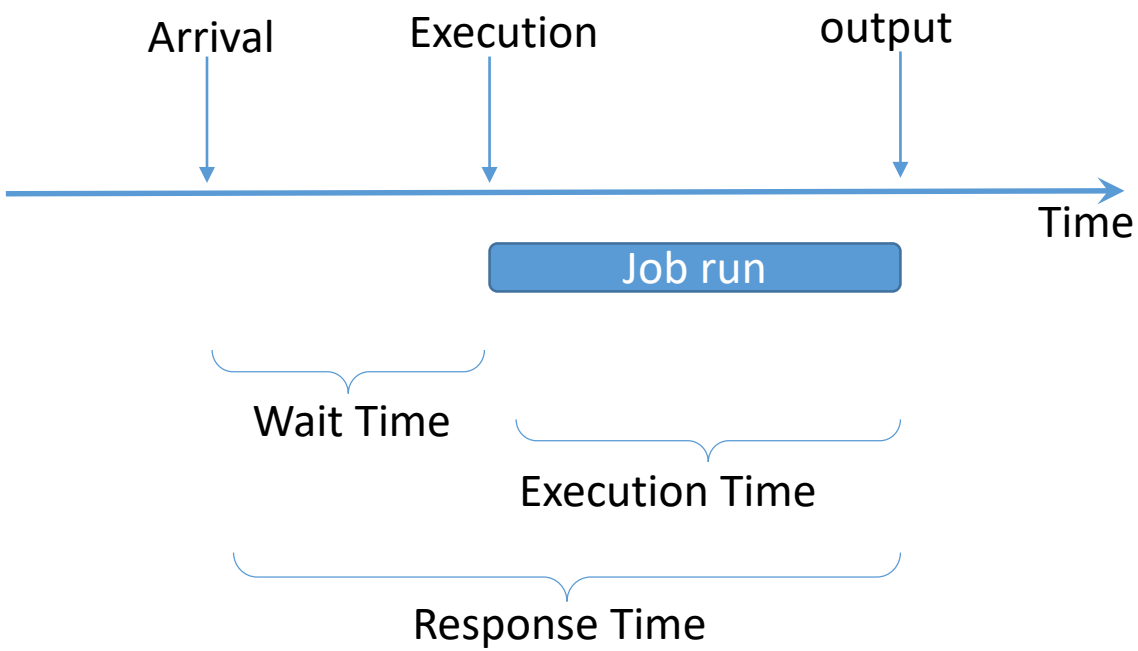
- Execution time with no other tasks sharing resources
  - depends on processor speed and complexity of instructions
  - variations due to cache, pipelining, software structures such as conditional branches
  - depends on which branch is taken
- Which time should be used?
  - Worst case execution time



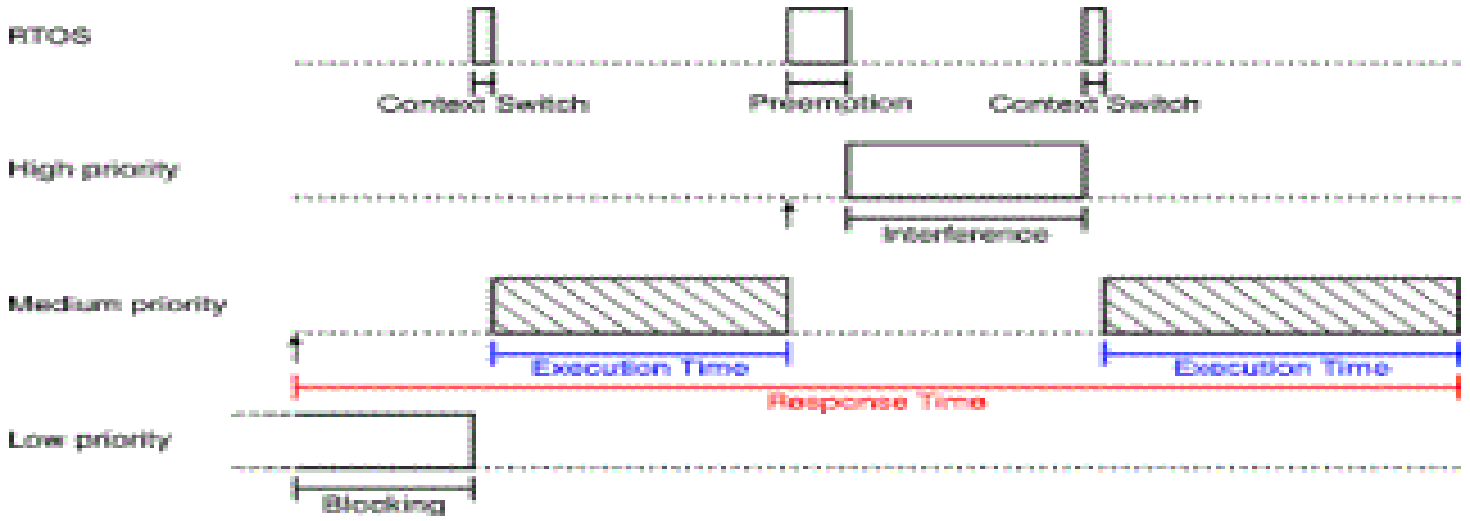
Response Time v.s. Execution Time

Response Time: The time span between the task activation and its completion.

Execution Time: The actual amount of time required by a job to complete its execution. It may vary for many reasons. What can be determined a priori through analysis and measurements is the max and min amounts of time to complete execution. It normally refers to the maximum time.



Response Time When the Execution Is Preempted

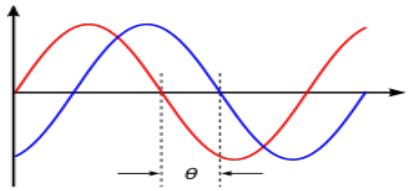


Periods and Phases of Periodic Tasks

- A **Period** (p_i) of a periodic task T_i is the minimum length of all time intervals between release times of consecutive tasks.

- **Phase** of a Task (ϕ_i) : The **release time $r_{i,1}$** of a task T_i is called the phase of T_i $(\phi_i = r_{i,1})$.

- The first instances of several tasks may be released simultaneously. They are called in phase and have a zero phase.



Typical Task Model for Periodic Tasks

- All tasks in a task set are strictly periodic.
- The relative deadline of a task is equal to its period (if not specified).
- All tasks are independent, i.e. there are no precedence constraints.
- No task has any non-preemptible section and the cost of preemption is negligible.
- CPU processing requirements are significant; memory and I/O requirements are negligible.

Representation of Periodic Tasks

- A periodic task Ti can be represented by a 4 tuple: $(\emptyset_i, P_i, e_i, D_i)$
- If using 3 tuple (P_i, e_i, D_i), it is equivalent to $(0, P_i, e_i, D_i)$
- If using 2 tuple (P_i, e_i), it is equivalent to $(0, P_i, e_i, P_i)$
- Example: A system with two real-time tasks (3, 1) and (5, 2).

CPU Utilization

- It is a measure of the percentage of non-idle processing, denoted as U. It is calculated by summing the contribution of utilization factors for each (periodic or aperiodic) task. The utilization factor u_i for a task T_i with execution time e_i, and period p_i is given by:

$$u_i = {e_i}/{p_i}$$

- And for a system with *n* tasks the overall system utilization is

$$U = \sum_{i=1}^{n} u_i = \sum_{i=1}^{n} {e_i}/{p_i}$$

End