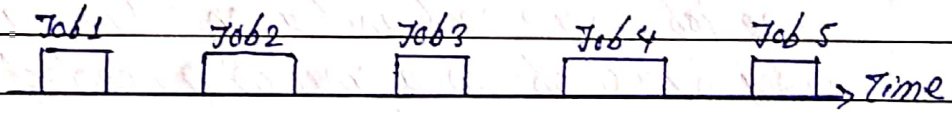
Unit 1-2

Hard versus soft real time systems

Jobs and Task

→ A job is a unit of work that is scheduled and executed by a system.
E.g. Computation of a control-law, transmission of a data packet, retrieval of a file.
→ Every job executes on some resources. For e.g. the jobs mentioned above execute on a CPU, a network, and a disk, respectively.
→ A job J_i is characterized by many parameters: execution time, deadlines, preemptivity, resource requirements.

→ A task is a set of related jobs which jointly provide some system function
E.g. The set of jobs that constitute the "maintain constant altitude" task, keeping an airplane flying at a constant altitude.



Processors

→ A processor, P, is an active component on which jobs are scheduled.
E.g. - Threads scheduled on a CPU
- Data scheduled on a transmission link
- Read/write requests scheduled to a disk
- Transaction scheduled on a database server.
→ Each processor has a speed attribute which determines the rate of progress a job makes towards completion.

→ Two processors are of the same type if they are functionally identical and can be used interchangeably.
- Two transmission links with the same transmission rate between a pair of sender and receiver are processors of the same type.

Resources

→ A resource, R, is a passive entity upon which jobs may depend. For e.g. memory, sequence numbers, mutexes, database locks etc.
→ Resources have different types and sizes, but do not have a speed attribute.
→ If the system contains s ("rho") types of resources, this means:
- There are s different types of serially reusable resources.
- There are one or more units of each type of resource, only one job can use each unit at once (mutually exclusive access)
- A job must obtain a unit of a needed resource, use it, and then release it.

→ A resource is plentiful if no job is ever prevented from executing by the unavailability of units of the resource.

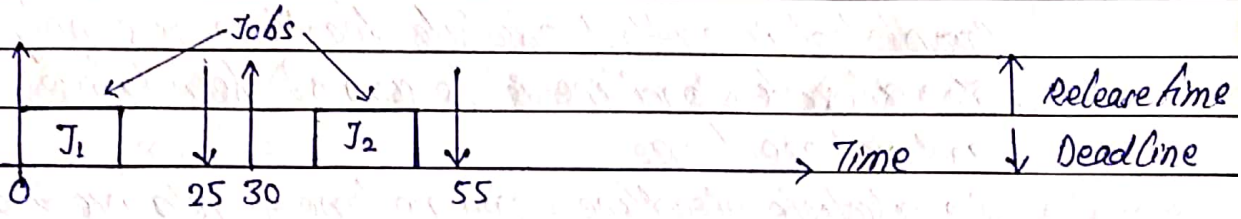
Execution Time

→ Execution time e_i is the amount of time required to complete job J_i when it executes alone and has all resources it takes.
→ The value of execution time e_i depends upon the

Complexity of job and speed of the processor on which it is scheduled
→ The execution time falls into an interval $[e_i^-, e_i^+]$.

Release Time (Also called arrival time)

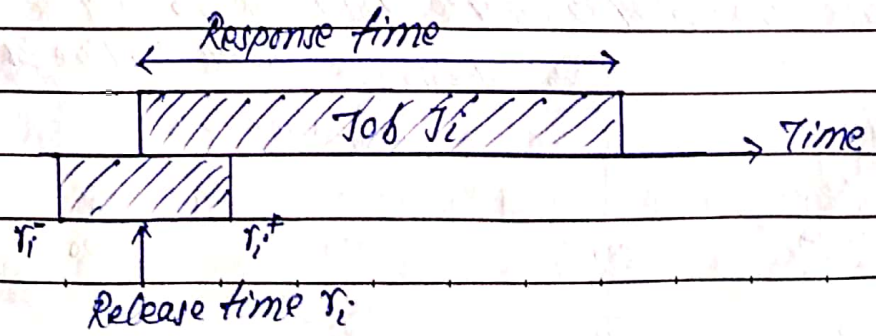
→ The release time of a job is the instant of time at which job becomes available for execution.
→ It may not be exact: Release time jitter so r_i is in the interval $[r_i^-, r_i^+]$.
→ The job can be scheduled and executed at any time at or after its release time whenever its data and control dependency conditions are met.



Response Time

→ The response time is the length of time from the release time of the job to the time instant when it completes.
Response time = Job completion time − Job release time

→ The response time may not be the same as execution time because the job may not execute continuously.



Deadlines

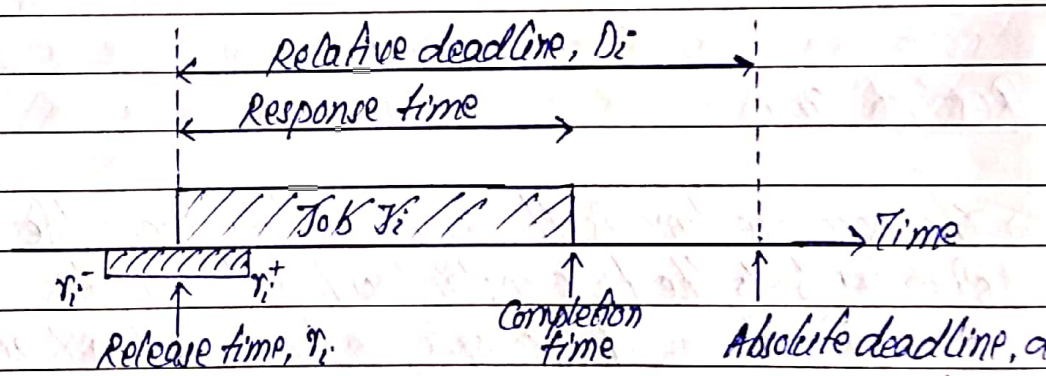
The deadline of a job is the instant of time by which its execution is required to be completed.

→ Deadline time is divided into the two categories:

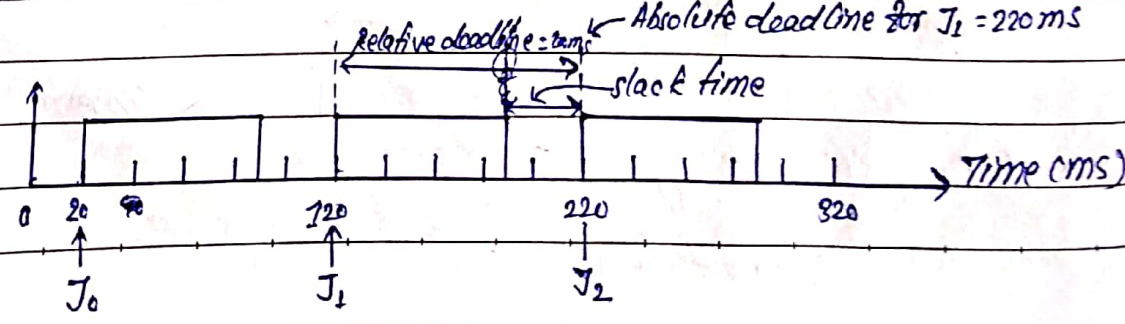
1. Relative Deadline:
   - The maximum allowable response time of a job is called a relative deadline.
   - It is the time interval bet^n the start of the job (release time) and the instant at which the deadline occurs.

2. Absolute Deadline:
   - The instant of time for which a job is required to be completed is called absolute deadline or simply a deadline.
   - The absolute deadline is the sum of the relative deadlines and release time.

   i.e. absolute deadline = release time + relative deadline



The feasible interval for a job, J_i is the interval between the release time and absolute deadline i.e. $[r_i; d_i]$



Suppose each job must complete before release of next job.
- It's relative deadline is 100 ms
- It's absolute deadline is $20 + ((i+1) * 100)$ ms.

Timing Constraints

A constraint imposed on the timing behavior of a job is called a timing constraint. It may be:
- Simple timing constraints: can be specified in terms of its release time and relative or absolute deadlines.
- Complex timing constraints: cannot be specified conveniently in terms of release times and deadlines.

Timing constraints into two types: Hard and soft
They are defined based on
- Functional criticality of jobs
- Usefulness of late results and
- Deterministic or probabilistic nature of the constraints

~~Hard Timing Constraints~~

Based on Functional criticality of Jobs

Hard:
A timing constraint or deadline is hard if the failure to meet if is considered to be a fatal fault.
E.g - a late command to stop a train may cause a collision
- a bomb dropped too late may hit a civilian population instead of the intended military target.

Soft:

A timing constraint or deadline is soft if a few misses of deadlines do no serious harm but only the system's overall performance becomes poor.
- The late completion of a job that has a soft deadline is undesirable.

Based on usefulness of late results

~~Hard~~: Usefulness of result is the function of the tardiness of jobs.

Tardiness = Completion time - Deadline
- Tardiness is zero if the job completes at or before its deadline.
- If the job is late, the tardiness of the job increases gradually

Soft:

The usefulness of a result produced by a soft real-time job decreases gradually as the tardiness of the job increases.

Hard:

The usefulness of a result ~~produce~~ falls off abruptly and may even become negative when the tardiness of the job becomes larger than zero.

Based on Deterministic or Probabilistic nature of the constraints

Hard:

If a job must never miss its deadline, then the deadline is hard.
- Hard timing constraints are deterministic in nature.

Soft:

If its deadline can be missed occassionally with some acceptably low probability, then its timing constraint is soft.
- Soft timing constraints are probabilistic in nature.

Hard Timing Constraints and Temporal Quality-of-service Guarantees

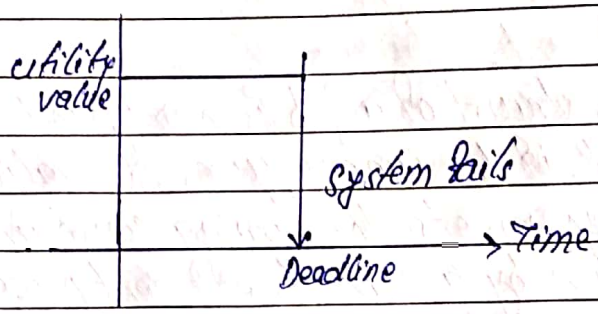
The timing constraint of a job is hard, and the job is a hard real-time job, if the user requires the validation that the system always meets the timing constraint. validation means a demonstration by a correct, efficient procedure or by exhaustive simulation and testing.
On the other hand, If no validation is required, or only a demonstration that the job meets some statistical constraint (i.e. timing constraint specified in terms of statistical averages) ~~is~~ suffices, then the timing constraint of the job is soft.

The temporal quality of service measures the systems in terms of different parameters like response time, jitter etc.
→ If the user wants the temporal quality of the service provided by a task are guaranteed and the satisfication of the timing constraints defining the temporal quality are validated, then the timing constraints are hard.
→ On the other hand, If the user wants the best quality of service that the system can provide but allows the system to deliver qualities only below what is defined by the timing constraints, then the timing constraints are soft.

→ Distinction bet^ hard & soft timing Constraints based on Guaranteed and best-effort services.

Hard Real-Time System

If a job must never misses its deadline then the system is called hard real-time system. For a hard real-time system, every deadline must be hit. In a real hard real-time system, if the system fails to hit the deadline even once the system is said to have failed.



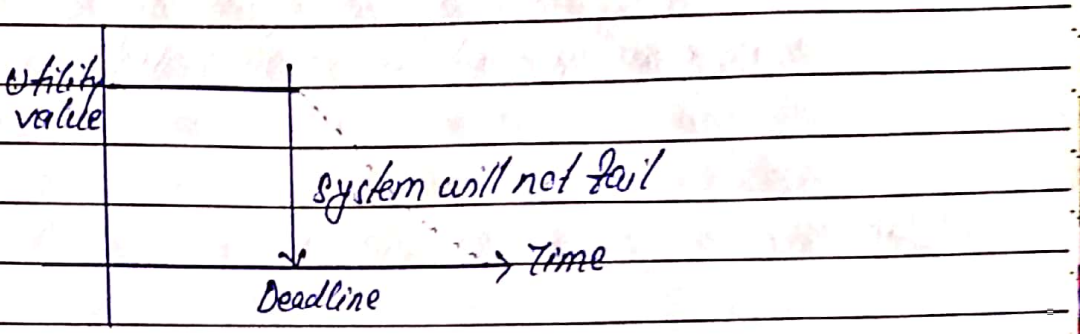
e.g.: flight control, manufacturing control, pacemakers, anti-lock brakes, railway signaling, robot.

Characteristics of hard real time system:
1. The hard real time system is called guaranteed services.
2. Response time is hard.
3. often safety critical.
4. Size of data files are small and medium.
5. Peak load performance is predictable.
6. Use the autonomous error detection.
7. Have short-term data integrity.
8. Controlled by environment.
9. User can obtain the validation.

Soft Real-Time system

If some deadline can be missed occasionally acceptably with low probability then the system is called soft real time system. In a soft real time system, even if

the system fails to meet the deadline one or more than once, the system is still not considered to have failed.



E.g; On-line transaction systems, telephone switches (Mobile network), multimedia system, web browsing, DVD player, streaming audio-video, stoke price quotation system, electronic games.

Some characteristics of soft real time system:

1. The soft real-time system is called best effort service.
2. Response time is soft.
3. Non-critical safety.
4. Size of data files are large
5. Peak load performance is degradable.
6. Use user assisted error detection.
7. Have long-term data integrity.
8. Controlled by computer
9. Not always obtain the validation.

Jayanta Poudel