2. Define and describe Multirate Systems. (10) [CO3-L1]

Implementing code that satisfies timing requirements is even more complex when multiple rates of computation must be handled. **Multirate** embedded computing systems are very common, including automobile engines, printers, and cell phones. In all these systems, certain operations must be executed periodically, and each operation is executed at its own rate.

Timing Requirements on Processes

- Processes can have several different types of timing requirements imposed on them by the application. The timing requirements on a set of processes strongly influence the type of scheduling that is appropriate. A scheduling policy must define the timing requirements that it uses to determine whether a schedule is valid. Before studying scheduling proper, we

outline the types of process timing requirements that are useful in embedded system design.

- Figure 3.2 illustrates different ways in which we can define two important requirements on processes: **release time** and **deadline**.

- The release time is the time at which the process becomes ready to execute; this is not necessarily the time at which it actually takes control of the CPU and starts to run. An aperiodic process is by definition initiated by an event, such as external data arriving or data computed by another process.
- The release time is generally measured from that event, although the system may want to make the process ready at some interval after the event itself. For a periodically executed process, there are two common possibilities.
- In simpler systems, the process may become ready at the beginning of the period. More sophisticated systems, such as those with data dependencies between processes, may set the release time at the arrival time of certain data, at a time after the start of the eriod.

- A deadline specifies when a computation must be finished. The deadline for an aperiodic process is generally measured from the release time, since that is the only reasonable time reference. The deadline for a periodic process may in general occur at some time other than the end of the period.

- Rate requirements are also fairly common. A rate requirement specifies how quickly processes must be initiated.

- The **period** of a process is the time between successive executions. For example, the period of a digital filter is defined by the time interval between successive input samples.

- The process's **rate** is the inverse of its period. In a multirate system, each process executes at its own distinct rate.

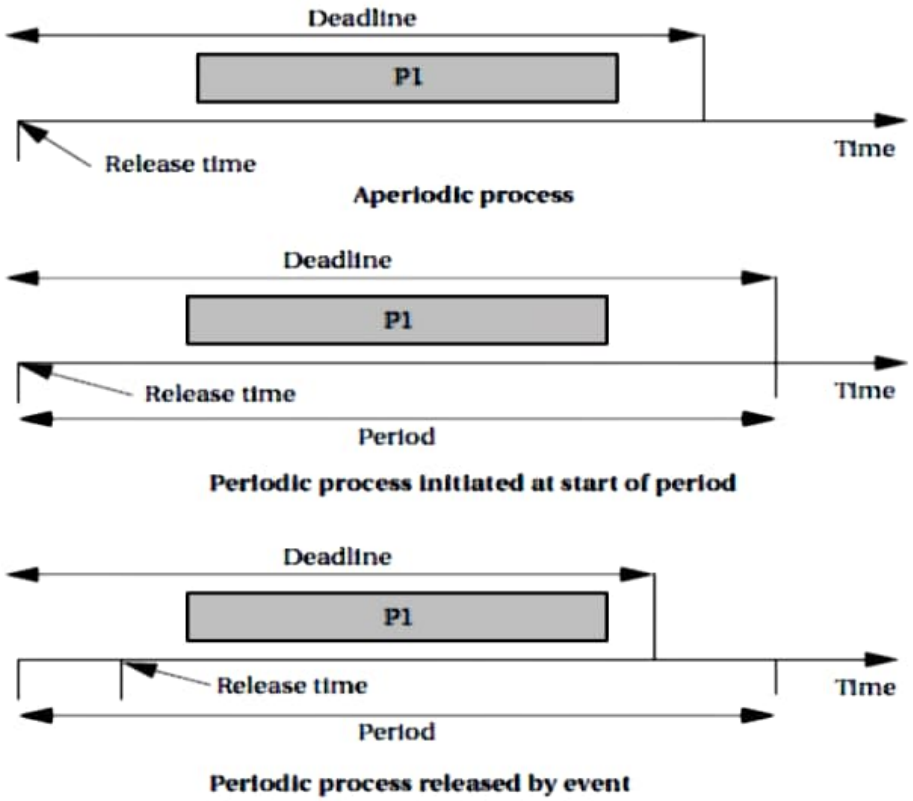

Fig 3.2 Example definitions of release times and deadlines.

- The most common case for periodic processes is for the initiation interval to be equal to the period. However, pipelined execution of processes allows the initiation interval to be less than the period. Figure 3.3 illustrates process execution in a system with four CPUs.

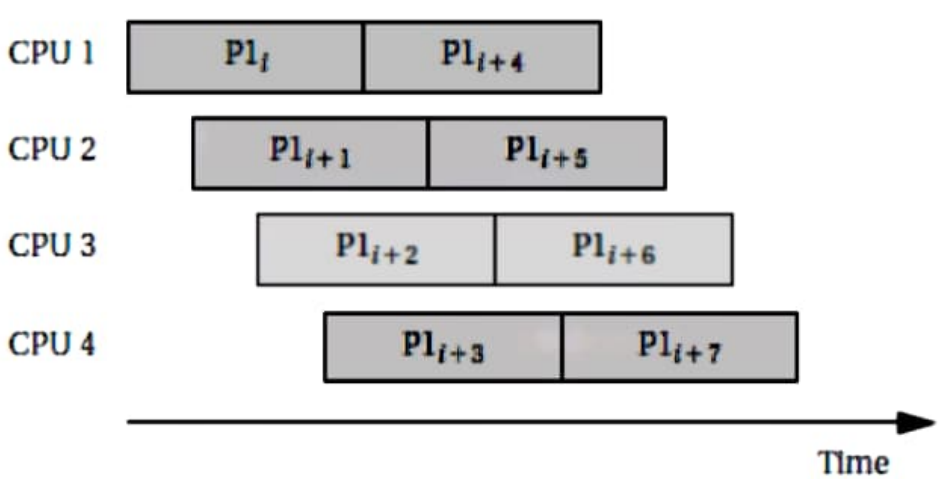

Fig 3.3 A sequence of processes with a high initiation rate.

CPU Metrics

We also need some terminology to describe how the process actually executes. The **initiation time** is the time at which a process actually starts

executing on the CPU. The **completion time** is the time at which the process finishes its work.

The most basic measure of work is the amount of **CPU time** expended by a process. The CPU time of process i is called Ci . Note that the CPU time is not equal to the completion time minus initiation time; several other processes may interrupt execution. The total CPU time consumed by a set of processes is

$$T = \sum T_i \quad (3.1)$$

We need a basic measure of the efficiency with which we use the CPU. The simplest and most direct measure is **utilization**:

U=CPU time for useful work/total available CPU time

(3.2) Utilization is the ratio of the CPU time that is being used for useful computations to the total

available CPU time. This ratio ranges between 0 and 1, with 1 meaning that all of the available CPU time is being used for system purposes. The utilization is often expressed as a percentage. If we measure the total execution time of all processes over an interval of time t, then the CPU utilization is

$$U = T/t. \quad (3.3)$$