Rate monotonic scheduling is a priority algorithm that belongs to the static priority scheduling category of Real Time Operating Systems. It is preemptive in nature. The priority is decided according to the cycle time of the processes that are involved. If the process has a small job duration, then it has the highest priority. Thus if a process with highest priority starts execution, it will preempt the other running processes. The priority of a process is inversely proportional to the period it will run for.
A set of processes can be scheduled only if they satisfy the following equation :

$$\sum_{k=1}^{n} \frac{Ci}{Ti} \leq U = n\left(2^{1/n} - 1\right)$$

Where n is the number of processes in the process set, Ci is the computation time of the process, Ti is the Time period for the process to run and U is the processor utilization.

Example:
An example to understand the working of Rate monotonic scheduling algorithm.

Processes	Execution Time (C)	Time period(T)
P1	3	20
P2	2	5
P3	2	10

```
n( 2^1/n - 1 ) = 3 ( 2^1/3 - 1 ) = 0.7977
```

```
U = 3/20 + 2/5 + 2/10 = 0.75
```

It is less than 1 or 100% utilization. The combined utilization of three processes is less than the threshold of these processes that means the above set of processes are schedulable and thus satisfies the above equation of the algorithm.

1. **Scheduling time –**
   For calculating the Scheduling time of algorithm we have to take the LCM of the Time period of all the processes. LCM (20, 5, 10) of the above example is 20. Thus we can schedule it by 20 time units.
2. **Priority –**
   Rate Monotonic:
   The priority will be the highest for the process which has the least running time period. Thus P2 will have the highest priority, after that P3 and lastly P1.

Process	Capacity	Period	Deadline
P1	3	20	7
P2	2	5	4
P3	2	10	8

P2(5) > P3(10) > P1 (20)

Deadline Monotonic:

The priority will be the highest for the process which has the earliest deadline first.

P2(4) >P1(7) >P3 (8)

Earliest Deadline First

The priority will be given to the next immediate deadline in the table of P1,P2,P3

Rate Monotonic Algorithm

Step I: feasibility check:

$$\sum_{K=1}^{n} \frac{C_i}{T_i} \le U = n\left(2^{1/n} - 1\right)$$

$C_i \rightarrow$ Capacity; $n \rightarrow$ no. of processes

$T_i \rightarrow$ Period; $U \rightarrow$ utilization

Given:-

Process	Capacity	Period
T_1	3	20
T_2	2	05
T_3	2	10

$$\sum_{K=1}^{n} \frac{C_i}{T_i} = \frac{3}{20} + \frac{2}{05} + \frac{2}{10} = 0.75$$

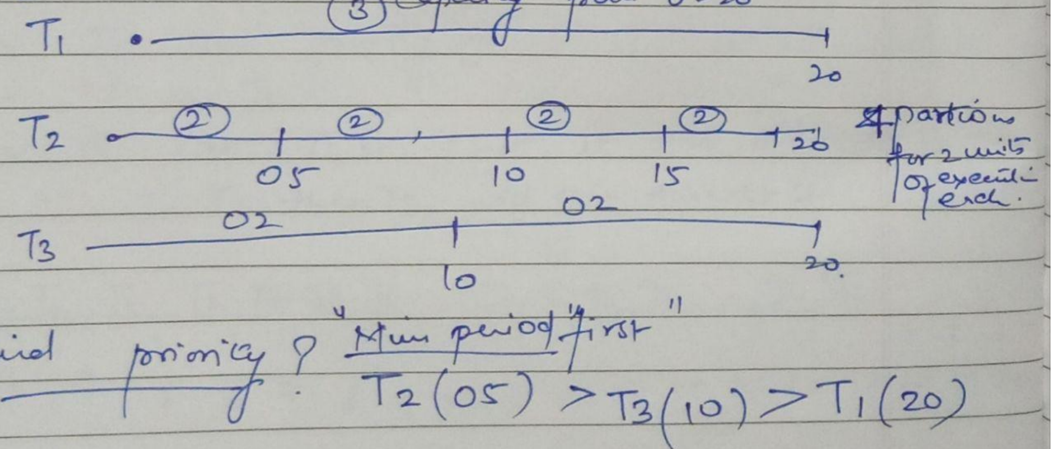
$$n\left(2^{1/n} - 1\right) = 3\left(2^{1/3} - 1\right) = 0.7977$$

Step II: find L.C.M of period

LCM (20, 05, 10) = 20

Step III:- schedule all processes as per time line = 20 and period as per respective process.

i.e.



find priority? "Min period first"

$T_2(05) > T_3(10) > T_1(20)$

PAGE No.

DATE / /

1) T_2 appears every 5 time stamp with highest priority.

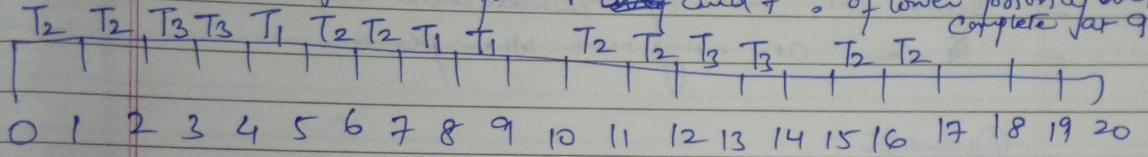
∴ put T_2 at 0, 5, 10, 15 for 2 units each.

↳ capacity

2) T_3 with next highest priority. Put 2 times at intervals 0 ~~&~~ & ~~&~~ 10. Start from 2 & 12 ∵ T_2 with higher priority is already running!

→ capacity

(3) Schedule T_1 which ~~complete~~ runs only ~~fo~~ once from 0 – 20, so put it at 0. but it will start from 4 ~~and~~ and 7 ∵ of lower priority and complete at 9.



PAGE No.
DATE / /

Deadline Monotonic Algorithm:-

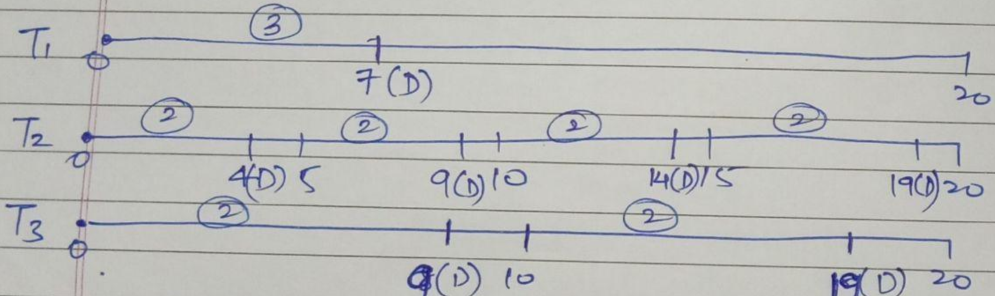
Process	Capacity	Period	Deadline.
T_1	3	20	7
T_2	2	05	4
T_3	2	10	9.

Step I:- Calculate feasibility

Step II: Calculate LCM of Period (20, 05, 10) = 20

Step III: Calculate priority. {Min deadline first}

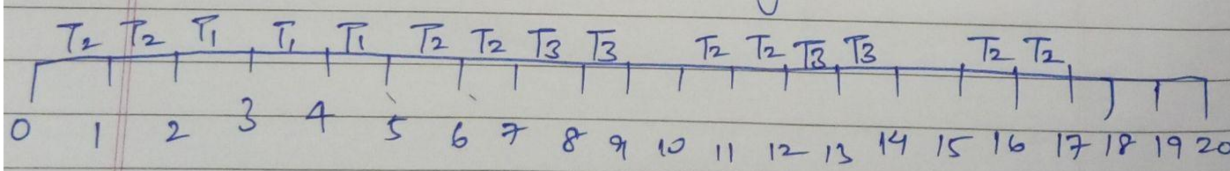
$T_2(4) > T_1(7) > T_3(8)$



① Put T2 on priority at 0, 5, 10, 15

② Put T_1 next priority at 0. but it will start at 2 ∵ of T_2.

③ Put T3 at 0 & 10 but it will start at 7 ∵ T_2 T1 then T3. will get time at 7.



Earliest Deadline first :-

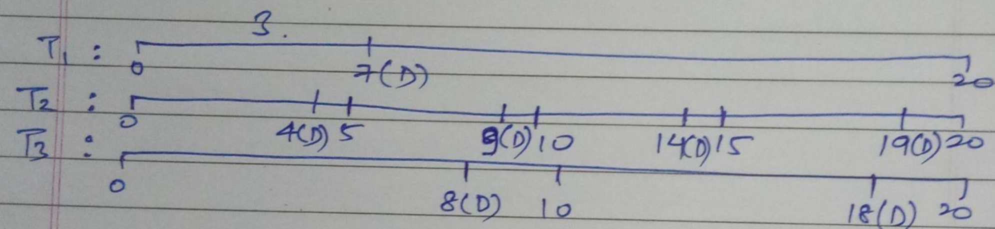
Step I : feasibility ?

	Capacity	Period	Deadline
T_1	3	7	20
T_2	2	4	05
T_3	2	8	10

Step II : LCM (Period) = LCM (20, 05, 10) = 20.

Step-III :- Priority will be considered at run time.

Timeline chart



Deadline mapping for priority calculation.

$4 > 7 > 8 > 9 > 14 > 18 > 19.$

T_2 T_1 T_3 T_2 T_2 T_3 T_2