Real Time Operating System
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 09
RMA Task Schedulability

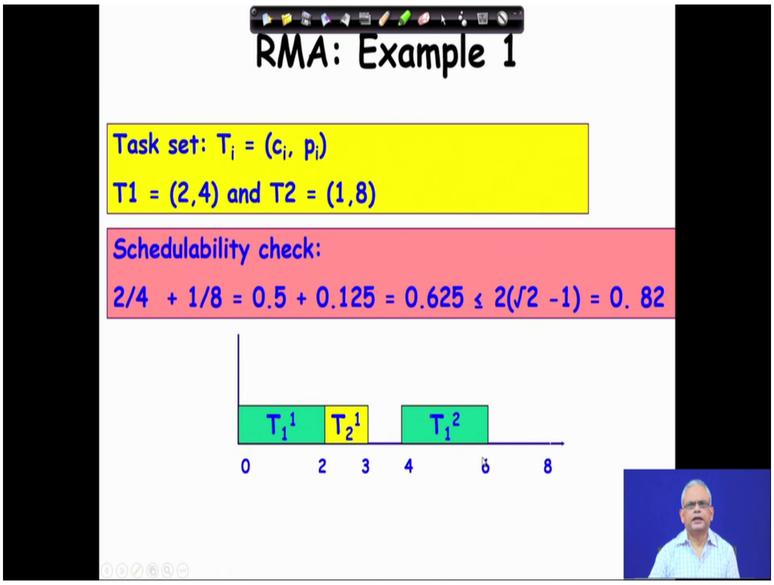
Welcome to this lecture, in the last lecture we had discussed an important category of Real Time Tasks Scheduler; the name of the scheduler is rate monotonic algorithm or RMA. We had seen that this algorithm is simple the designer assigns static priority to the tasks that are there in the tasks set and the algorithm or the technique that uses to assign priority is that the tasks which have higher rate are given as higher priority, those who have shorter period are assigned higher priority and this higher priority tasks they maintain their priority this static priority scheduler.

Once the designer assigns the priority to a task it does not change and then higher priority task always runs if even if there are low priority tasks the lower priority tasks are preempted or they just keep on waiting until the higher priority task completes. One important consideration here is that, given a tasks set how do we check whether the tasks set can be feasibly run by the scheduler and we had said that we check this by using certain utilization bounds.

The first one is the basic one says that the utilization bound of a tasks set cannot be more than 1 to run on a uni processor or sigma e i by P i should be less than equal to 1, but that was a necessary condition not the sufficient condition and then we looked at the sufficient condition given by Liu Laylands expression which is given by n into 2 to the power 1 by n minus 1, where n is the number of tasks to be scheduled that gives the utilization bound. So, as long as the utilization bound for a tasks set having n tasks is less than or equal to n equal to 2 to the power 1 by n minus 1 the tasks set can be feasibly run by the rate monotonic algorithm.

Now, let us look at some examples.

(Refer Slide Time: 03:00)

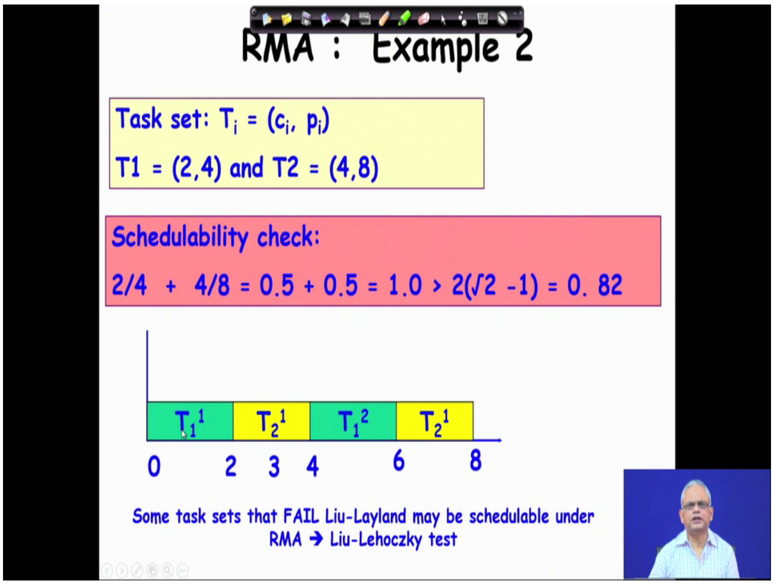


Let us assume that we have 2 tasks and we have given only the execution time E i and P i and we assume that P i is equal to D i the deadline is the same as the period. Now there are 2 tasks execution time 2 period 4 task 2 execution time 1 period 8, can this run on a uni processor under the rate monotonic algorithm. To check that we need to use the Liu Layland criterion; under Liu Layland criterion we first need to compute the utilization due to the 2 tasks.

The first task needs 2 units of execution time every 4 units. So, the utilization is 2 by 4, the second task needs 1 unit of execution time every 8 units. So, the utilization is 1 by 8 and if we simplify we get 0.5 plus 0.125 which is 0.625 and then we compute the Liu Layland bound because there are 2 tasks n into 2 to the power 1 by n minus 1, becomes 2 into 2 to the power 1 by 2 minus 1, which is 0.82 and 0.625 is less than 0.82 and therefore, the tasks set can be run feasibly on a uni processor under the rate monotonic algorithm.

Now, let us just plot out that, how will tasks set run. Let us assume that both of these arrive at time 0 that is they have 0 phasing all of them they get ready at time 0, but then the task T 1 has higher priority than task T 2 because it has a softer period and therefore, T 1 should be first taken up for scheduling. So, the first instance of T 1 runs and it takes 2 units of time to complete and till that time the first instance of T 2 keeps on waiting and at this point the T 2 instance start running and T 2 needs only 1 unit of time. So, it completes by 3 and at 4 the second instance of T 1 arrives and it keeps running up to 6.

(Refer Slide Time: 06:13)



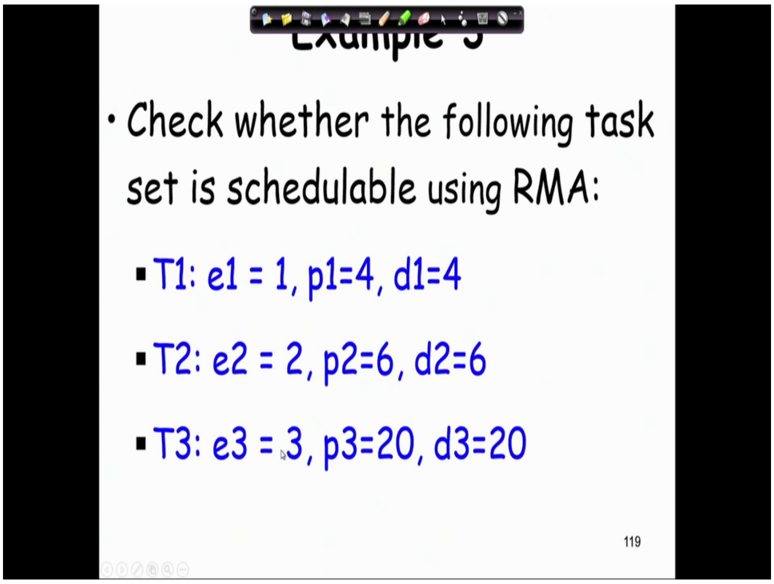
And then the next instance of T 2 arrives at 8 and so on. So, if we plot the tasks their arrival and based on their priority which task will run we can see that both tasks meet their respective deadlines as given by the Liu Layland expression now let us try another example for practise.

So, here it is 2 by 4 the again 2 tasks, the first task takes 2 unit of execution every 4 units, but now we have the task 2 taking 4 units of execution time every 8. So, the utilization due to the first task is 0.5 and even for the second task is 0.5 and therefore, it is 1, but the Liu Layland bound is 0.82. So, according to the Liu Layland expression this tasks set should not be schedulable let us check out whether it is so.

So, first T 1 will run because it has the higher priority run for 2 units and then as it completes T 2 will start it takes 4 units, but then after it run for 2 units the second instance of T 1 will arrive it will preempt T 2 and start running from 4 to 6 and as the instance of T 1 completes at 6. The preempted T 2 will start running, but at 8 it has completed and the new instance of T 1 will arrive, but then just see that both T 1 has met it is deadlines and T 2 also has met it is deadlines. Even if we draw this schedule for a long time period we will see that both T 1 and T 2 meet their deadline, but then the Liu Layland criterion indicated that the tasks set is not feasibly schedulable.

So, we can conclude here that Liu Layland criterion is actually a pessimistic criterion in the sense that, if a tasks set meets the criterion definitely it will schedulable, but in some instances like this example even if the tasks set does not meet the Liu Layland bound still it is getting feasibly scheduled. We need to prove further and the results of this category where probed by Liu and Lehoczky and in 1986 I think 1987 they came up with a new criterion to check whether a tasks set which fails the Liu Layland criterion can still meet it is deadlines, we look at the Liu Lehoczky criterion. So, even a tasks set misses or does not meet the Liu Layland criterion still it may be possible to feasibly schedule it.

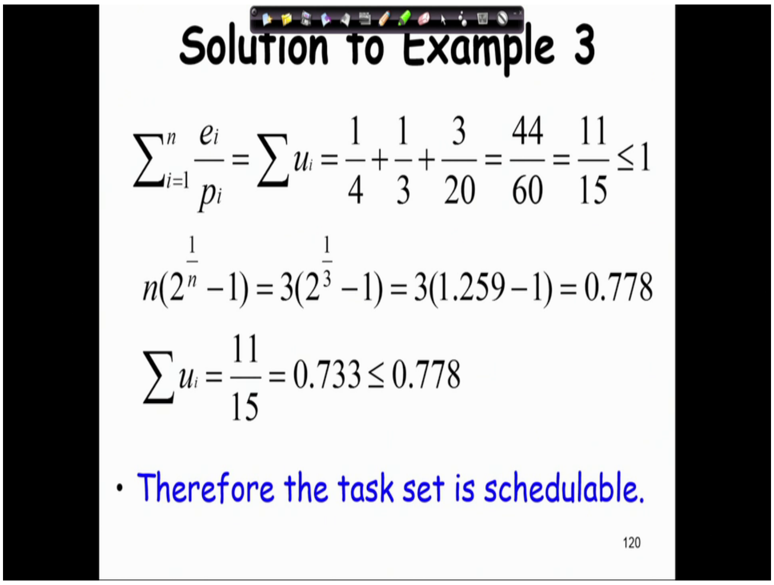
(Refer Slide Time: 10:28)



There is another example here we have 3 tasks T 1 T 2 and T 3 the first task takes 1 unit of execution every 4 units, the second task needs 2 units of execution every 6 unit, third task needs 3 units of execution every 20 units of time can the task set be feasibly scheduled under RMA.

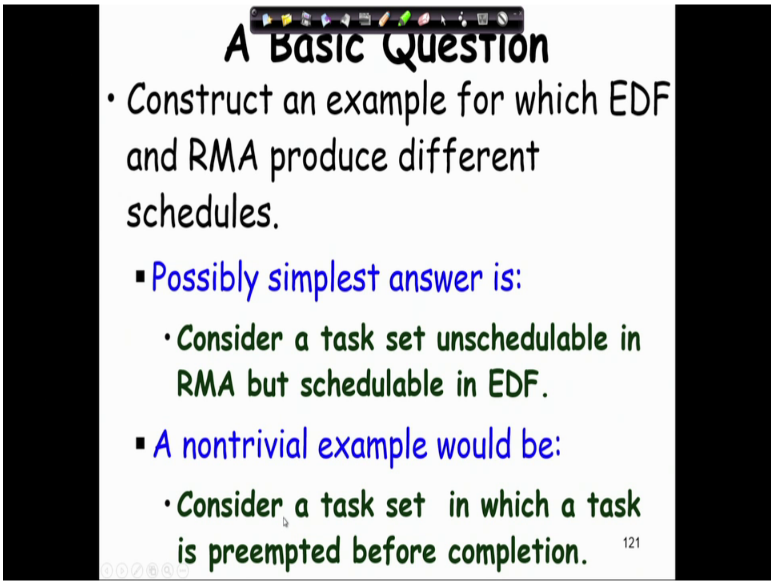
To do this we need first to check the basic criterion whether the utilization due to the 3 tasks sets is less than 1, then we can check the Liu Layland criterion for 3 tasks we need to compute the utilization bound proposed by Liu Layland which is 3 into 2 to the power 1 by 3 minus 1 and then check whether the utilization meets or is under that bound.

(Refer Slide Time: 11:41)



If we compute the utilization due to the 3 tasks we find that it is 40 by 60 44 by 60 or 11 by 15 which is of course, meets the basic criterion that the utilization is less than 1, but let us look at the Liu Layland criterion the Liu Layland criterion is if we you substitute any 3 then 3 into 2 the power 1 by 3 minus 1, which is 0.778 and if we simplify 11 by 15 it becomes 0.733 which is less than 0.778 and therefore, by the Liu Layland criterion we can say that these tasks sets can be feasibly scheduled in the rate monotonic scheduler.

(Refer Slide Time: 12:43)



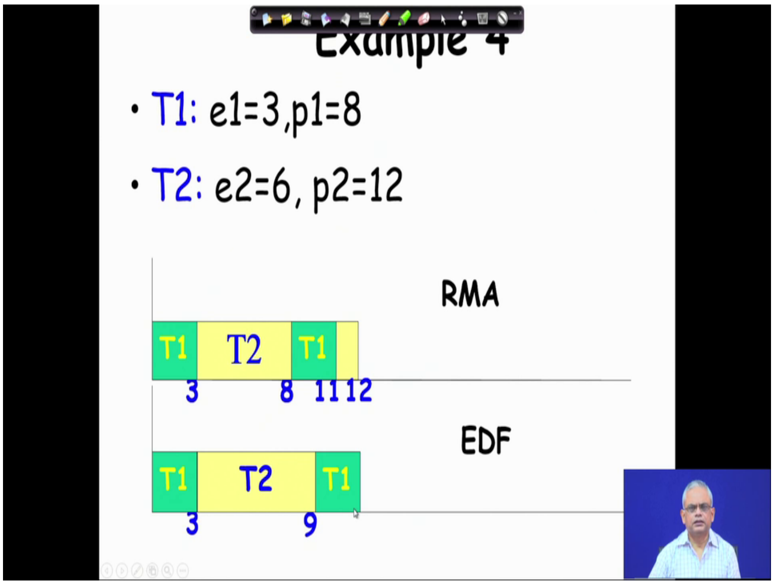
But now is the time to just look back and just think about 1 thing is that EDF and RMA they are actually different types of schedulers.

In the EDF at any time the scheduler examines the task that are ready finds out which has the shorter deadline the shortest deadline and then dispatches it for execution, on the other hand in RMA we assign static priority to tasks based on their rate of arrival or the task period, but then if you plot the task executions for any arbitrary task set may be you can take it as a homework you just take a arbitrary tasks set may be 2 tasks 3 tasks or 4 tasks and check the schedule that is worked out by the EDF and the schedule that will worked out by the RMA and you match them they are actually identical, but is there any example that you can construct where they produce different algorithms sorry different schedules, if you think smartly then you can come up with a simple answer.

So, you are trying to find out a tasks set for which the EDF and the RMA will produce different schedules the simple answer is that for EDF you need the utilization is 100 Percent. It will produce a feasible schedule for RMA unless the schedule is less than the utilization bound it will not be able to produce a feasible schedule. So, can we take some arbitrary task set which is schedulable in EDF, but not schedulable in RMA and then the schedule will be different.

So, we will consider a tasks set unschedulable under RMA , but schedulable in EDF and then of course, we can find the 2 schedulers produce different schedules a non-trivial example can be a task set in which the task is preempted before completion.

(Refer Slide Time: 15:48)

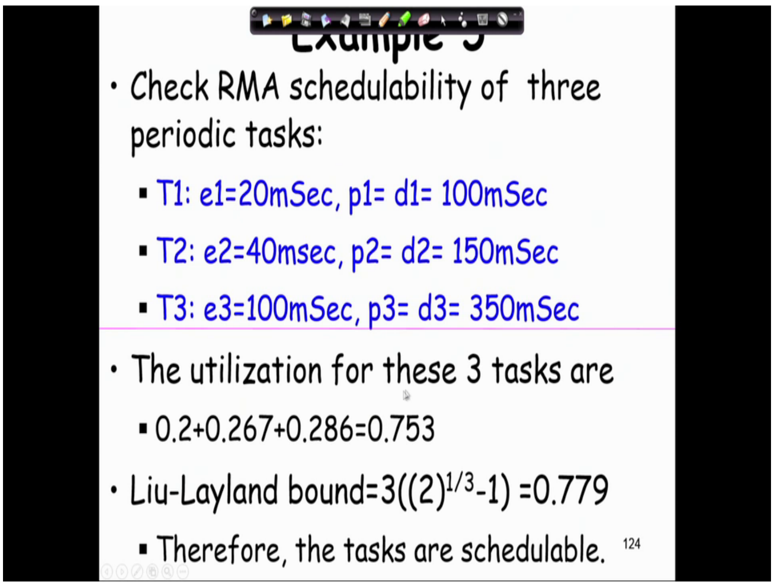


Now, let us look at 1 more example we have 2 tasks T 1 and T 2 execution time is 3 and period 8 and another task 6 and 12, in rate monotonic algorithm the schedule that will be produced is that initially T 1 will run because T has the higher priority than T 2 as it completes at 3 T 2 will start running it will run till 8 and then T 1 will start running, because T 1 next instance will arrive on the other hand in EDF T 1 will run up to 3 T 2 will run up to 9 and then T 1 will run, because at this point when T 1 is running and at 8 the task T 1 arrived at 8, but by that time the deadline for T 1 is sorry T 2 is near already. So, it is deadline is 12 whereas; the deadline for T 1 is 16 because it arrived at 8 and therefore, T 2 will run here till 9 and then only T 1 can run.

(Refer Slide Time: 17:37)

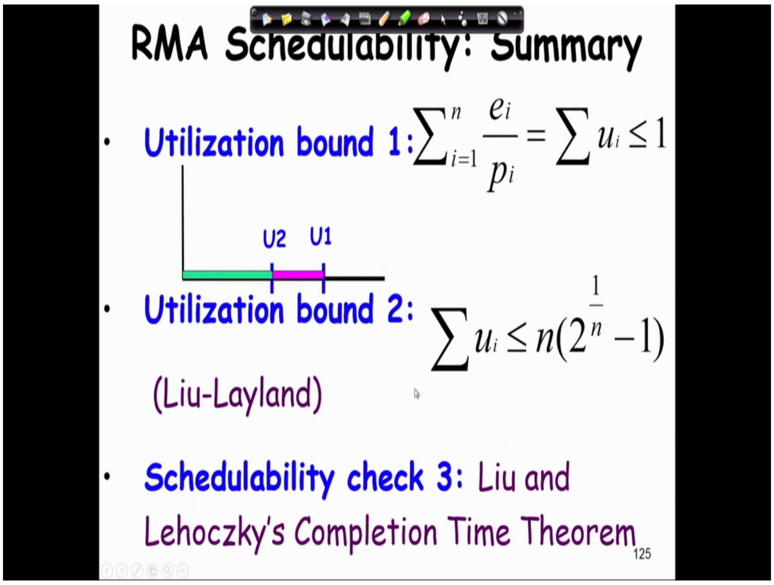
We have already seen through examples that Liu and Layland condition is actually a pessimistic condition as the number of tasks increases it settles at 69 percent and it is if it is a sufficient condition that if a tasks set meets this definitely it will be schedulable, but then even if it fails the Liu Layland condition, then still there is a chance that it may be schedulable. So, we look at another result called as the completion time theorem given by Liu and Lehoczky where we can check if the tasks set will actually meets it is deadline even if it is not schedulable in the Liu Layland condition.

(Refer Slide Time: 18:39)



So, few more examples please work them out on your own say here 3 tasks different execution times and deadlines and we need to check the Liu Layland result. We find that the utilization is 0.75 there and the bound is 0.779 and therefore, the tasks are schedulable please work out your own and check if you also find the same answer.

(Refer Slide Time: 19:14)

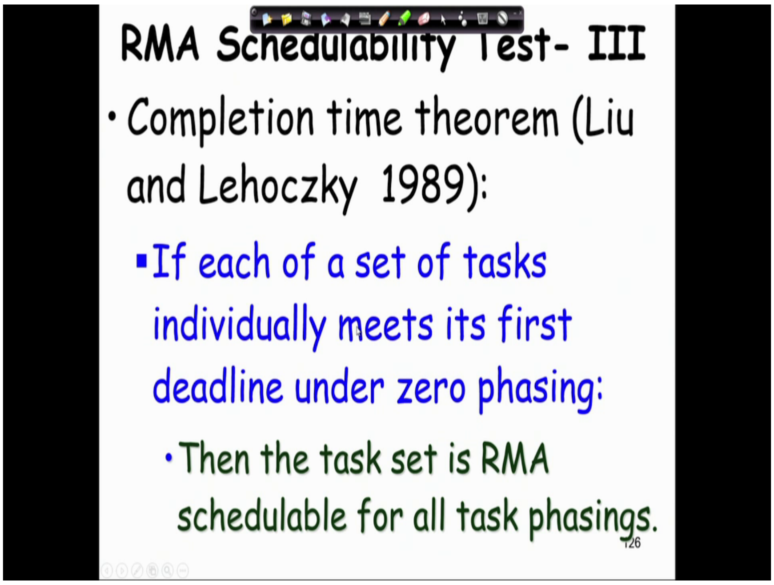


Just to summarise what we have discussed in this lecture. So, far is that we have first looked at the basic algorithm rate monotonic algorithm find found that the programmer needs to assign higher priorities to tasks with lower periods and once he does that the scheduler will run the tasks, but then we need to know whether they will satisfactorily run for that we were discussing some bounds the first is that utilization bound 1 the sum of utilization should be less than 1.

The second bound is the Liu Layland bound is given by n into 2 to the power 1 by n minus 1 the u 2 bound is typically less than e 1 for a given tasks set, but of course, if the tasks set contains only 1 task then u 1 is equal to u 2 and u 2 stabilises at around 70 percent utilization. So, if a tasks set is less the utilization for a tasks set is less than 70 percent it will definitely be schedulable, but we just throw some example found that Liu Layland is actually a pessimistic result. Even if a tasks set is will meet it is deadline, but it may violate the Liu Layland condition and we need to check further that the schedulability check 3 and the name of the result is completion time theorem given by

Liu and Lehoczky in 1987 and we need to check if a tasks set fails Liu Layland , but still it is schedulable.

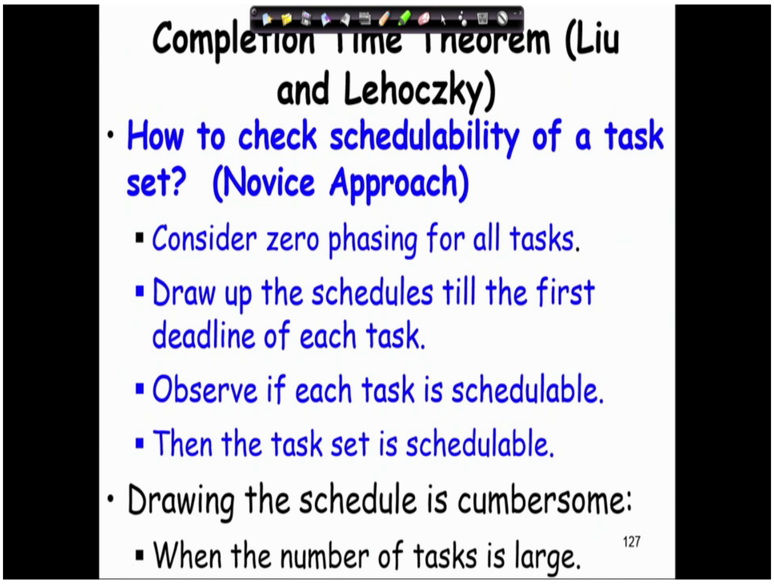
(Refer Slide Time: 21:35)



Now, let us look at the test 3 even if a task set does not meet the Liu Layland condition there is a chance that it may actually be schedulable and we need to check at Liu Lehoczkys condition it is given in 1989 sorry I told 87 , but it is actually 89 Liu Lehoczky published this research in 1989 the main result that they gave is that we look at every task in the tasks set and then we assume 0 phasing among the tasks that is all tasks start at time 0 that is 0 phasing. Even if they have different phasing's we just remove the phasing's and considered they as to be 0 phasing and then under 0 phasing we check whether all tasks will meet their first deadline if they meet their first deadline then the tasks set is schedulable.

So, the result as it is stated is that we will consider for a given tasks set all tasks become ready at time 0 and we just need to check whether they meet their first deadlines. If each of the tasks meet it is first deadline then we can safely say that the tasks set is schedulable, but then why 0 phasing and why check only up to first deadline. The answer to that question is that Liu Lehoczky found that the worst case condition for schedulability occurs when all the tasks arrive in phase and that is the time if all tasks arrive in phase then the lowest priority task will get delayed until all other high priority tasks complete and therefore, that is the condition 0 phasing where some tasks are likely to miss their deadline, but even under 0 phasing if the tasks set is able to meet the respective deadlines even the lowest tasks lowest priority tasks are able to meet their deadlines then the tasks set will be schedulable.

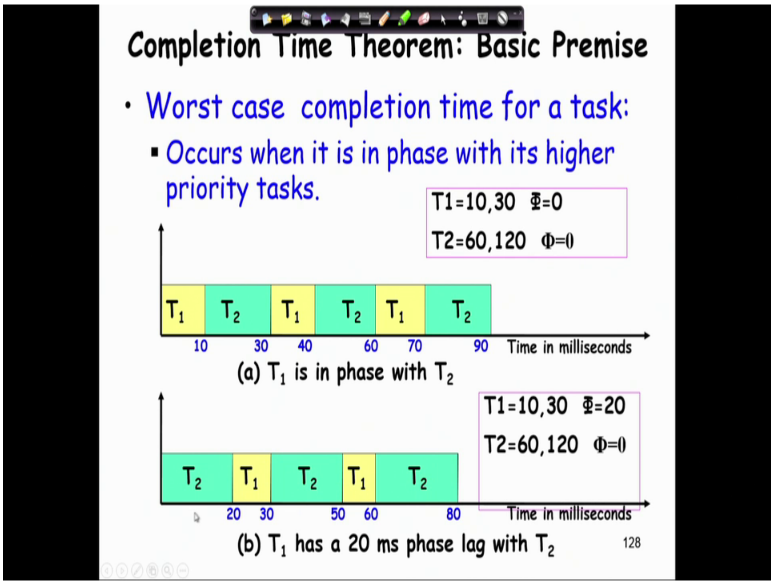
(Refer Slide Time: 24:19)



Now, we know the result given by Liu Lehoczky that consider 0 phasing for the tasks and check if all the tasks meet their first deadline and then we can safely say that they will meet all their deadlines, but how do we check the simplest or the novice approach is that we actually plot out the tasks schedules when the tasks arrive at time 0. So, we consider 0 phasing for the tasks we drop the schedule till the first deadline for each and observe is each task is schedulable and then we conclude the tasks set is schedulable.

But if we have large number of tasks with different periods drawing the schedule is cumbersome prone to errors and this may not be the best approach can we have a mathematical expression where we substitute values and then we get the results whether it is schedulable or not.

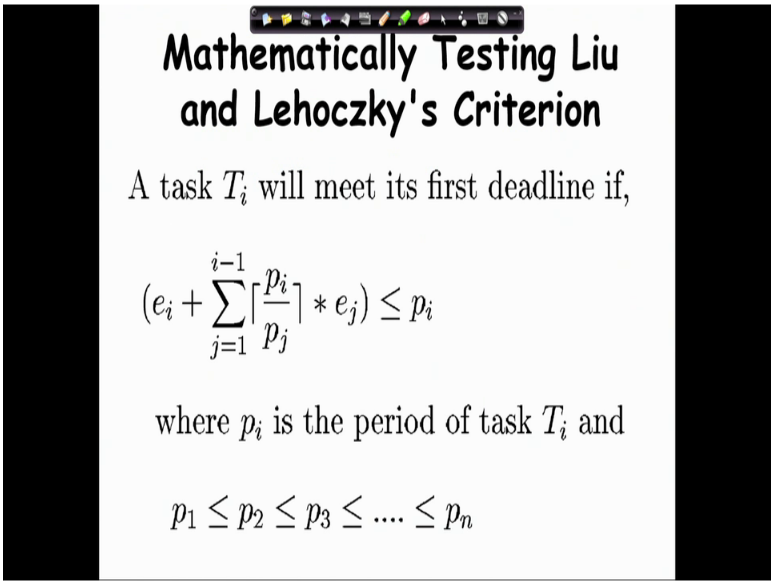
(Refer Slide Time: 25:36)



We will just do that in the next slide we have already said that the basic assumption in this completion time theorem due to Liu Lehoczky is that the worst case phasing occurs when a task arrives with it is higher priority. We can check that out here the T 1 and T 2 are 2 tasks and T ones period is 30 T 2 period is 120 and let us assume that they arrive at 0 phasing then for T 2 to complete execution, we can draw the schedule here and see that T 2 needs ninety units of time to complete because whenever T 1 becomes ready during 0 during 30 during 60 it will run for 10 units of time the rest of the time T 2 can run and we can see that it will complete at 90, but now let us assume that there is a phasing T 1 arrives only after 20 and T 2 arrives at 0. So, again we plot since T 1 phasing's is 20 T 1 arrives at 20 displaces T 2 T 2 runs from 0 to 20 gets displaced and so on we find that T 2 completes by 80.

So, this indicates this example is indicating of the fact that the worst case completion time for a task occurs when it arrives in phase with it is higher priority task. So, that possibly prompted Liu and Lehoczkyto give theorem that we need to check for every task whether it meets it is first deadline when all the task arrive in phase.

(Refer Slide Time: 27:55)



But how do we mathematically test Liu and Lehoczky criterion the mathematical expression is given as e i plus sigma j equal to 1 to i minus 1 ceiling of p i by p j into e j is less than p i the intuitive idea behind this expression is that e i is the time the task e i needs to complete, but then whenever it is higher priority tasks are ready they have to run the task T i has to wait and the higher priority tasks are the tasks 1 to i minus 1 and how many times will they occur before, the e i is period the number of times they will occur is given by p i by p j ceiling you can work it out you can just check couple of examples the number of times a higher priority arrives before the e ones period is given by p i by p j.

For example, p j is 5 and p i is 3. So, p i can occur 2 times ones at 0 and another is at 3 before p j which will 5 and each time they will take e j amount of time and therefore, e i sigma j equal to 1 to i minus 1 p i by p j ceiling into e j we need to check whether it is less than equal to p i and that will give us the schedulability of that task. Now we need to check for every task whether they meet this expression. So, today we will just stop here and in the next lecture we will take up few examples and exercises on the Liu and Lehoczkys completion time theorem and then we will proceed further.

Thank you.