L-4.8: GATE 2018 Question Explaination on deadlock | Operating system

Short notes or summery of the whole notes

The question provides a system with three processes, P1, P2, and P3, that share four instances of the same resource type. Each process can request a maximum of K instances, and the goal is to find the largest value of K that will always avoid deadlock.

To solve the problem, the solution starts by considering the case where each process needs only one instance of the resource. In this case, the total demand is three, which is less than the total number of resources available. Therefore, all the processes can be granted the requested resources, and deadlock is avoided. Thus, the value of K can be 1.

However, the question asks for the largest value of K that will always avoid deadlock, so the solution considers other cases as well. For example, if each process needs two instances of the resource, deadlock can still be avoided by allocating one instance to each process and one to any of the processes. In this case, the value of K is 2.

Yes, you have derived the formula for determining the minimum number of resources required to avoid deadlock in a system. The formula is R + n > summation of i=1 to n (di - 1), where R is the total number of resources in the system, n is the number of processes, and di is the demand of process i. This formula ensures that there are enough resources available to satisfy the maximum demand of each process, while also leaving enough resources free to avoid deadlock.

To explain this formula, we can say that the left-hand side of the equation represents the total number of resources available in the system (R) plus the number of processes (n), which means the total number of resources that could potentially be used by all the processes. The right-hand side of the equation represents the total demand of all the processes, but with each process's demand reduced by one. By subtracting one from each process's demand, we ensure that at least one resource is always available for other processes to use, which prevents deadlock.

So, if the inequality R + n > summation of i=1 to n (di - 1) holds true, then the system has enough resources to avoid deadlock. If the inequality is not true, then the system may experience deadlock. Therefore, to ensure a deadlock-free system, you must ensure that the formula is satisfied by providing enough resources to the system.

Detailed explanation of the concept

The question is on Deadlock. Consider a system with 3 processes that shares 4 instances of the same resource type. Each process can request a maximum of K instances. The largest value of K that will always avoid the deadlock is.

Means we have given 3 processes, P1, P2, P3.

3 processes are given. And they're saying if they share 4 instances of same resource type, means there is some resource R, and we are having 4 instances means we are having 4 units of R. We are having the 4 units of R. Now they're saying these 3 processes can request a resource K unit. Means K instances, a process can request for K instances you have to find the value of K and the value of K should be maximum.

Means, P1 needs only 1 unit of R to get executed. P2 also needs only 1 R, P3 also needs only 1. So, 3 resources I have provided. I am having 4 in my hand, but out of 4, I have given 1, 1, 1 resource and all the processes will get successfully execute because they said they need 1, I have provided 1 easily because I am having 4 and their total demand is 3 only, so I can easily provide.

So, the answer can be 1. The K value can be 1. But this is one of the answers. The value of K we have to find maximum. So, whether it is maximum or not, let's check. If K value is 1 means the demand of each process is 1, I can easily give 1, 1, 1 resource, but still there will be what? It is not the maximum value. K value is not the maximum.

So, let's increase the value of K. Let's make the value as 2. Let's make the value as 2. Now what is the meaning of this? Every process says their demand is 2. They need 2 resources; we have 4. So, what can we do? One is we can give 1, 1, 1 resource to every process and the fourth one we give to any of the like, P1, P2 or P3.

The thing is, whenever there is a question like this. Whenever the question is asked like this, what you need to do is, first write down all the processes. Try to give the minimum need means we are having 4 resources in our hand. What I did, I have given 1, 1, 1, to all the processes and the fourth one, I can give any of the process let's say, the fourth one I have given to P1.

So how many processes does P1 get? 2. P1 gets 2 and what is the need of P1? 2. So, when the need is fulfilled, the P1 will get executed and when P1 will get executed, the 2 resources will be free. Whatever the 2 resources it is having that will be free and that 2 resources I will provide to P2 and P3. So P2 need is also 2, fulfilled, P3 need is 2, fulfilled.

So yes, K value can be 2 also. So, K value can be 1 also, K value can be 2, but as the question is asking for the largest value of K, so we cannot say that 1 is the appropriate answer because we find 2 also appropriate. So now 2 is the answer. Let's check for 3. Let's say, the demand is 3.

Means every process needs at least 3 resources or you can say 3 resources to get executed and how many processes given? 3. 3 processes are given. So, if it says that I need 3 resources, what can we do? means out of 4 resources I provided 3 to P1 and the fourth one which is left, I provide to P2. What I did? I have 4, each process needs 3. So out of 4, I picked 3.

And I give these 3 to the P1 and the fourth one I give to the P2. Now you can easily say P1 needs 3 so P1 get 3. So when P1 gets 3, it will be successfully executed and when it gets successfully executed, the 3 resources will be free. So when the 3 resources will be free, I will give it to the other processes.

Like 3 resources I am having free, so let's say I give 2 to them, 2 to P2 and 1 to P3 and now P2 also gets all the 3 resources, it needs 3 I gave it 3. So it will also get successfully executed and removed. And whatever the resources were occupied by P2 they will be free. I will give it to P3. Then P3 will also successfully execute.

So can we say the value of K can be 3? No, there must be you have to follow all the cases because there is for 'always' means will 'always' avoid the deadlock situation. So whenever the question is asked on the always, you have to analyse all the cases. Now let's say the demand is still 3.

What we can do? Let's say there can be one of the case 1, 1, 1. Out of 4, 1 resource I gave to P1, 1 resource I gave to P2, 1 resource I gave to P3. The fourth one. Let's say I gave it to the P1. Now see P1 needs 1 more resource to get executed, P2 needs 2 more, P3 needs 2 more. How P1 will get 1 more? No, there is no chance that P1 gets 1 more.

There is no chance that P2 gets 2 more, there is no chance that P3 gets 2 more. So this location will be empty because P1 needs 1 more resource as we have the request is three so P1 needs 1 more, P2 needs 2 more, P3 needs 2 more. Who will fulfil? No, nobody is going to fulfil the request. So in this situation there will be a deadlock because P1 is waiting for 1 resource, P2 is waiting for 2 resources, P3 waiting for 2 resources.

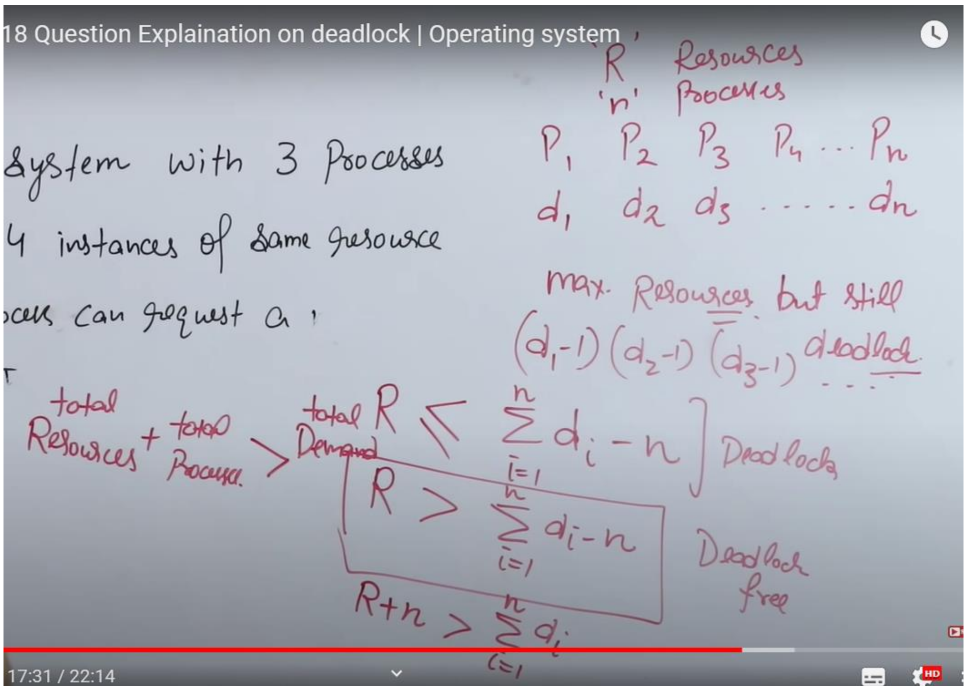
And who will be providing them with the resources? Nobody, because P1 has taken 2, P2 has taken 1, P3 has taken 1. So, what is the meaning of this? Why have I written this? We have to examine all the cases, don't directly say that K value 3 is the answer. For one or two cases, K value 3 can be correct.

But you have to examine all the cases, and this is one of the cases, if I take the number of resources as 4 and demand as 3 then there will be a deadlock. So, you can say 3 values cannot be the right answer. So, what is the value next? 2, because 3 is not the right answer. How can we take the demand as 4, 5, or 6? We cannot increase the demand because when the demand is 3 only, still there is the possibility of deadlock in one of the cases. So, we cannot increase the value of K from 3 to 4, 5, 6.

So, what is an appropriate answer? The value will be 2. The answer should be 2 only. So, if I take the demand of each process as 2, then there is not any case in which a deadlock can occur.

No, neither. If I take the value of demand as 2, all the cases will be deadlock free. We can say all the cases will be free from the deadlock. So, the answer should be 2.

Let's find some other way. This is the manual; this is the manual method. You can follow this method also, but there is another method, another method to find. Let's derive the formula. Let's say I'm having R number of resources. The total number of resources I have R.



In my system, total resources R

let's say I'm having 'n' number of processes.

I have written like this P1, P2, P3 up to Pn and the demand,

let's say the demand of P1 is D1,

Demand of P2 is D2, and up to Dn.

So, what we have assumed, let's say R number of resources, n number of processes and D1, D2, D3 is the demand.

So, let's derive some formula which is related to all these three. Now, if I want the system to be deadlock free, one thing is, I can provide the D1 resource, P1 needs D1 resource, D1 number of resources. I will provide them. P2 needs D2, I will provide, P3 needs D3, I will provide, means whatever their need, if I provide that the need will be fulfilled, there will be no deadlock.

But here we have to find the minimum means it should not be like What is the maximum demand? I will fulfil directly. We have to find the minimum number of resources means the resources should be less and the deadlock should be removed. It should not be like if there are 3 number of resources or let's say 3 number of processes and 2 number of, 2 is the demand. So, the total will be 6 and I will give the 6 resources. If I give the 6 resources, that is the normal answer.

Still deadlock means I'm trying to provide the resources to the processes according to their demand. But still there is a deadlock. So, this is quite tricky. Let's say I'm having P1, P2, P3. Every process says their demand is 2 means it says 2, it says 2, it says 2. If I provide 2 to him, 2 to him, 2 to him, means total if I have 6 resources, then there is no problem, deadlock is free because there is no deadlock. P1 needs 2, I provide 2, 2, I provide 2, 2, I provide 2.

So total how many resources must I have? 6. But no, this is not the appropriate answer. You have to minimize the value. You should not purchase the maximum resources. You should purchase a minimum, but free from the deadlock. So how do we derive it? What can we do? rather than directly giving the demand. Their demand is 2, I'm not giving 2, I'm giving 1, I'm giving 1, I'm giving 1.

Means how many resources I'm giving? 3. But now there is a deadlock because everybody needs 1 of the resources. Everybody needs 1 resource. Every process needs 1 resource. What will I do? I will add 1 more and it will be free from the deadlock means if I had 1 more, if I gave like this. Means if I give 2 to P1, it will be executed. The resources will be free.

It will be given to P2 and P3, they will also execute. So, what we are trying to find is, we're trying to find the maximum number of resources we have, but still there is a deadlock. How to find it? Same, the demand is D1, I'm not giving the demand D1, I'm giving D1 minus 1. One less than the demand. Same for this, D2 minus 1, same for that D3 minus 1, same for the last.

Now, what is the total number of resources I have? Total number of resources is whatever the total number of resources, like D summation, just add all the resources, summation of Di where i=1 to n. Di minus 1, minus 1, minus 1, minus 1, it will be up to end. So, this is the formula we can derive. So, what is this? The demand, the total demand minus n, means this number of resources, if I have the resources, what is the number of resources I have? If I have the resources R and number of resources R is less than equal to this.

If the number of resources R is less than equal to this, there will be a deadlock. There will be a deadlock because whatever the demand, I'm not fulfilling that demand directly. I'm giving 1 less than, I'm giving 1 less than, I'm giving 1 less than. So, there is still a deadlock existing.

If I provide these number of R, there will be a deadlock. So, R should be greater than this. If I want to free from the deadlock means in this case, there is still a deadlock. But if I do like R should be greater than summation of i=1 to n, di-n, in this case, it will be deadlock free. means the number of resources I have provided that is more than the summation of all the requirement minus n.

That is the number of processes. So, we can say R+n should be greater than summation of i=1 to n, di. This is what, R+n is greater than this. What is the meaning of R+n? R+n is, R is what? The number of resources. Total resources plus, total resources that is R, n is the total processes. Total number of resources plus total number of processes should be greater than the total demand.

This is the meaning of this. So, the total number of resources plus the total number of processes in the system. It should be greater than the total demand; total demand means demand of all the processes. If I provide this, then there will be no deadlock. In this situation there will be no deadlock. So, if the question is asked like this, you can solve manually also as given in the previous way.

But you can follow this method. What is the method? What is the formula? Total number of resources plus total number of processes should always be greater than the total demand. So, let's say what is the total number of resources in this question? 4. What is the total number of processes in this question? 3.

It should be greater than the total demand. What is the demand? The demand should be, let's say if I say every process demand is, like 3 processes are there, and every process demands for 2. So, what is this? 7 should be greater than 3*2 that is 6. It is true. So yes, this is a valid answer. So, every process can demand a maximum of 2.

If you increase the demand to 3, let's say if you increase the demand to 3. So what is the total number of resources? 4. Number of processes 3. It should be greater than the total demand. Total demand means the number of processes is 3 and demand of every process is 3. So what it is? 7 should be greater than 3*3 that is 9. It is false. So the demand of each process cannot be 3, even if cannot be 3, you can also say 4, 5, 6 is not possible, but yes, if the demand of every process is 2, so you can simply say yes, this is valid.

So always remember this formula. Resources + processes, Total number of resources plus total number of processes should always be greater than the total demand. This is the important formula. So we have seen also how to derive this formula and you can, you don't need to derive it again and again in the exam, you just remember this formula and according to the question, you can apply it. So according to this question now we can apply and the answer is 2.

I'm having 4 number of processes and I'm having 4 number of resources. So what can be the maximum demand? The maximum demand can be like now what is the number of processes? 4, number of resources, 4. 8 should be greater than total demand.

So each process if demand for 1, each process if demand for 1, then 8 is greater than 4. This is valid. Try to increase the demand. 4 + 4 is greater than, each process demand for 2. So what is 8 greater than 8? This is false. So this is a very important formula. If I am having 4 processes, 4 resources and each process demand for 2 to get execute, this is false. There will be a deadlock. But if every process demand for only 1, there will be no deadlock.

You can do it like that also. P1, P2, P3 P4, 4 processes, each process demand for let's say 2, so 1, 1, 1, 1. 4 I have given. 2, How can I give 2? Everybody is waiting for 1 resource and I have already allocated 4 number of resources, what I have, I have provided. So there will be a deadlock. So demand of each process cannot be 2. They cannot demand for 2 process, 2 resource.

They can demand for 1, no problem. If they demand for 1, I can easily give 1, 1, 1 to every process and they will get successfully executed. So this is how we can solve the questions like this manually also, and by using this formula also.