Hello and welcome to yet another module of program TYBSc.

Subject Electronics Semester 5 Paper Code ELC

105 operating systems.

Unit three: concurrency,mutual exclusion and

synchronization. Module name principles of concurrency, Part

2. Model number 13 presented byMr. Girish Abhyankar, Associate

Professor, Dnyanprassarak Mandal's college and Research Center, Assagao Goa..

Outline operating system concerns due to presence of

concurrency.

Process interactions and requirements for mutual

exclusion.

Learning outcomes the student will be able to explain OS

concerns that arise due to presence of concurrency.

Explain the process of interactions based on the degree

of awareness. And explain the requirements for mutual

exclusions.

So dear students, last module we have studied the principles of

concurrency. Let us carry forward the discussion on that.

What are the concerns of the operating system due to

presence of concurrency?

The OS must be able to keeptrack of various processes.

And previously in the previous modules, we've seen that this is

done using process control

block. PCB's.

The second concern: the OS must allocate and deallocate

various resources for each active process.

At times, multiple process want access to the same resource.

The OS must protect the data and physical resources of each

process against unintended interference by other processes.

The functioning of process and the output it produces must be

independent of the speed at which its execution is carried

out related to the speed of the other concurrent processes.

So what is the speed of the other process should not vary

the output of this process. The Knowledge of the ways in which

processes indirect helps to understand how the issue of

speed independence can be

addressed. So there are three categories we can classify based

on the degree of awareness of process, how processes know each

other number, one, the first category process is unaware of

each other. These are independent processes that are

not intended to work together.

The best example is the multiprogramming of multiple

independent processes. Although The processes are not working together, the OS needs to be concerned about competition for resources among these processes.

For example, two independent applications may both want to access the same file or same printer simultaneously. No,there will be a conflict between these two processes. So OS has to regulate or resolve this conflict. Another category processes indirectly aware of each other. These are the processes that are not necessarily aware of existence of each other by their respective IDs. They don't know their respective IDs others, but they share access to some object such as input, output buffer, such process.They exhibit cooperation by sharing. There is no competition. Now they're sharing, so they are cooperating in sharing the common objects. A third categories processes directly aware of each other, so each process knows the other concurrent process by their Ds.

OK, so here there is a communication, so these processes are able to communicate with each other by process ideas and that are designed to work jointly on some

activity. Such a process exhibit cooperation by communication. So we have cooperation in sharing or cooperation by communication and in the first category we had competition among the processes..

So concurrent processes may come into conflict with each other when they are competing for the use of same resource.

Two or more processes need to access a resource during their execution. It will happen sometime or the other. Each process is unaware of the existence of the other. We are looking at a competition. So first category process is unaware of each other and each is to be unaffected by the execution of the other processor. This means that each process should leave the state of any resource it uses, unaffected.

There is no exchange of information between the competing processes OK. However,execution of 1 process may affect the behavior of the competing process. Suppose Resource is given to one process.Then the second process who wants the access to that resource have to wait.

Competition among processes forus is continuing that if two processes which to access a single resource than one process will be allocated that resource by OS and the other will have to

wait. Therefore, the process that is denied access will be

slowed down. In an extreme case,it may. It may so happen that

the block process may never get access to the result and hence

will never terminates successfully. Something like

starvation may also result in this. In the case of competing

processes, 3 control problems must be faced. Number one, the

need for mutual exclusion.

And if we enforce this need for mutual exclusion., it results

into deadlock and starvation.

Mutual exclusion suppose two or more processes require access to

a single shareable resource.Very important single shareable.

If there are two printers and two processes want to print

similarly yes you can do it, but if there is only one printer and

two processes want to print at the same time, you can imagine

what will be printed on that paper, so it should prevent

this. During execution, each process will be sending commands

to the IO device, receiving status information, sending data

and or receiving of code is receiving data will not be in

the case of printer, such a resource is a critical resource

and the portion of that program that uses is a critical section

of the program. It is important that only one program at a time

be allowed in this critical section. In case of a printer,

any individual process should have the control of the printer,

but it prints an entire file.Otherwise, as we have discussed,

the lines from computing

resources will be. Interleaved.

If you force mutual exclusion,two more problems creep up.

One is deadlock, one is starvation. Let's look at

them. What is deadlock? Seehere. Suppose there are two

processes P1 and P2, and there are two resources are

abundant.

Imagine that OS assigns the resource R1 to P1 and

resource R2 to P2.

Process P1 requires R2 to complete a certain operation and

process P2 requests R1 we are looking at a specific

sequence which might occur.

Neither P1 is ready to release R1, neither P2 is ready to release R2, so

both are waiting for the other to release. Nobody is wishing to

release the resource which is its control,

and then it results in a deadlock situation. So this is

what is called as a deadlock.OK, so neither of the process

release the resource it already owns until it acquires the

resource from and are unable to

complete. OK, starvation, look at this situation. We have three

processes, P1 ,P2, P3, and they're competing for a resource R.

Suppose P1 is already in possession of R. OS.

Granted permission to P1 ,now. P2 and P3 are delayed

waiting for R. Let us assume that P1 exits the critical

section, either P2 or P3 will be selected by OS.

Let us say, OS, grants

access to P3. And then before P3 completes, P1 again requires the

same resource R and it may so happen that operating system

goes on granting permission to P1 and P3 in the interleaved

fashion without even looking at P2. Example OK, in that case

P2 will indefinitely be denied access to the resource even

though there is no deadlock

situation. The process that interact with other cooperation

among the processes. Now the process that interact with

processes without being explicitly aware of them exhibit

cooperation by sharing, for example, multiple process may

have access to a shared variable on the shared files or database.

The shared data may be used and updated by a process without

reference to the other process,but being aware that the other

process may have access to the same data. Thus, the processes must cooperate to ensure that the data they share are properly managed. Then the integrity of the shared data must be ensured. The control problems of mutual exclusion, deadlock starvation are also present in, as in the case of competition by processes OK, the data items may be accessed in two modes, reading and writing.Reading is concerned there is no problem but with the writing comes there is a problem. OK, that should be mutually exclusion. Let us consider the problem of data coherence. A simple example. Suppose two items of data A&B are to be maintaining. Relationship a is equal to B. That is, any program that update one value must also update the other. Let us say,for example, suppose say a equal to a plus 5 b is equal to b plus 5 program one or process one. Process two is 5 into B is equal to be an A is equal to a into 5 initially. Let us say a is equal to b is equal to 3. Then each process taken separately will result in A equal . To b is equal to 8 because 3 + 5will be eight. . Process two will have 3 into 5 15, so initially they both were A and B 8. Now A&B are 15.

Now consider the following.

Concurrent execution sequence in which two processes respect

mutual exclusion and this is the way this is the order in which.

Sequences are executed and let us assume a is equal to b is

equal to 2. At the end of this execution, sequence a will be

35. Because a here will be 2 + 5 seven and here 7 into

5 35 whereas B will be 2 into5 10 and 10 + 5 15. So there is no

coherence among the data if

This Sequence is executed OK. The problem can be

avoided by declaring the entire sequence in each

process to be critical section.

Cooperation among processes by

communication. When processes cooperate by communication, the

various processes participate in the common effort that links all

the processes. The various activities are synchronized,

coordinated through communication and what type of

communication it could be inform of sending and receiving

message primitives which may be provided in the part of

programming language or provided by the OS kernel. Since there is

no sharing between the processes in the act of passing messages,

mutual exclusion is not a control problem. However, the

problem of deadlock and starvation are still present.

So let us revise number one processes unaware of each other relationship is competition for resources and timing of process may be affected. Output of 1 process independent of the actions of the other process.What problems potential problem may creep up mutual exclusion,deadlock and starvation deadlock because of renewable resources.It's the resource which is not depleted by its use file.Suppose a variable, a memory location. It's not depleted,keyboard is not depleted by its use OK. Second category processes indirectly over of each other. There is a cooperation by sharing. But Maybe influence of 1 process another in this category. Timing Of process may be affected and output or one presence may depend on the information obtained from the other process because they are sharing again mutual exclusion deadlock with respect to renewable resources, starvation and data coherence.We explained just now that also is a problem creeping up here.The last category processes directly aware of each other.Cooperation by communication, timing or process may be affected as normal output of 1 process may depend on the information obtained from the

other process. And deadlock consumable resources consumable means like message. If oneprocess passes a message for the other than the one who passesthe message is the producer. One who takes the message. The Consumer the messages over. So it is a consumable resource and starvation may result last slide. What are the requirements for imutual exclusion total 6 requirements will see at this in details later on, but let us revise them. OK, let us go through them once only one process is allowed at a time in the critical section for Resource. A process that.

Halts in its noncritical section must do so without interfering with other processes.

3rd Requirement, no deadlock or starvation, so process requiring access to critical ascension must not be delayed indefinitely. It must be given

access. Next process must not be delayed access to critical section for a resource when there is no other process using it. So if it is idle it should be allowing immediately. No assumptions are made about relative process speeds or number of processes. And a process remains inside critical section for a finite time only. OK? So these are the

six requirements for mutual exclusion the matter.

Done with reference to these textbooks. Thank you.