DHA SUFFA UNIVERSITY
Department of Computer Science
CS-2004L
Operating Systems
Fall 2020

LAB 11 – Banker's Algorithm for Deadlock Avoidance

OBJECTIVE(S)

- Learn about Deadlock
- Learn about Banker's Algorithm

What is Deadlock?

A deadlock is a situation in which computer programs sharing the same resource are effectively preventing each other from accessing the resource, resulting in both programs ceasing to function.

When a new process enters a system, it must declare the maximum number of instances of each resource type it needs. This number may exceed the total number of resources remaining in the system. When the user requests a set of resources, the system must determine whether the allocation of each resource will leave the system in a safe state. If it will, the resources are allocated; otherwise the process must wait until some other process releases the resources.

Condition for Deadlock

Mutual exclusion

- There must exist at least one resource in the system which can be used by only one process at a time.
- If there exists no such resource, then deadlock will never occur.
- Printer is an example of a resource that can be used by only one process at a time.

Hold and wait

- There must exist a process which holds some resource and waits for another resource held by some other process.

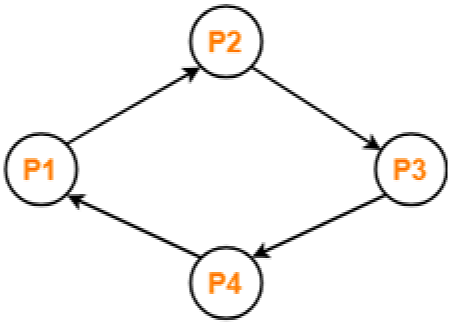
No preemption

- Once the resource has been allocated to the process, it can't be preempted.

- It means resources can’t be snatched forcefully from one process and be given to another process.
- The process must release the resource voluntarily by itself.

Circular wait

- All the processes must wait for the resource in a cyclic manner where the last process waits for the resource held by the first process.



Safe state

Safe state is one which

- Is not a deadlocked state
- There is some sequence by which all requests can be satisfied.

To avoid deadlocks, we try to make only those transitions that will take you from one safe state to another. We avoid transitions to unsafe states.

Banker’s Algorithm

The Banker algorithm, sometimes referred to as the detection algorithm, is a resource allocation and deadlock avoidance algorithm developed by Edsger Dijkstra that tests for safety by simulating the allocation of predetermined maximum possible amounts of all resources, and then makes an "s-state" check to test for possible deadlock conditions for all other pending activities, before deciding whether allocation should be allowed to continue.

Deadlock Avoidance by Banker’s Algorithm:

Avoid actions that may lead to a deadlock. Think of it as a state machine moving from one state to another as each instruction is executed.

Algorithm

1. Start the program.
2. Get the values of resources and processes.
3. Get the available resources values.
4. After allocation, find the need value.
5. Check whether it’s possible to allocate.
6. If it is possible then the system is in safe state.
7. Else system is not in a safe state.
8. If the new request comes then check that the system is in a safe state.
9. Or not if we allow the request.
10. Stop the program.

Example

5 processes P1 through P5

3 resources type A (10 units), type B (5 units) and type C (7 units).

DHA SUFFA UNIVERSITY
Department of Computer Science
CS-2004L
Operating Systems
Fall 2020

Snapshots at time T0:

No. of Processes	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P1	0	1	0	7	5	3	3	3	2
P2	2	0	0	3	2	2			
P3	3	0	2	9	0	2			
P4	2	1	1	2	2	2			
P5	0	0	2	4	3	3			

Need = Max – Allocation

Processes	Need		
	A	B	C
P1	7	4	3
P2	1	2	2
P3	6	0	0
P4	0	1	1
P5	4	3	1

The system is in a safe state since the sequence < P2, P4, P5, P1, P3> satisfies safety criteria.

No. of Processes	Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C
P1	0	1	0	7	4	3	3	3	2
P2	2	0	0	1	2	2	5	3	2
P3	3	0	2	6	0	0	7	4	3
P4	2	1	1	0	1	1	7	4	5
P5	0	0	2	4	3	1	7	5	5

Advantage of Banker’s algorithm

Avoids deadlock and it is less restrictive than deadlock prevention.

Disadvantage of Banker’s algorithm

Only works with a fixed number of resources and processes.

DHA SUFFA UNIVERSITY
Department of Computer Science
CS-2004L
Operating Systems
Fall 2020

ASSIGNMENT # 11

1. Write a shell script or a C++/ Java program to implement Banker's Algorithm for Deadlock avoidance.

SUBMISSION GUIDELINES

- Take a screenshot of each task (code and output).
- Place all the screenshots in a single word file labeled with Roll No and Lab No. e.g. **'cs172xxx_Lab01'.**
- Convert the file into PDF.
- Make a folder labeled with Roll No, place the pdf and .cpp/.java/.sh files in this folder
- Submit the folder at LMS