Department of Computer Science and Engineering

FACULTY OF ENGINEERING AND TECHNOLOGY UNIVERSITY OF LUCKNOW LUCKNOW



Dr. Zeeshan Ali Siddiqui Assistant Professor Deptt. of C.S.E.

METHODS FOR HANDLING DEADLOCKS (Part-5) Banker's Algorithm Example

Methods for Handling Deadlocks

Deadlock Prevention

Deadlock Avoidance

- Deadlock Detection
- Ignore the problem

DEADLOCK AVOIDANCE Continue...

AVOIDANCE ALGORITHMS

Avoidance algorithms

- Single instance of a resource type:
 - ➤ Use a resource-allocation graph

- Multiple instances of a resource type:
 - > Use the banker's algorithm

BANKER'S ALGORITHM Example

Banker's Algorithm: Example

- Let a system with
 - Processes: 5 Processes (P0 through P4)
 - Resource types: 3 (A, B, and C)
 - Resource type A has 10 instances,
 - Resource type B has 5 instances,
 - Resource type C has 7 instances.
- Suppose that, at time T0, the following snapshot of the system has been taken:

	Allocation	Max	Available
	ABC	ABC	ABC
P_0	010	753	332
P_1	200	322	
P_2	302	902	
P_3	211	222	
P_4	002	433	

Banker's Algorithm: Solution 1/5

	Allocation	Max	Available
	ABC	ABC	ABC
P_0	010	753	3 3 2
P_1	200	322	
P_2	302	902	
P_3	211	222	
P_4	002	433	

Matrix, Need = Max – Allocation

	Need
	ABC
P_0	743
P_1	122
P_2	600
P_3	011
P_4	431

Banker's Algorithm: Solution_{2/5}

	Allocation	Max	Available		Need
	ABC	ABC	ABC		\overline{ABC}
P_0	010	753	3 3 2	P_0	743
P_1	200	322			
P_2	302	902		-	
P_3	211	222		_	
P_4	002	433			431
P ₁ P ₂ P ₃	3 0 2 2 1 1	9 0 2 2 2 2		$P_1 \\ P_2 \\ P_3$	1 2 6 0 0 1

- Now, by applying Safety Algorithm:
- Step 1:
 - ➤ Work: = Available: =332,
 - Finish[i]= false, where i=0 to 4

Banker's Algorithm: Solution 3/5

Need

			4 D C	4 D C	4 D C		ABC
			ABC	ABC	ABC	\mathbf{p}_{\circ}	743
		P_0	010	753	332	P_1	
		P_1	200	322		1	122
• Step 2:		P_2	302	902		P_2	600
Step 2.	work	P_3	211	222		P_3	
	VVOIR	P_{A}	002	433		P_4	431
	227	14	002	100			

```
1. for i=0, Finish[0]=false,
Needi <=work;
Need0 <=work;
743<=332 ->False
```

Allocation

Max

Available

Banker's Algorithm: Solution 3/5

Allocation N	lax Available ——	_
\overline{ABC} \overline{A}	\overline{BC} \overline{ABC} \overline{ABC}	C
	$P_0 = 7.43$	3
$P_0 010 7$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$)
P_1 200 3	77	
1	$P_2 = 6.00$)
_	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
P_3 2 1 1 2	22	
$P_4 002 4$	P_4 431	L

Need

```
2. for i=1, Finish[1]=false,
Needi <=work;
Need1 <=work;
122<=332 ->True

if True Then:
Work = Work + Allocation1;
= 332+200=532;
```

Finish[1] = True;

Step 2:

work 332

work **532**

Banker's Algorithm: Solution_{3/5}

	Allocation	Max	Available		
	ABC	\overline{ABC}	\overline{ABC}		ABC
	II D C	IIDC	11 D C	P_0	743
P_0	0 1 0	753	3 3 2	10	7 10
10	010	700	002	P_1	122
P_1	200	322		1 1	1 2 2
1 1	200	0 2 2		P_2	600
P_2	302	902		1 2	000
12	302	902		P_{2}	011
P_3	211	222		13	011
13	211	2 2 2		P_{A}	431
P_4	002	433		r_4	431
- 4	0 0 =	100			

• Step 2:

work **532**

Need

Banker's Algorithm: Solution 3/5

P_0 P_1		A B C 7 5 3 3 2 2	A B C 3 3 2	P_0 P_1	ABC 743 122 600
Step 2: P_2	211	902 222 433		P_3	$\begin{smallmatrix}0&1&1\\4&3&1\end{smallmatrix}$

Allocation

Max

Available

```
4.
       for i=3, Finish[3]=false,
       Needi <=work;
        Need3 <=work;
          011<=532 ->True
        if True Then:
       Work = Work + Allocation3;
               =532+211=743;
          Finish[3] = True;
```

work **532**

Need

work 743

Banker's Algorithm: Solution4/5

	Allocation	Max	Available		TVCCII
	<u>Intocurron</u>	MIUA	<u> </u>		ABC
	ABC	ABC	ABC	D	
P_0	010	753	332	P_0	743
10	010		002	P_1	122
P_1	200	3 2 2		- 1	
P_2	302	902		P_2	600
_				P_3	011
P_3	2 1 1	222			
P_4	002	433		P_4	431
P_4	002	433		- 4	101

Need

```
5.
       for i=4, Finish[4]=false,
       Needi <=work;
        Need4 <=work;
            431<=743 ->True
       if True Then:
       Work = Work + Allocation4;
               =743+002=745;
       Finish[4] = True;
```

work **743**

work 745

Banker's Algorithm: Solution4/5

	Allocation	Max	Available		iveeu
	Intocurron	1111111	21011111011		ABC
	ABC	ABC	ABC	D	
P_0	010	753	332	P_0	743
_	200	322	002	P_1	122
P_1				P_2	600
P_2	302	902		P_3	011
P_3	211	222		_	
	002	433		P_4	431
P_4	002	433			

Nood

```
6. for i=0, Finish[0]=false,
Needi <=work;
Need0 <=work;
743<=745 ->True
```

work **745**

if True Then:
Work = Work + Allocation0;
=745+010=755;

work **755**

Finish[0] = True;

Banker's Algorithm: Solution 5/5

	Allocation	Max	Available		TVCCH
	111100111011	IVIUA	<u> </u>		ABC
	ABC	ABC	ABC		
P_0	010	753	332	P_0	743
10	010	733	332	P_1	122
P_1	200	322		- 1	
P_2	302	902		P_2	600
12	302	902		P_3	011
P_3	2 1 1	222			
	0.0.2	422		P_4	431
P_4	0 0 2	433			

```
7.
        for i=2, Finish[2]=false,
        Needi <=work;
        Need2 <=work;
          600<=755 ->True
        if True Then:
        Work = Work + Allocation2;
           =755+302=10 5 7;
        Finish[2] = True;
        Safe Sequence:
        <P1, P3, P4, P0, P2>
```

work **755**

Need

work 10 5 7

Banker's Algorithm: Homework Q1

 Process P1 requests one additional instance of resource type A, and two instances of resource type C, so Request1 = (1, 0, 2).

• By Applying Resource Allocation Algorithm:
$$P_0$$
 P_0 P_1 P_2 P_3 P_4 P_2 P_3 P_4 P_4 P_5 P_6 P_6 P_6 P_7 P_8 P_8 P_9 P

Allocation

0.02

Max

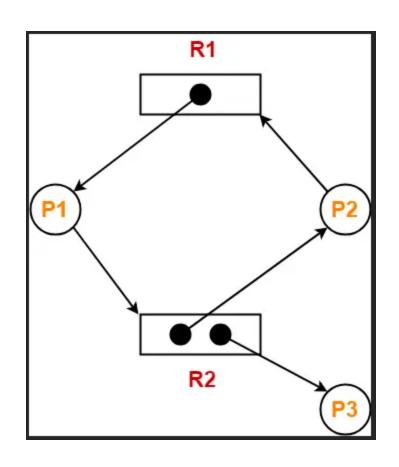
433

Available

- That is, that $(1, 0, 2) \le (3, 3, 2)$, which is true.
- We then pretend that this request has been fulfilled, and we arrive at the following new state:
 Allocation Need Available
- ABCABCABC010 743 230 P_0 Find the safe sequence. P_1 302 0.20302 600 P_2 211 P_3 0.11 P_{Δ} 0.02431

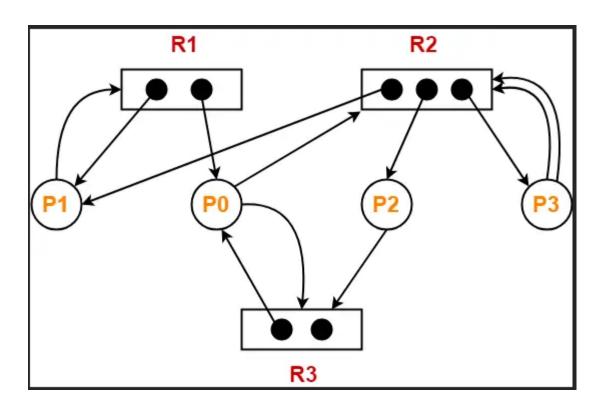
Banker's Algorithm: Homework Q2

 Take a look at the figure's resource allocation graph. Check for deadlocks in the system; if not, find a safe sequence.



Banker's Algorithm: Homework Q3

 Take a look at the figure's resource allocation graph. Check for deadlocks in the system; if not, find a safe sequence.



References

- 1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley.
- 2. William Stallings, "Operating Systems: Internals and Design Principles", 6th Edition, Pearson Education.
- D M Dhamdhere, "Operating Systems: A Concept based Approach", 2nd Edition, TMH.

