OPERATING SYSTEMS

Threads and Concurrency 13

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 2

UNIT 2: Threads and Concurrency

Introduction to Threads, types of threads, Multicore Programming, Multithreading Models, Thread creation, Thread Scheduling, PThreads and Windows Threads, Mutual Exclusion and Synchronization: software approaches, principles of concurrency, hardware support, Mutex Locks, Semaphores. Classic problems of Synchronization: Bounded-Buffer Problem, Readers -Writers problem, Dining Philosophers Problem concepts. Synchronization Examples - Synchronisation mechanisms provided by Linux/Windows/Pthreads. Deadlocks: principles of deadlock, tools for detection and Prevention.

Course Outline - Unit 2

13	4.1,4.2	Introduction to Threads, types of threads, Multicore Programming.	4	42.8
14	4.3,5.4	Multithreading Models, Thread creation, Thread Scheduling	4	
15	4.4	Pthreads and Windows Threads	4	
16	6.1-6.3	Mutual Exclusion and Synchronization: software approaches	6	
17	6.3-6.4	principles of concurrency, hardware support	6	
18	6.5,6.6	Mutex Locks, Semaphores	6	
19	6.7.1-6.7.3	Classic problems of Synchronization: Bounded-Buffer Problem, Readers -Writers problem, Dining Philosophers Problem concepts	6	
20	6.9	Synchronization Examples - Synchronisation mechanisms provided by Linux/Windows/Pthreads.	6	
21	Handouts	Demonstration of programming examples on process synchronization		
22	7.1-7.3	Deadlocks: principles of deadlock, Deadlock Characterization.	7	
23	7.4	Deadlock Prevention, Deadlock example	7	
24	7.6	Deadlock Detection	7	

Topics Outline

- Examples : Deadlock Avoidance & Deadlock Detection

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows, iS THE SYSTEM SAFE ? , after finding AND SATISFYING the need

RMax		
A	B	C
10	5	7

Available=>Rmax-Allocated		
A	B	C
3	3	2

Process	Allocation			Max		
	A	B	C	A	B	C
P0	0	1	0	7	5	3
P1	2	0	0	3	2	2
P2	3	0	2	9	0	2
P3	2	1	1	2	2	2
P4	0	0	2	4	3	3
Total	7	2	5	25	12	12

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			work<= Available => (Rmax-Allocated)		
A	B	C	A	B	C
10	5	7	3	3	2

i	Not Initialised

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	5	3	2

i	

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Work		
A	B	C
5+2	3+1	2+1

Process	Flag
P0	False
P1	True
P2	False
P3	True
P4	False

Safe Sequence				
P1	P3			

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	3	3	2

i	Not Initialised

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Process	Flag
P0	False
P1	False
P2	False
P3	False
P4	False

Safe Sequence				

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	3	3	2

i	1

Work		
A	B	C
3+2	3+0	2+0

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Process	Flag
P0	False
P1	True
P2	False
P3	False
P4	False

Safe Sequence				

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	3	3	2

i	2

Work		
A	B	C
5	3	2

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Process	Flag
P0	False
P1	True
P2	False
P3	False
P4	False

Safe Sequence				
P1				

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	7	4	3

i	4

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Process	Flag
P0	False
P1	True
P2	False
P3	True
P4	True

Work		
A	B	C
7+0	4+0	3+2

Safe Sequence				
P1	P3	P4		

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	7	4	5

i	2

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Work		
A	B	C
7+3	4+0	3+2+2

Process	Flag
P0	False
P1	True
P2	True
P3	True
P4	True

Safe Sequence				
P1	P3	P4	P2	

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	10	4	7

i	0

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Process	Flag
P0	True
P1	True
P2	True
P3	True
P4	True

Work		
A	B	C
10+0	4+1	7+0

Safe Sequence				
P1	P3	P4	P2	P0

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Safety Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. Suppose at time t1 if the snapshot of the system taken is as follows then find the system is in a safe state or not, after finding the need

RMax			Work		
A	B	C	A	B	C
10	5	7	10	5	7

i	0

Work		
A	B	C
10	5	7

Process	Allocation			Max			Need=>Max-Allocated		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1
Total	7	2	5	25	12	12	18	10	7

Process	Flag
P0	True
P1	True
P2	True
P3	True
P4	True

Safe Sequence				
P1	P3	P4	P2	P0

Deadlocks: Deadlock Avoidance & Detection

Banker's Algorithm: Resource Request Algorithm

1. Let ***Work*** and ***Finish*** be vectors of length ***m*** and ***n***, respectively Initialize:
   (a) ***Work*** = ***Available***
   (b) For ***i*** = **1,2, …, *n***, if $\boldsymbol{Allocation_i \neq 0}$, then
   Finish**[i]** = ***false***; otherwise, ***Finish*****[i]** = ***true***

2. Find an index ***i*** such that both:
   (a) ***Finish*****[*i*]** == ***false***
   (b) $\boldsymbol{Request_i \leq Work}$

   If no such ***i*** exists, go to step 4

Deadlocks: Deadlock Avoidance & Detection

Banker's Algorithm: Safety Algorithm

3. $Work = Work + Allocation_i$
   $Finish[i] = true$
   go to step 2

4. If $Finish[i] == false$, for some i, $1 \leq i \leq n$, then the system is in deadlock state. Moreover, if $Finish[i] == false$, then P_i is deadlocked

Algorithm requires an order of $O(m \times n^2)$ operations to detect whether the system is in deadlocked state

Deadlocks: Deadlock Avoidance & Detection

Banker's Algorithm: Safety Algorithm

3. Now, the algorithm assumes that the resources have been allocated and modifies the data structure accordingly.

   Available = Available - Request(i)

   Allocation(i) = Allocation(i) + Request(i)

   Need(i) = Need(i) - Request(i)

Algorithm requires an order of $O(m \times n^2)$ operations to detect whether the system is in deadlocked state

Deadlocks: Deadlock Avoidance & Detection - Examples

Example 1 - Resource Request Algorithm

Suppose we have 5 processes(P0, P1, P2, P3, P4) and 3 resource types(A, B, C) each having (10,5,7) instances. a new request comes from P1 =>(1,0,2). Can the resource request be granted immediately and safely

RMax		
A	B	C
10	5	7

Available=>Rmax-Allocated		
A	B	C
3	3	2

Request by P1		
A	B	C
1	0	2

Process	Allocation			Max		
	A	B	C	A	B	C
P0	0	1	0	7	5	3
P1	2	0	0	3	2	2
P2	3	0	2	9	0	2
P3	2	1	1	2	2	2
P4	0	0	2	4	3	3
Total	7	2	5	25	12	12

Need=>Max-Allocated		
A	B	C
7	4	3
0	2	0
6	0	0
0	1	1
4	3	1
18	10	7

Work		
A	B	C
5	3	2

Process	Flag
P0	False
P1	False
P2	False
P3	False

Topics Uncovered in this Session

- **Examples : Deadlock Avoidance & Deadlock Detection**

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com