- Operating System | Banker's Algorithm : Print all the safe state (or safe sequences)
- Operating System | Deadlock detection in Distributed systems
- Techniques used in centralized approach of deadlock detection in distributed systems
- How to prepare for GATE CSE - 2019
- GATE CSE Test Series - 2019 | Sudo GATE
- Functions of Operating System
- Introduction to memory and memory units
- PSU Recruitment through GATE
- Preemptive and Non-Preemptive Scheduling
- DBMS | Categories of End Users
- Computer Network | IPv4 classless Subnet equation
- Differences between IPv4 and IPv6
- Memory Hierarchy Design and its Characteristics

Operating System | Deadlock detection algorithm

If a system does not employ either a deadlock prevention or deadlock avoidance algorithm then a deadlock situation may occur. In this case-

- Apply an algorithm to examine state of system to determine whether deadlock has has occurred or not.
- Apply an algorithm to recover from the deadlock. For more refer- Deadlock Recovery

Deadlock Detection Algorithm:
The algorithm employs several time varying data structures:

- Available- A vector of length m indicates the number of available resources of each type.
- Allocation- An n*m matrix defines the number of resources of each type currently allocated to a process. Column represents resource and resource represent process.
- Request- An n*m matrix indicates the current request of each process. If request[i][j] equals k then process P_i is requesting k more instances of resource type R_j.

We treat rows in the matrices Allocation and Request as vectors, we refer them as $Allocation_i$ and $Request_i$.

Steps of Algorithm:

1. Let *Work* and *Finish* be vectors of length m and n respectively. Initialize *Work= Available*. For *i=0, 1, ..., n-1*, if $Allocation_i$ = 0, then *Finish[i]* = true; otherwise, *Finish[i]*= false.
2. Find an index i such that both
   a) *Finish[i] == false*
   b) *$Request_i$ <= Work*
   If no such *i* exists go to step 4.
3. *Work= Work+ $Allocation_i$*
   Finish[i]= true
   Go to Step 2.
4. If *Finish[i]== false* for some i, 0<=i<n, then the system is in a deadlocked state. Moreover, if *Finish[i]==false* the process P_i is deadlocked.

Most popular in GATE CS

- Operating System | Privileged and Non-Privileged Instructions
- Best College Prediction based on GATE CS All India Rank (AIR)/Score
- CPU Scheduling | Longest Remaining Time First (LRTF) Program

Computer Organization | Control Unit and design

Operating System | Kernel I/O Subsystem (I/O System)

Digital Logic | Latches

Introduction to basic Networking terminology

Digital Logic | Binary representations

Computer Organization | Horizontal micro-programmed Vs Vertical micro-programmed control unit

Computer Organization | RAM vs ROM

Fixed (or static) Partitioning in Operating System

Theory of computation | Halting Problem

What are the differences between HTTP, FTP, and SMTP?

Hardwired Vs Micro-programmed Control unit | Set 2

Variable (or dynamic) Partitioning in Operating System

Non-Contiguous Allocation in Operating System

Computer Organization | Read and Write operations in memory

For example,

	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	0	0	0	0	0	0
P1	2	0	0	2	0	2			
P2	3	0	3	0	0	0			
P3	2	1	1	1	0	0			
P4	0	0	2	0	0	2			

1. In this, Work = [0, 0, 0] &
   Finish = [false, false, false, false, false]
2. *i=0* is selected as both Finish[0] = false and [0, 0, 0]<=[0, 0, 0].
3. Work =[0, 0, 0]+[0, 1, 0] =>[0, 1, 0] &
   Finish = [true, false, false, false, false].
4. *i=2* is selected as both Finish[2] = false and [0, 0, 0]<=[0, 1, 0].
5. Work =[0, 1, 0]+[3, 0, 3] =>[3, 1, 3] &
   Finish = [true, false, true, false, false].
6. *i=1* is selected as both Finish[1] = false and [2, 0, 2]<=[3, 1, 3].
7. Work =[3, 1, 3]+[2, 0, 0] =>[5, 1, 3] &
   Finish = [true, true, true, false, false].
8. *i=3* is selected as both Finish[3] = false and [1, 0, 0]<=[5, 1, 3].
9. Work =[5, 1, 3]+[2, 1, 1] =>[7, 2, 4] &
   Finish = [true, true, true, true, false].
10. *i=4* is selected as both Finish[4] = false and [0, 0, 2]<=[7, 2, 4].
11. Work =[7, 2, 4]+[0, 0, 2] =>[7, 2, 6] &
    Finish = [true, true, true, true, true].
12. Since Finish is a vector of all true it means there is no deadlock in this example.

Recommended Posts:

Operating System | Deadlock detection in Distributed systems
Program for Deadlock free condition in Operating System
Operating System | Process Management | Deadlock Introduction
Deadlock Detection And Recovery
Techniques used in centralized approach of deadlock detection in distributed systems
Operating System | Banker's Algorithm
Operating System | Bakery Algorithm

Difference between Stop and Wait, GoBackN and Selective Repeat

Operating System | Memory Interleaving

Most visited in Operating Systems

Round Robin Scheduling with different arrival times

Deadlock, Starvation, and Livelock

Operating System | Boot Block

Logical vs Physical Address in Operating System

Operating System | Privileged and Non-Privileged Instructions

aastha98
Sophomore at Netaji Subhas Institute of Technology

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Improved By : EduardoNodarse

Article Tags : GATE CS Operating Systems

Practice Tags : Operating Systems

1

To-do Done

2.3

Based on 3 vote(s)

Most visited in GATE CS

Feedback/ Suggest Improvement | Add Notes | Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Previous
Operating System | Techniques to handle Thrashing

Next
Operating System | Benefits of Multithreading

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments | Share this post!

Arden's Theorem and Challenging Applications | Set 2
TCP Sequence Number | Wrap Around Concept
Computer Organization | Locality of Reference and Cache Operation
IEEE 802.11 Mac Frame
Difference between Byte Addressable Memory and Word Addressable Memory
Multilevel Cache Organisation
Operating System | Structures of Directory
Digital Logic | 5 variable K-Map
Concurrency Control | Polygraph to check View Serializabilty
Difference between Simultaneous and Hierarchical Access Memory Organisations
How to solve RSA Algorithm Problems?
COMPARE Instructions in 8085
Computer Organization | Register Allocation