Deadlock Prevention And Avoidance

Deadlock Characteristics

As discussed in the previous post, deadlock has following characteristics.

```
Mutual Exclusion.
Hold and Wait.
No preemption.
Circular wait.
```

Deadlock Prevention

We can prevent Deadlock by eliminating any of the above four condition.

Eliminate Mutual Exclusion

It is not possible to dis-satisfy the mutual exclusion because some resources, such as the tap drive and printer, are inherently non-shareable.

Eliminate Hold and wait

1. Allocate all required resources to the process before start of its execution, this way hold and wait condition is eliminated but it will lead to low device utilization. for example, if a process requires printer at a later time and we have allocated printer before the start of its execution printer will remained blocked till it has completed its execution.

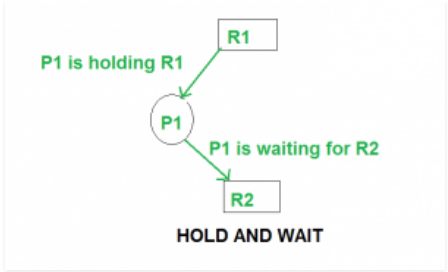
2. Process will make new request for resources after releasing the current set of resources. This solution may lead to starvation.

Most popular articles

Must Do Coding Questions for Companies like Amazon, Microsoft, Adobe, ...
Must Do Coding Questions Company-wise
Taking input from console in Python
Python list sort()

Disadvantages

Step by Step guide to Write your own WordPress Template



Eliminate No Preemption

Preempt resources from process when resources required by other high priority process.

Eliminate Circular Wait

Each resource will be assigned with a numerical number. A process can request for the resources only in increasing order of numbering.

For Example, if P1 process is allocated R5 resources, now next time if P1 ask for R4, R3 lesser than R5 such request will not be granted, only request for resources more than R5 will be granted.

Deadlock Avoidance

Deadlock avoidance can be done with Banker's Algorithm.

Banker's Algorithm

Bankers's Algorithm is resource allocation and deadlock avoidance algorithm which test all the request made by processes for resources, it check for safe state, if after granting request system remains in the safe state it allows the request and if their is no safe state it don't allow the request made by the process.

Inputs to Banker's Algorithm

1. Max need of resources by each process.

2. Currently allocated resources by each process.

3. Max free available resources in the system.

Request will only be granted under below condition.

1. If request made by process is less than equal to max need to that process.

2. If request made by process is less than equal to freely availbale resource in the system.

Example

Functions in Python

Most visited in Operating Systems

Functions of Operating System

Preemptive and Non-Preemptive Scheduling

Round Robin Scheduling with different arrival times

Memory Hierarchy Design and its Characteristics

Introduction to memory and memory units


```
Total resources in system:
A B C D
6 5 7 6
```

```
Available system resources are:
A B C D
3 1 1 2
```

```
Processes (currently allocated resources):
    A B C D
P1  1 2 2 1
P2  1 0 3 3
P3  1 2 1 0
```

```
Processes (maximum resources):
    A B C D
P1  3 3 2 2
P2  1 2 3 4
P3  1 3 5 0
```

```
Need = maximum resources - currently allocated resources.
Processes (need resources):
    A B C D
P1  2 1 0 1
P2  0 2 0 1
P3  0 1 4 0
```

Following are Gate Previous Year Question

http://quiz.geeksforgeeks.org/gate-gate-cs-2014-set-1-question-41/

http://quiz.geeksforgeeks.org/gate-gate-cs-2014-set-3-question-41/

http://quiz.geeksforgeeks.org/gate-gate-cs-2010-question-46/

References

https://en.wikipedia.org/wiki/Banker's_algorithm

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Recommended Posts:

DBMS | Introduction to TimeStamp and Deadlock Prevention Schemes

Most Trending Articles

- Practice for cracking any coding interview
- Implementing a Linked List in Java using Class
- Coin game of two corners (Greedy Approach)
- OOPs | Object Oriented Design
- Compare two Strings in Java
- Interview Questions for Java Professionals
- How to get rid of Java TLE problem
- Quick Sort vs Merge Sort
- Design a data structure for LRU Cache

Zombie Processes and their Prevention

Deadlock, Starvation, and Livelock

Deadlock Detection And Recovery

Operating Systems | Deadlock | Question 2

Operating Systems | Deadlock | Question 1

Operating System | Deadlock detection algorithm

Operating System | Process Management | Deadlock Introduction

Program for Deadlock free condition in Operating System

Operating System | Deadlock detection in Distributed systems

Techniques used in centralized approach of deadlock detection in distributed systems

Operating System | Translation Lookaside Buffer (TLB)

Operating System | The Linux Kernel

Computer Organization | Random Access Memory (RAM) vs Hard Disk Drive (HDD)

Article Tags : Articles | Operating Systems | Deadlocks

Practice Tags : Operating Systems

5

To-do Done

2.7

Based on 20 vote(s)

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 | Process Management | Deadlock Introduction

Next
Deadlock Detection And Recovery

Understanding The Coin Change Problem With Dynamic Programming

Implement a stack using singly linked list

Reverse an array in Java

Delete a Node from linked list without head pointer

Remove duplicates from unsorted array

AbstractCollection in Java with Examples

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

Load Comments

Share this post!