

Define a deadlock ? Write down the conditions responsible for deadlock? How can we recover from deadlock ?

A deadlock is a situation in computer systems where two or more processes are blocked, and each process is waiting for another process to release a resource that it needs in order to proceed.

In other words, a deadlock occurs when two or more processes are stuck in a loop, waiting for each other to finish their work, and neither can proceed until the other finishes.

There are four necessary conditions for a deadlock to occur:

1. Mutual exclusion: At least one resource must be held in a non-sharable mode, meaning only one process can use it at a time.
2. Hold and Wait: A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.
3. No Preemption: Resources cannot be forcibly taken away from a process. A process must release its resources voluntarily.
4. Circular Wait: There must be a circular chain of two or more processes, each of which is waiting for a resource held by the next process in the chain.

To recover from a deadlock, there are several strategies that can be used:

1. Prevention: By designing the system in such a way that at least one of the four necessary conditions for a deadlock is not met, we can prevent deadlocks from occurring. For example, we can ensure that resources are not held for too long, or we can use preemption to take away resources from a process that has been waiting for a long time.

Define a deadlock ? Write down the conditions responsible for deadlock? How can we recover from deadlock ?

2. Detection and Recovery: If prevention is not possible or practical, we can use a deadlock detection algorithm to identify when a deadlock has occurred. Once a deadlock is detected, we can use recovery strategies such as killing one or more processes, rolling back the system to a previous state, or forcing processes to release resources.

3. Avoidance: A more proactive approach is to use a resource allocation algorithm that ensures that requests for resources do not lead to deadlocks. This can be done by maintaining a global state of the system and predicting whether a resource allocation will lead to a deadlock before granting the request.

Related Posts:

1. Operating System: A List of Video Lectures RGPV Notes
2. GATE, Context switch calculation in SRTF algorithm | Prof. Jayesh Umre
3. Introduction to Operating Systems
4. Different Types of OS
5. Characteristics and features of an OS
6. Operating systems services
7. System Calls in OS
8. File Systems
9. How many page faults
10. Process State Diagram
11. Operating System Scheduler
12. FIFO page replacement algorithm
13. LRU page replacement algorithms
14. Optimal page replacement algorithm
15. SRTF shortest remaining time first
16. OS 4

Define a deadlock ? Write down the conditions responsible for deadlock? How can we recover from deadlock ?

17. OS 3
18. Os 2
19. Os 1
20. CBSE NET 2004 38
21. Cbse net 2004 37
22. Cbse net 2004
23. CBSE Net 2017
24. Ugc net 2017 solved
25. NET 4
26. NET 1
27. Net 28
28. Net 26
29. Net 50
30. Net 49
31. Net 48
32. Net 46
33. Net 44
34. Net 40
35. Net 39
36. GATE, Longest Remaining Time First Algorithm | Prof. Jayesh Umre
37. GATE SRTF | What is the total waiting time for process P2?
38. GATE Calculate Total Waiting Time SRTF algorithm | Prof. Jayesh Umre
39. Memory management
40. Concept of Threads
41. Process concept
42. Directory Structure OS
43. Contiguous disk space allocation method

Define a deadlock ? Write down the conditions responsible for deadlock? How can we recover from deadlock ?

44. File systems
45. Types of os
46. Evolution of os
47. Functions of os
48. Why is operating system a mandatory software?
49. Bankers algorithm problems
50. Diploma Linux Unit 3
51. RGPV Diploma Linnux Unit 2
52. Program to print string in reverse order
53. Program to implement while loop in Linux
54. Program to implement for loop using sequence keyword in Liux
55. Program to implement different types of increment in Linux
56. For loop without in keyword in Linux
57. Program to implement for loop using in keyword in Linux
58. Multiple Processor Scheduling
59. What do you mean by Virtual Memory? Write down its advantages?
60. Compare Paging and Segmentation?
61. What is Process Scheduling, CPU Scheduling, Disk Scheduling? Explain Short, Medium and Long term Scheduler?
62. Explain concept of a process with its components ?
63. Explain the following in brief Contiguous and Linked list allocation for implementing file system?
64. Explain various Disk scheduling algorithms with Illustrations ?
65. Define process and thread. What is PCB ? Explain its various entries with their usefulness ?
66. Discuss advantages and disadvantages of the Buffer cache ?
67. Explain different types of OS with examples of each ?

Define a deadlock ? Write down the conditions responsible for deadlock? How can we recover from deadlock ?

68. What is an Operating System? Write down its desirable characteristics ?
69. What are the various services provided by Operating system ?
70. What do you mean by PCB? Where is it used? What are its contents? Explain.
71. What is Binary and Counting semaphores ?
72. What is File? What are the different File attribute and operations?
73. What are System call? Explain briefly about various types of system call provided by an Operating System?
74. Describe necessary conditions for deadlocks situation to arise.
75. What are points to be consider in file system design? Explain linked list allocation in detail?
76. Write a Semaphore solution for dining Philosopher's problem?
77. Consider the following page reference string:1,2,3,4,5,3,4,1,2,7,8,7,8,9,7,8,9,5,4,5.
How many page faults would occur for the following replacement algorithm, assuming four frames:a) FIFO b) LRU
78. Explain CPU schedulers in operating system?
79. Write the different state of a process with the help of Process state deagram?
80. What is Mutex in operating system?
81. Explain Network operating system?
82. What do you mean by paging in operating system ?