

Project Proposal

Project to be undertaken

Our decision to concentrate on "System Deadlocks" as the subject of our research was motivated by the fact that a system deadlock occurs when two or more processes are stalled while awaiting resources that they each need.

Topics to be covered

A deadlock happens when several processes are impeded because they are all holding resources and waiting for resources held by other processes.

Figure 1: Deadlock problem

Our research project on system deadlocks will cover a variety of topics, including the history and model of deadlocks, the four necessary conditions for a deadlock, and various methods for preventing, avoiding, detecting, and recovering from deadlocks.

Investigation

Our research seeks to investigate the fundamental causes of system deadlocks, particularly the four necessary conditions for a deadlock.

Mutual Exclusion: According to this requirement, at least one resource must be stored in a non-shareable mode.

Hold-and-wait: When a process retains one or more resources while waiting for other resources that are held by other processes.

No Preemption: This restriction prohibits forcefully removing resources from a process that is actively utilizing them.

Circular Wait: This state develops when a group of processes are all waiting on resources that one or more of them are holding.

Development

In the development section of our project, we will be researching efficient methods and techniques to prevent, avoid, detect, and recover from deadlocks.

Prevention: means creating the system in such a way that deadlocks are avoided. This can be accomplished by ensuring that the four necessary conditions for a deadlock are not met.

Avoidance: includes searching for potential deadlocks before distributing resources to processes. This can be accomplished by using algorithms such as the Banker's algorithm.

Detection and Recovery: This involves finding deadlocks and then recovering from them. This can be accomplished by using algorithms such as the Resource Allocation Graph (RAG) and the Safety Algorithm.

Resource allocation: This entails distributing resources in a way that lessens the likelihood that deadlocks will occur.

Prevention and Recovery: To deal with deadlocks, a mix of preventative and recovery techniques is used.

Regardless of the method used, understanding the system and its needs, selecting the proper approach, and testing thoroughly are all essential.

Experimentation

We will use Ubuntu, a Linux-based operating system, to run the suitable application that illustrates deadlocks.

Figure 2: expected experimentation

Expected results

After completing our research project, we aim to apply the knowledge we've learned in class to improve our understanding of deadlocks and how to prevent, avoid, detect, and recover from them.

Final form of the project

We are expecting to finish a study report and a PowerPoint presentation for our research assignment. The study report will include a detailed description of the problem, the methods used, the results, and the conclusions.

References

E. R. S. BANGER, "Multiprocessor Operating System: Examples, Types, Advantages, & Feature!," 6 Feb 2023. [Online]. Available: <https://www.includehelp.com/multiprocessor-operating-system/>

A. Sugandhi, "What is Deadlock in Operating System (OS)?," 23 January 2023. [Online]. Available: <https://www.includehelp.com/operating-systems/deadlock-in-operating-system/>

P. Jain, "Deadlock and method for handling deadlock," 5 October 2018. [Online]. Available: <https://www.includehelp.com/operating-systems/deadlock-and-method-for-handling-deadlock.aspx>

[Accessed 10/10/2023].