CPU Deadlock Performance Report

# By: Dylan Silva-Rivas

# Deadlock Overview

I will explain all fo the necessary deadlock conditions and explain their relevance.

1. Mutual exclusion: A single process can use a resource at a time.
2. Hold and wait: Process is holding one or more resources while waiting for resources.
3. No Preemption: Resource can’t be allocated unless another process releases the resource.
4. Circular wait: A set of processes waiting for resources while taking resources in a circular fashion.

While looking into CPU scheduling algorithms, none of them inherently cause deadlock. The way a CPU scheduling algorithm is implemented can cause deadlock. Which is another reason why the deadlock conditions are necessary for this simulation.

# Deadlock Detection

I have decided to use banker’s algorithm for deadlock detection algorithm.

# Simulation of Deadlock Scenarios

# Deadlock Handling

# Performance Impact

# Comparison of Detection and Handling

# Conclusion