

Solution to Deadlock: Stack-based Priority Ceiling (SBPC) Protocol

CS303
Autumn 2018

Criteria for Deadlock Occurrence

- Four (w.r.t accessing of shared resources)
 - Mutual Exclusion
 - Hold and wait
 - No preemption (of shared resources before intended unlocking)
 - Circular waiting
- Typically, for deadlock to occur ALL criteria shall hold
- Solutions to DL:
 - Prevention: Any of the criteria if NOT allowed to become TRUE, DL can be **prevented**
 - Avoidance: Deadlock can be avoided by identifying when system is in the LEADS-to-DL state i.e. in a state, which would in the next step land in DL state
 - Identification and Mitigation

Stack-based Priority Ceiling Protocol

- Scheduling policy: Priority-based scheduling
- Protocol approach: process scheduling and resource allocation in an integrated fashion
- Significant points to note:
 - Scheduler follows priority based scheduling
 - Resources have fixed priorities and these priorities are defined as following
 - If a shared resource R_i would be shared by processes P_j , P_k and P_l
 - For processes P_j , P_k and P_l the priorities are natural numbers j , k and l , respectively.
 - Then resource R_i priority is given by $\max([j, k, l])$

Stack-based Priority Ceiling Protocol

- Scheduling policy: Priority-based scheduling
- Protocol approach: process scheduling and resource allocation in an integrated fashion
- Significant points to note:
 - Scheduler follows priority based scheduling
 - Resources have fixed priorities and these priorities are defined as following
 - If a shared resource R_i would be shared by processes P_j , P_k and P_l
 - For processes P_j , P_k and P_l the priorities are natural numbers j , k and l , respectively.
 - Then resource R_i priority is given by $\max([j, k, l])$