# Definitions

Resource Ceiling: Highest preemption level among all tasks that can use the shared resource (statically defined)  
System Ceiling: Largest resource ceiling among all currently locked shared resources (changes at runtime)

# Phase 2

1. System Ceiling– OS\_SYSTEM\_CEILING
   1. Stack used to store Resource Ceilings
   2. head->data->Resource\_Ceiling->Deadline;
      1. Find minimum deadline function
         1. Min deadline = highest preemption leve
   3. Pushing and popping mutex pointer
      1. Push --> pend
      2. Pop --> post
2. Resource Ceiling 🡪 highest preemption level among all task that can use shared resource
   1. T1 is the resource ceiling for all mutexes
   2. Periods: T1 = 3, T2 = 4, T3 = 5.
   3. Period implicitly the deadline also
   4. Store as TCB pointer
   5. Store inside mutex object
3. Mutex pend
   1. Update system ceiling
4. Mutex post
   1. Update system ceiling
   2. Unblock tasks from RB Tree that have higher preemption (lower deadline) than the newly changed system ceiling.
      1. If not we risk the blocked tasks never being able to run
5. OSSched
   1. If a task that is NOT the current task wants to pre-empt, need check   
      deadline from EDF < deadline (from system ceiling) 🡪 higher priority than system ceiling
      1. If fail check, remove task from ready list, add to RB tree. Run EDF scheduler again for new task to run; run the above check again.
   2. If task picked by EDF is current task, resume current task

To improve:

* Use event flags for synchronous release
* Use memget for memory management