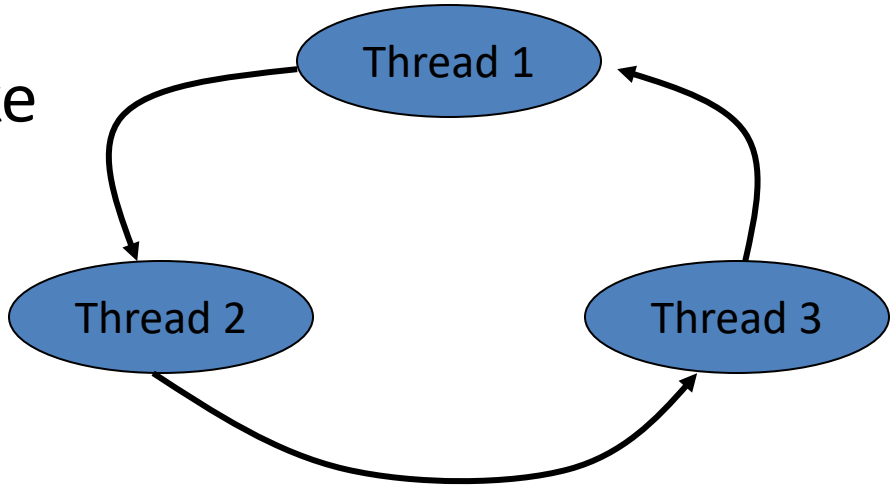
Deadlock

การติดตาย

https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Deadlock

- Each thread is waiting on a resource held by another thread
  - So, there is no way to make progress



https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Necessary Conditions for Deadlock

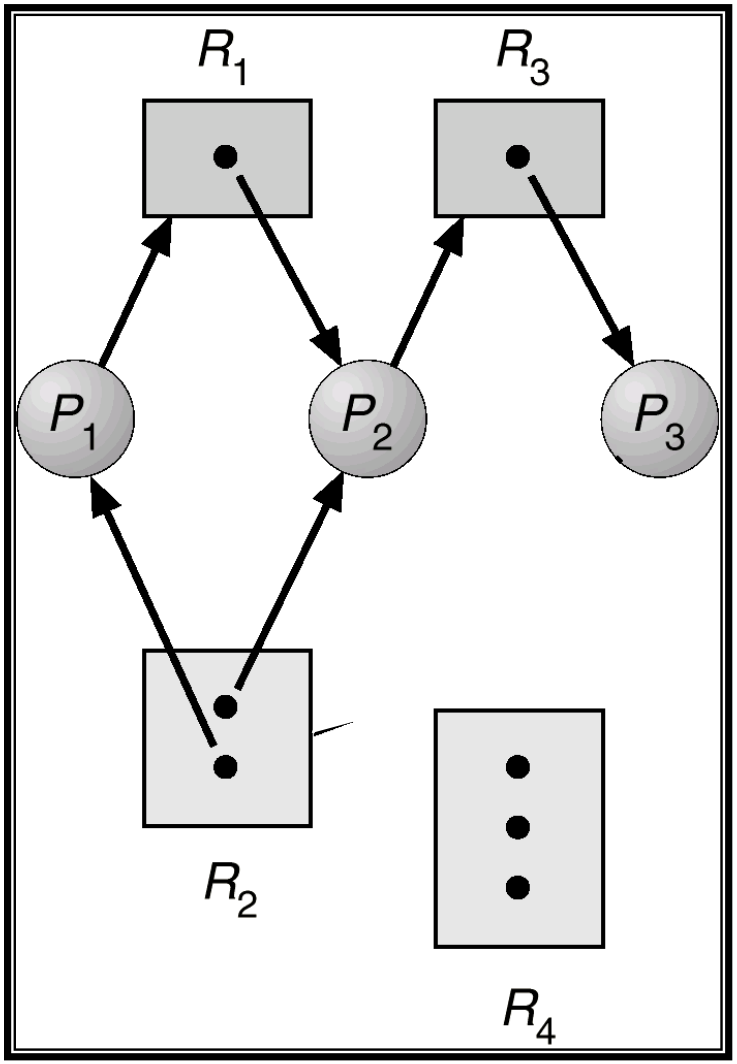
- **Mutual exclusion**
  - Resources cannot be shared
    - e.g., buffers, locks
- **Hold and wait (subset property)**
  - A thread must hold a *subset* of its resource needs
    - And, the thread is waiting for more resources
- **No preemption**
  - Resources cannot be taken away
- **Circular wait**
  - A needs a resource that B has
  - B has a resource that A has

https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Resource Allocation Graph without a Deadlock

P => R: request edge

R => P: assignment edge

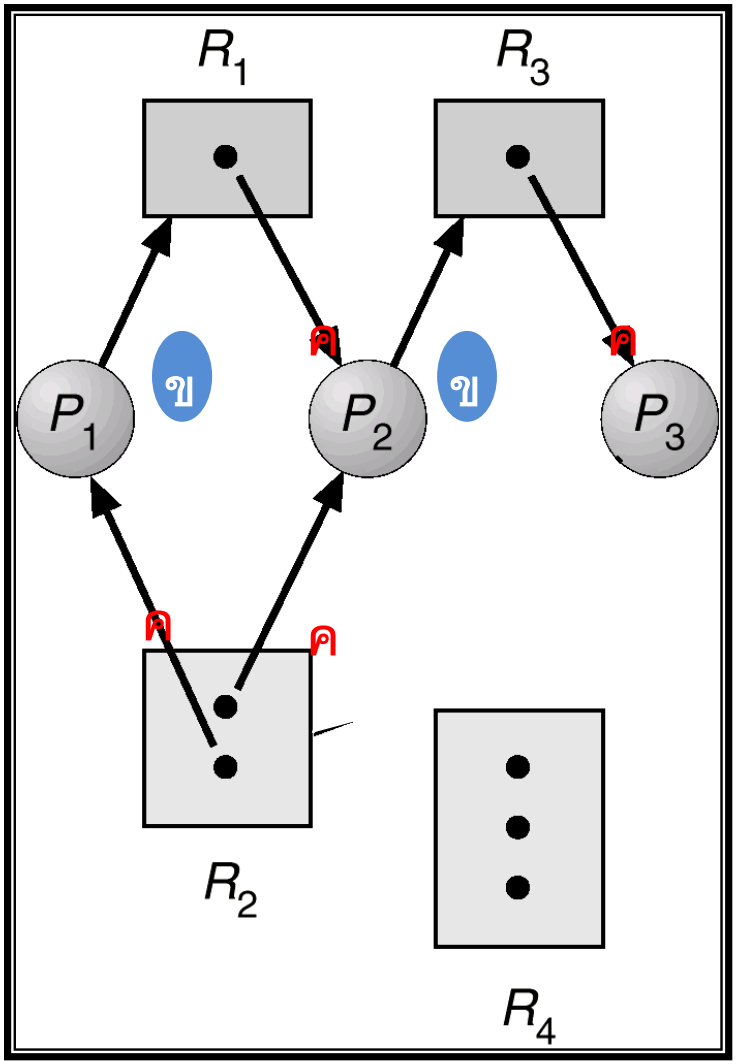


https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Resource Allocation Graph without a Deadlock

P => R: request edge

R => P: assignment edge

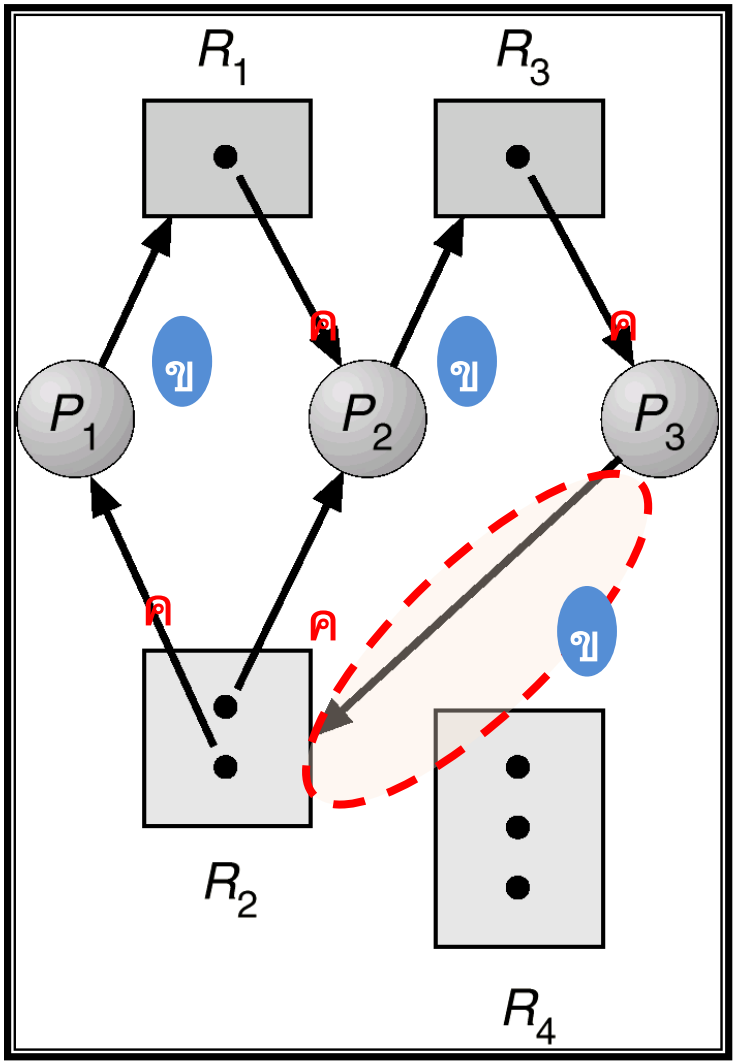


https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Resource Allocation Graph with a Deadlock

P => R: request edge

R => P: assignment edge



https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Approaches to Deadlock

1. Avoid threads
2. Deadlock prevention
   - Break up one of the four necessary conditions
3. Deadlock avoidance
   - Stay live even in the presence of the four conditions
4. Detect and recover

https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Deadlock Prevention

Can we eliminate:

- Mutual exclusion?
- Hold and wait (subset property)?
- No Preemption?
- Circular waiting?

https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Deadlock Avoidance : Bankers Algorithm

- Basic idea: ensure that we always have an “escape route”
  - The resource graph is reducible
- This can be enforced with the bankers algorithm:
  - When a request is made
    - Pretend you granted it
    - Pretend all other legal requests were made
  - Can the graph be reduced?
    - If so, allocate the requested resource
    - If not, block the thread

https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt

Deadlock Detection and Recovery

- Not commonly used
  - Detection is expensive
  - Recovery is tricky
- Possible exception: databases

https://courses.cs.washington.edu/courses/cse451/06au/slides/os-deadlock.ppt