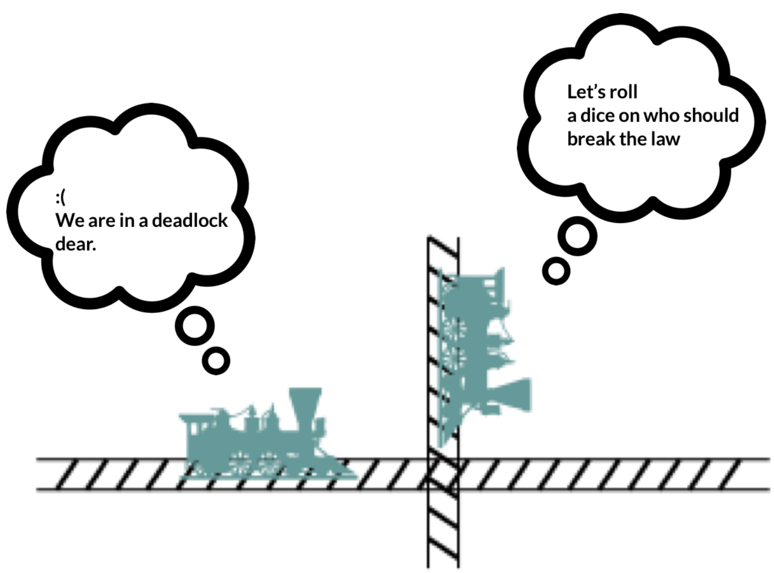
CSC369 Week 10 Notes

Hyungmo Gu

May 31, 2020

- Deadlock Defined
  - **Google Definition:** Is a situation one typically involving opposing parties, in which no progress can be made.
  - Is permanent
  - Happens to set of processes that
    - Compete for same system resources
    - Communicate with each other
- Example of Deadlock
  - Law passed by Kansas Legislature in in early 20th century
    - "When two trains approach each other at a crossing, both shall come to a full stop and <u>neither</u> shall start upon again until the other is gone"



- Conditions for Deadlock

- Neccesary and Sufficient Conditions
  1. Mutual Exclusion
     * Only one process may use a resource at a time
  2. Hold and wait
     * A process may hold allocated resources while awaitng assignment of others
  3. No preemption
     * No Resource can be forcibly removed from a process holding it
  4. Circular wait
     * Each process must be waiting for a resource which is being held by another process, which in turn is waiting for the first process to release the resource [3]

Aside

1. Wait. Necessary condition? [1]
   - We say N is a necessary condition for S if we don't have N, we won't have S.
2. Wait. Sufficient condition? [1]
   - We say S is a necessary condition for N if we have S, then we know that N must follow, i.e. $S \Rightarrow N$
3. Hold on. How about necessary and sufficient condition?
   - Is when necessary and sufficient conditions are put together similar to if and only if [2]

Sufficient Condition
Necessary Condition
Is Sufficient
Is Necessary
Is Necessary and Sufficient

Truth table

S	N	$S \Rightarrow N$	$S \Leftarrow N$	$S \Leftrightarrow N$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

References

1) Fayetteville State University: Necessary and Sufficient Conditions, link

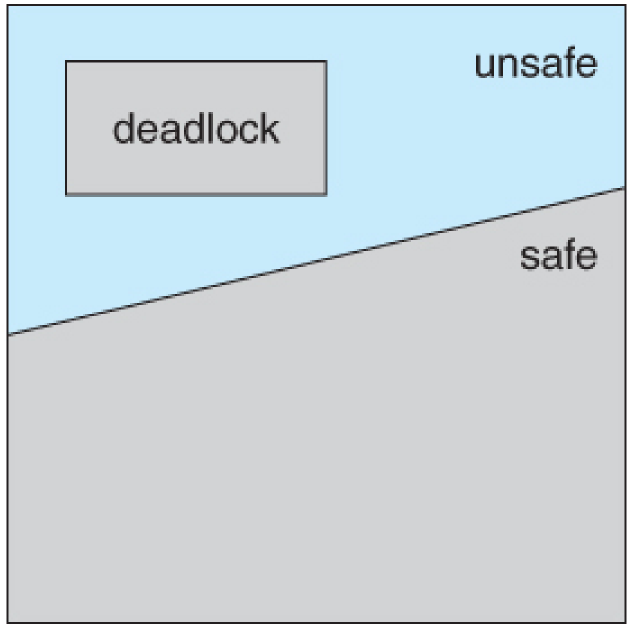
2) Wikipedia: Necessity and Sufficiency, link
3) Wikipedia: Deadlock, link

- Solutions
  - Prevention
    * Ensures that at least one of the necessary conditions to deadlock will never occur [1]
  - Avoidance
    * Ensures that the system will not enter an unsafe mode [1]
  - Detection and Recovery
    * Ensures that the system recovers from deadlock if a deadlock has occured in the system. [2]
  - Do Nothing and hope
    * Is done by Windows, Linux, and JVM
    * Works until eventually the deadlock snowballs, no longer functions, and requires manual intervention (Press power, and restart) [3]

References

1) pediaa: What is the Difference Between Deadlock Prevention and Deadlock Avoidance, link
2) Geeks for Geeks: Recovery from Deadlock in Operating System, link
3) Casanova H. (2018). *Synchronization: Deadlocks*. Medium. link

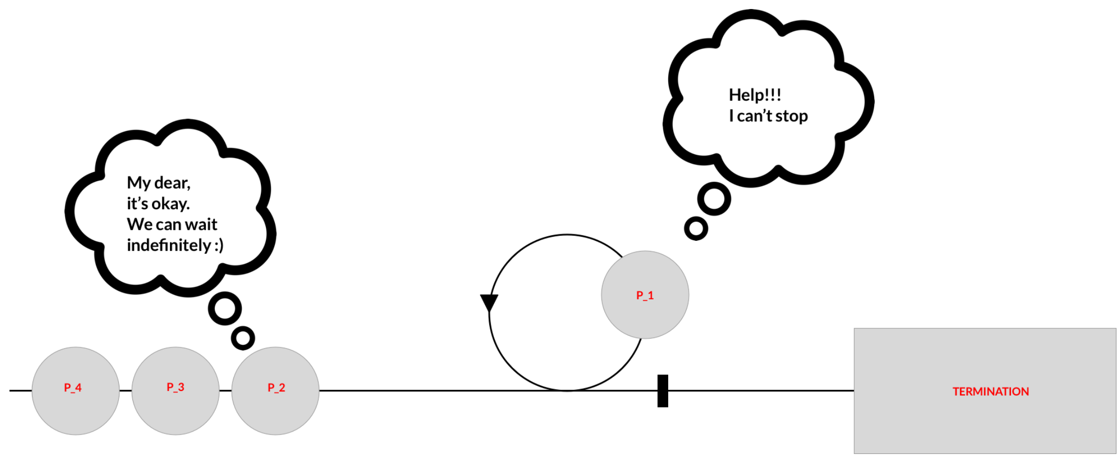
- Safe States
  - A state is **safe** if the system can allocate all resources requested by all processes without entering a deadlocked state [1]



References

1) University of Chicago: Deadlock Avoidance, link

- Unsafe States & Algorithm
  - An **unsafe** state is one which is not safe
- What is Atomicity?
  - **atomic** means without interruption (from week 2 notes)
  - e.g. Using outdoor ATM machine (Single account)
    - Only one person can use the machine.
    - Others need to wait in line
    - None can interrupt the transaction
- Deadlock and Starvation
  - A set of threads is in a **deadlocked** state when every process in the set is waiting for an event that can be caused only by another proess in the set.

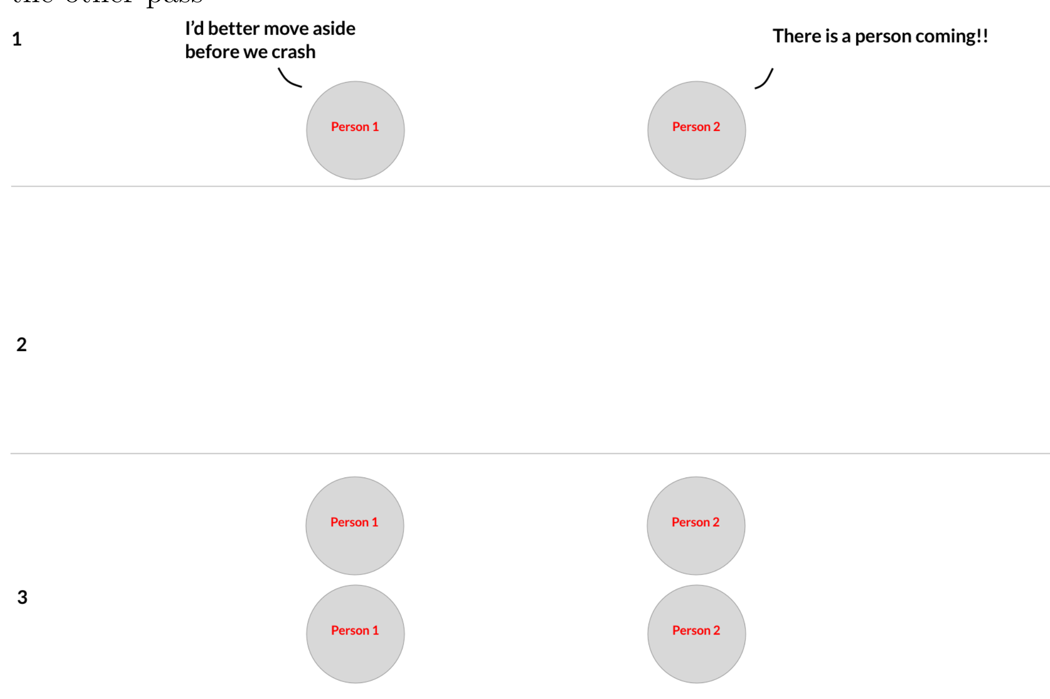


- A thread is suffering **starvation** if it is waiting indefinitely because other threads are in some way preferred

- Livelock
  - Occurs when a set of processes continually retry some failed operation and prevent other processes in the set from making progress

  Example:

  1. Two people who meet face-to-face in a corridor, and both move aside to let the other pass [1]



References

1) Guru 99: Livelock: What is, Example, Difference with Deadlock, link