compile → link → load → run time

global

before | after

Thread synchronization

Static
- before run-time
- unchangable @ run-time
- could mean private

Dynamic
- at run-time
- not fixed until run-time
- changable @ run-time

Static buffer = global variable

strtok & ~~strtok_r~~

→ static/global + local/stack ☆ → see online table

thread-safe concerns global variables

↑

all concerns storage + linkage classes

atomic - one instruction

count++ is not atomic

- for monitor of Hoare model
  - lock & unlock need to be reentrant to prevent dead lock

- writing after signal is called
  - signal & wait (Hoare's model) or signal & continue (in Java)?
  - if code after signal, could modify conditions that could mess up intentions

Deadlock
- two (or more) processes, each holding a resource, requesting something that is waiting on another request that is waiting on another request... (cycle)
  - waiting on each other

- prevention
  - ~~cycle might cause~~
  - deadlock doesn't come with a program
    - only happens sometime
    - possible, not necessarily present
- unsafe & safe (see lecture notes)

- conditions (4)
  - all must be met (necessary)
  - fulfill requirements (sufficient)
  1. mutual exclusion
  2. hold & wait [for another resource]
  3. no preemption
  4. circular wait / cycle

deadlock states

- prevention or avoidance

how to prevent preemption (prevention)
- let resources go to be put w/ waiting on queue

2 ways to handle deadlock
- see lecture notes on deadlock ★
1. avoid / prevent
2. detect / recover

avoidance
- banker's algorithm
- particularly needed if kernel has race condition/deadlock
  - → OS deadlocks / blue screen

thrashing
job ⟷ ready
(both wait)

CPU utilization going down
- thrashing
- lack of applications running

detecting / recovering
- extreme: → act when issue occurs
- average: → check when CPU utilization low && less than 40%

→ how to deal
- inform operator & deal manually
- recover automatically
  - abort one or more process to break circular wait
  - preempt resources to break the wait
    - from processes → still needs resource