$\frac{1}{(1-P)+P/c}$

RCH 307
H-8

current coroutine ~~last suspended~~ ~~current~~

resume vs suspend ↑ last resumer

P = prolagen --

V = verlagen ++

(bin) Mutex lock specifically allows owner to mult. acq. & only owner can un-acq.

uSemaphore TryP ret true if acquired (never blocks)

Barrier(n) is used to synch the first calls to the same point in time.

uOwnerLock = multi-acquisition mutex lock

uCondLock =

Dekker: ~~[illegible]~~ I want in, ~~and~~ if you do, I don't & loop) (if you don't, I crit

5 things:
- safety
- CPU time
- in crit section ⟺ doing critical stuff. only one at a time
  no others can prevent.
- starvation
- no indef postponement.

Static call: Always call to the same point (known @ compile time)

static Return: ~~always go~~ return loc'n is not checked ~~[illegible]~~ @ runtime

Mellor-Crummey & Scott: Ticketing System through linked List

Peterson: While ~~I want in~~ you're "want in" queued & I'm ~~the last~~ not the first (last) to request "in"; crit() I don't "want in"

~~$\lim_{c \to \infty}$~~ ~~$\frac{1}{1-p+p/c}$~~

CAA: CAA(&A, &B, C) if A == B A = C ret true
else ret false

Osacquire - Mutex For Streams