race condition → lock

Mutex → Mutual Exclusion

Only 1 thread can have the lock

self._lock = threading.Lock()

with self._lock:

logging

dead lock & RLock

D I W E

Lock bundle code up, during execution, not switch thread

primitives

producer - consumer threading

Threading & process synchronization

① using 2 locks, first in first out

② using queue pipeline only stores 1 value

threading.Event

queue.Queue → counter

③ Better queue

Semaphore

Timer

Barrier

threading Jim Anderson 3/21/21

daemon will shut down immediately when the program exits.

x.join() wait thread to finish

# comment out, still wait for it

↑ regular thread

{ regular thread

daemon thread x.join()

regular thread wait

no matter x.join(), both

daemon thread

x.join() wait

no join(), main thread does not wait

{ main thread

start with threading.Thread obj

A group of threads:

concurrent.futures

executor.map

.submit ThreadPoolExecutor

(max_workers = 2)

randint(a,b) $a \le r \le b$

producer – consumer problem

probably more difficult than the concepts