

race condition → lock

Mutex → Mutual Exclusion

Only 1 thread can have the

() self.lock = threading.Lock()

with self._lock: logging

dead lock & RLock

Lock bundle code up, during primitives execution, not switch thread

() producer - consumer threading

Threading & process synchronization

① using 2 locks, first in first out
pipeline only stores 1 value

② using queue

() threading.Event
queue.Queue → counter
Semaphore
Timer
Barrier

③ Better queue

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
no matter x.join(), both wait

() daemon thread
x.join() wait
no join(), main thread does
{ main thread not wait
start with threading.Thread obj

() A group of threads:

concurrent.futures
executor.map
.submit Thread Pool Executor
(max_workers=2)

$\text{randint}(a, b)$ as $r \leq b$

producer - consumer problem

probably more difficult than the concepts

