

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 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

- ③ Better queue
threading.Event
queue.Queue → counter
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 x.join()
} daemon thread x.join()

regular thread wait

no matter x.join(), both

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)

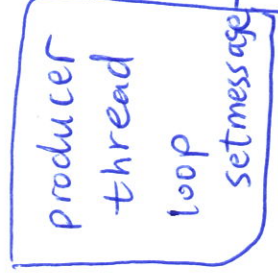
randint(a,b) $a \leq r \leq b$

• Producer - consumer problem

probably more difficult than the

concepts

Pipeline



loop 1

Start before
setmessage

It has 2 locks

After

Release Consumer_lock

loop 2

Until producer_lock is
released, wait

it has producer_lock

2 locks

Consumer_lock
producer_lock

loop 1

before start

It has 2 locks

After

Release Consumer_lock

loop 2

Until producer_lock is
released, wait

it has producer_lock

has Consumer_lock

After

release producer_lock

loop 2