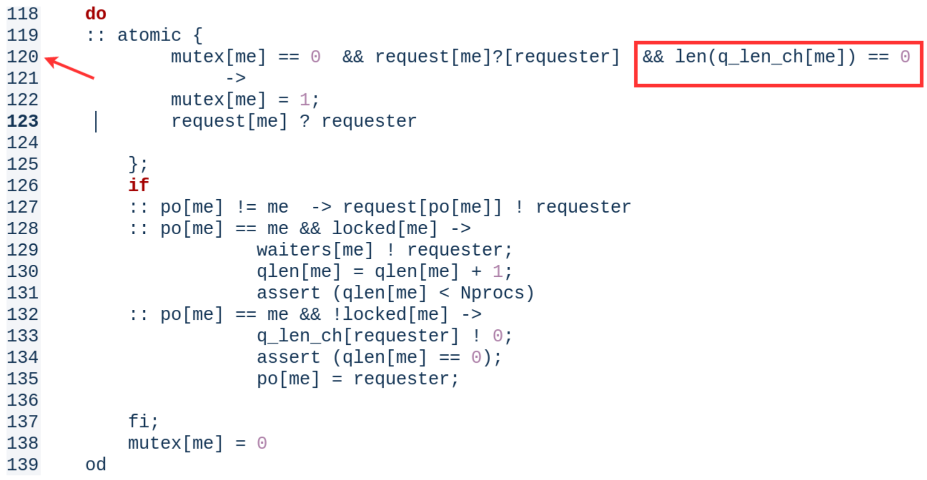
Assignment 3 Q3

tripti.agarwal

February 2022

1 Q3

You can find the full code Code Link

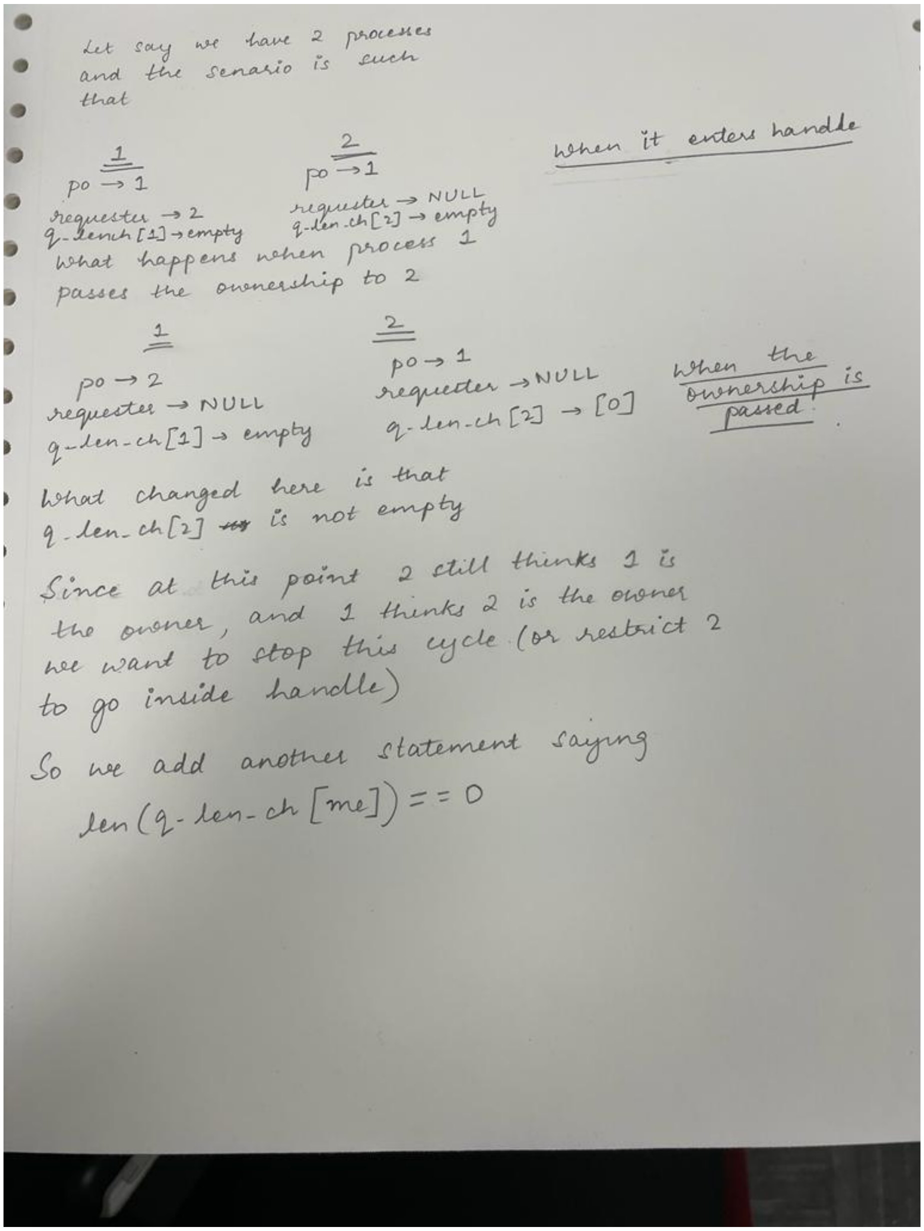


```
do
:: atomic {
        mutex[me] == 0  && request[me]?[requester]  && len(q_len_ch[me]) == 0
             ->
        mutex[me] = 1;
 |      request[me] ? requester

    };
    if
    :: po[me] != me  -> request[po[me]] ! requester
    :: po[me] == me && locked[me] ->
                waiters[me] ! requester;
                qlen[me] = qlen[me] + 1;
                assert (qlen[me] < Nprocs)
    :: po[me] == me && !locked[me] ->
                q_len_ch[requester] ! 0;
                assert (qlen[me] == 0);
                po[me] = requester;

    fi;
    mutex[me] = 0
od
```

Figure 1: Code snippet with correction labeled



Let say we have 2 processes and the senario is such that

1	2
po → 1	po → 1
requester → 2	requester → NULL
q-lench[1] → empty	q-len-ch[2] → empty

when it enters handle

What happens when process 1 passes the ownership to 2

1	2
po → 2	po → 1
requester → NULL	requetter → NULL
q-len-ch[1] → empty	q-len-ch[2] → [0]

when the ownership is passed.

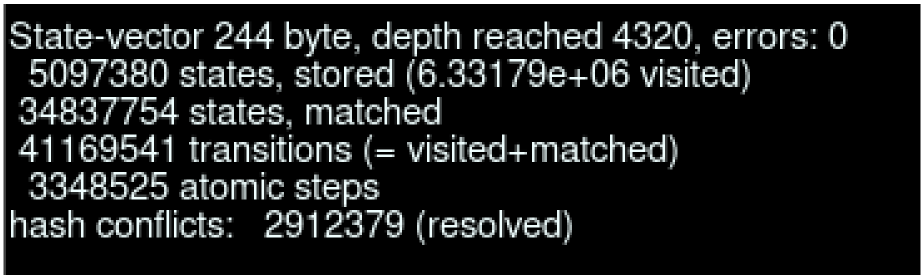
What changed here is that q-len-ch[2] ~~is~~ is not empty

Since at this point 2 still thinks 1 is the owner, and 1 thinks 2 is the owner we want to stop this cycle (or restrict 2 to go inside handle)

So we add another statement saying

len(q-len-ch[me]) == 0

Figure 2: Explanation for the edit in the code



```
State-vector 244 byte, depth reached 4320, errors: 0
  5097380 states, stored (6.33179e+06 visited)
 34837754 states, matched
 41169541 transitions (= visited+matched)
  3348525 atomic steps
hash conflicts:   2912379 (resolved)
```

Figure 3: Output without error