

1 Peterson's Algorithm

2 Peterson's Algorithm: Fairness

3 Linearizability and Sequentially Consistency

3.1 Stack

C: s.empty()
A: s.push(10)
B: s.pop()
A: s:void
A: s.push(20)
B: s:10
A: s:void
C: s:true

This history is linearizable and therefore as well sequential consistent because C's s.empty() call can finish before any value is pushed towards the stack. Furthermore the s.push(10) operation must happen before the pop() command and before the push(20) operation, because s.pop() will return 10. Therefore we get the following hystory.

C: s.empty()
C: s:true
A: s.push(10)
A: s:void
B: s.pop()
B: s:10
A: s.push(20)
A: s:void

3.2 Queue

A: q.enq(x)
B: q.enq(y)
A: q:void
B: q:void
A: q.deq()
C: q.deq()
A: q:y
C: q:y