

Quiz 1.1

- What if we reverse the order of the first two lines the 2-process Peterson's algorithm

P0:

turn = 1;

flag[0] = true;

...

P1:

turn = 0;

flag[1] = true;

...

Would it work?

- Prove that Peterson's N-process algorithm ensures:
 - ✓ mutual exclusion: no two processes are in the critical section at a time
 - ✓ starvation freedom: every process in the trying section eventually reaches the critical section (assuming no process fails in the trying, critical, or exit sections)

Quiz 1.2: safety

1. Let S be a safety property. Show that if all **finite runs** of an implementation I are **safe** (belong to S) then **all** runs of I are safe
2. Show that every **unsafe** run σ has an **unsafe finite prefix** σ' : every extension of σ' is unsafe
3. Show that every property is an **intersection** of a safety property and a liveness property

Quiz 1.3: linearizability/progress

- Show that linearizability is **compositional**:
 - ✓ A history H on $A \times B$ is linearizable if and only if H_A and H_B are linearizable

- Show how the elements of the “periodic table of progress” are related to each other:
 - ✓ Property P is **weaker** than property P' if P' is a subset of P

Quiz 1.4: linearizability

- Show that the sequential queue implementation considered before is linearizable and wait-free *as is* if used by two processes: one performing only enqueue operations and one performing only dequeue operations
- Devise a simple queue implementation shared by any number of processes in which enqueue and dequeue operations can run concurrently (data races between these operations are allowed)