Report: Fair Unisex Bathroom Synchronization Using Semaphores (Turnstile + Baton Passing)

1 Problem Description

We consider the classic *unisex bathroom* synchronization problem with two types of threads: `Men()` and `Women()`. The shared resource (the bathroom) must satisfy:

- **Mutual exclusion between genders:** men and women must never be inside at the same time.
- **Concurrency within a gender:** multiple men may be inside together, and multiple women may be inside together.
- **Fairness:** when the opposite gender is waiting, new arrivals of the current gender should not bypass the waiting queue and "steal" the bathroom next.
- **No deadlocks:** the system must not reach a state where threads block forever due to circular waits.

2 Shared State and Semaphores

The implementation maintains the following shared counters:

- `nm`, `nw`: number of men/women currently inside the bathroom.
- `dm`, `dw`: number of men/women delayed (waiting to enter).

We use four semaphores:

- `e` (entry/exit mutex): a binary semaphore protecting all shared counters (`nm,nw,dm,dw`) and controlling the handoff logic.
- `q` (turnstile): a binary semaphore ensuring that only one arriving thread at a time can perform the "enter or queue" decision. This prevents barging and race conditions in the entry protocol.
- `m` and `w`: condition semaphores used as waiting queues for men and women. Threads block on `m` or `w` when they must wait, and are awakened via *baton passing*.

3 High-Level Strategy

The solution combines two well-known synchronization patterns:

1. **Turnstile (`q`) to prevent barging:** arrivals pass through a single gate so that the decision "enter now vs. queue" is made by only one thread at a time.

2. **Baton passing for fairness and simplicity:** when a thread leaves and the room becomes empty, it does not merely unlock the mutex—instead it *hands off* control directly to a waiting thread of the opposite gender, if any. This provides an ordered, starvation-resistant handoff at each phase boundary (empty room).

4 Entry Protocol

4.1 Men Entry

A man arriving to enter executes the following logic:

1. **Pass the turnstile:** `SEM_WAIT(q)` ensures only one arriving thread performs the decision.
2. **Lock shared state:** `SEM_WAIT(e)`.
3. **Decide enter vs. wait:**
   - If women are currently inside (`nw>0`) *or* women are already waiting (`dw>0`), then the man must not barge. He increments `dm` and releases both `e` and `q`, then blocks on `m`.
   - Otherwise, he releases `q` (allowing the next arrival to make a decision) and proceeds while holding `e`.
4. **Enter:** increment `nm`.
5. **Baton pass within same gender:** If `dm>0`, wake another waiting man by posting `m`; otherwise release `e`.

4.2 Women Entry

The women entry protocol is symmetric:

- A woman waits if men are inside (`nm>0`) or if men are already waiting (`dm>0`), which prevents women from barging ahead when men are queued.
- Otherwise she enters, increments `nw`, and may pass the baton to another waiting woman.

4.3 Why `q` is Important

Without `q`, two arrivals of opposite genders could interleave their checks and updates such that:

- both believe they can enter, or
- one bypasses the waiting condition because the opposing `delayed` counter has not yet been updated.

The turnstile ensures the check-and-queue decision is serialized, which removes these races and enforces *no barging*.

5 Exit Protocol

5.1 Men Exit

When a man exits:

1. lock `e`

2. decrement `nm`
3. if `nm==0` (room becomes empty) and some women are waiting (`dw>0`), decrement `dw` and post `w` to wake a waiting woman
4. otherwise release `e`

The key idea: **when the room becomes empty, priority goes to the opposite gender if it is waiting.** This is the fairness boundary.

5.2 Women Exit

Symmetric to men: when `nw` becomes 0, and `dm>0`, wake a man.

6 Correctness

6.1 Safety: No Mixed-Gender Occupancy

We show that it is impossible to reach a state where both `nm>0` and `nw>0`.

- A man increments `nm` only while holding `e`, after checking `nw==0` and `dw==0` at the decision point (or after being released via a baton that preserves the same decision regime).
- If any women are inside (`nw>0`), men do not increment `nm`; they increment `dm` and wait.
- Similarly, if any men are inside (`nm>0`), women queue and cannot increment `nw`.

Because all modifications and checks of `nm,nw,dm,dw` are protected by `e`, no interleaving can violate the mutual-exclusion condition.

6.2 Liveness: Deadlock Freedom

A deadlock requires a cycle of threads each waiting for a resource held by another. This design prevents cycles due to the following properties:

1. **Bounded holding of `q`:** `q` is held only during the brief entry decision. Threads either (a) release it and proceed, or (b) release it before blocking on `m` or `w`. Thus no thread blocks while holding `q`.
2. **Bounded holding of `e`:** threads release `e` before blocking on `m` or `w`. A blocked thread on `m/w` does not hold `e`.
3. **Baton passing ensures progress:** whenever the room becomes empty and the opposite gender is waiting, the exiting thread wakes exactly one waiting thread of that gender. This creates a clear progress path: empty room $\Rightarrow$ someone is released.

Therefore, no circular wait is possible involving `q`, `e`, and the condition semaphores, and the system is deadlock-free.

6.3 Fairness Properties

6.3.1 No Barging (Entry Fairness)

The entry conditions use both "inside" and "waiting" information:

- Men wait if `nw>0` *or* `dw>0`.

- Women wait if `nm>0` *or* `dm>0`.

This ensures that once a gender begins waiting, newly arriving threads of the opposite gender do not cut in front. The turnstile `q` strengthens this guarantee by serializing the check/queue decision.

6.3.2 Alternation at Empty-Room Boundaries

True alternation is enforced at the natural boundary where the bathroom becomes empty:

- When `nm` drops to 0, if `dw>0`, a woman is awakened.
- When `nw` drops to 0, if `dm>0`, a man is awakened.

Thus, if both genders are contending, each time the room empties the *other* gender gets the baton. This prevents one gender from monopolizing the bathroom across empty-room transitions.

7 Conclusion

The presented solution correctly enforces unisex occupancy with concurrent same-gender access. It uses:

- `e` as the mutex for all shared state,
- `q` as a turnstile preventing barging and entry races,
- `m` and `w` as waiting queues with baton passing for clean handoff.

The design avoids deadlocks by ensuring no thread blocks while holding `q` or `e`, and it enforces a strong practical fairness policy: when the bathroom becomes empty, it hands control to the opposite gender if they are waiting, while also preventing newcomers from overtaking existing waiters.