A Reusable Barrier

The Problem

The previous barrier solution will not work in a loop

- Why?

How do we solve this problem?

We need a reusable barrier that locks itself after all the threads have passed through

- Do it now!

Bad Solution: Explain why

```
mutex . wait ()
    count += 1
mutex . signal ()

if count == n : turnstile . signal ()

turnstile . wait ()
turnstile . signal ()

critical point

mutex . wait ()
    count -= 1
mutex . signal ()

if count == 0: turnstile . wait ()
```

Barrier hint

Use two turnstiles

```
turnstile = Semaphore (0)
turnstile2 = Semaphore (1)
mutex = Semaphore (1)
```

One turnstile is always locked!

Sometimes called a two phase barrier

Lab 4

Create a reusable barrier class in C++

- Add demo code that shows it working properly
- Include all the usual
  - Doxy comments
  - Makefile