Concurrent Data Structures Lab Assignment 3

Bernat Xandri Zaragoza

1. **Task 1:** Treiber Stack.

I implemented the Treiber stack following the code from the slides, and found a weird result. The code seems to work except for one instruction, where it fails. I tried implementing other tasks to help me debug the code, as well as adding verbose statements, but it seems to work correctly, as far as I could see. The error seems to arise from the comparation with the reference stack, but the Treiber stack seems to execute all the orders correctly, at least as far as I could see.
I also tried making a second implementation, based on the code provided in the book, instead of the slides, running into the same results. Given this situation, I think the code is correct but I'm making a mistake when logging the events in the test set. Therefore, I don't know how to fix it.

The linearization policy for my code would be always at the atomic Compare-And-Swap(CAS), which in C++ is done with a specific function applied to the checked object.
For the push method, the linearization point is the line 58, when the CAS operation successfully changes the `top` pointer to point to the new node. This is the exact moment when the new element is considered to be part of the stack.
For the pop method, the linearization point is in the line 87, when the CAS operation successfully updates the `top` pointer to point to the next node in the stack, therefore deleting the previous top.

2. **Task 2:** Lock-Free Set.

I implemented this algorithm using the template provided in the book.
For this implementation, the linearization policy would be:
For the add method, the linearization point would be the line 177, where the CAS method on the pred->next is successfully called. This is where the new node is effectively inserted into the set.
For the rmv method, it would be in the line 205, with the CAS. This is where the node is deleted from the list.
For the ctn it would be the line 225, where each node is being read.

3. **Task 3:** Benchmarking.

I tried to run the benchmarking. I had to update the makefile to make sure that it ran in C++17, since otherwise it was giving me errors. After this, I ran into the issue of getting a

Segmentation fault. When trying to debug this error using gdb, I get this message:

Which makes me think that the issue is in the benchmarking code, and the memory allocation of a vector in the generator module. Since this is not part of the code that I wrote, I tried to fix it but it feels a bit over my head, so didn't succeed. I would appreciate an explanation on why this issue happens.