

The UPC Memory Model

Examples

Memory model rules

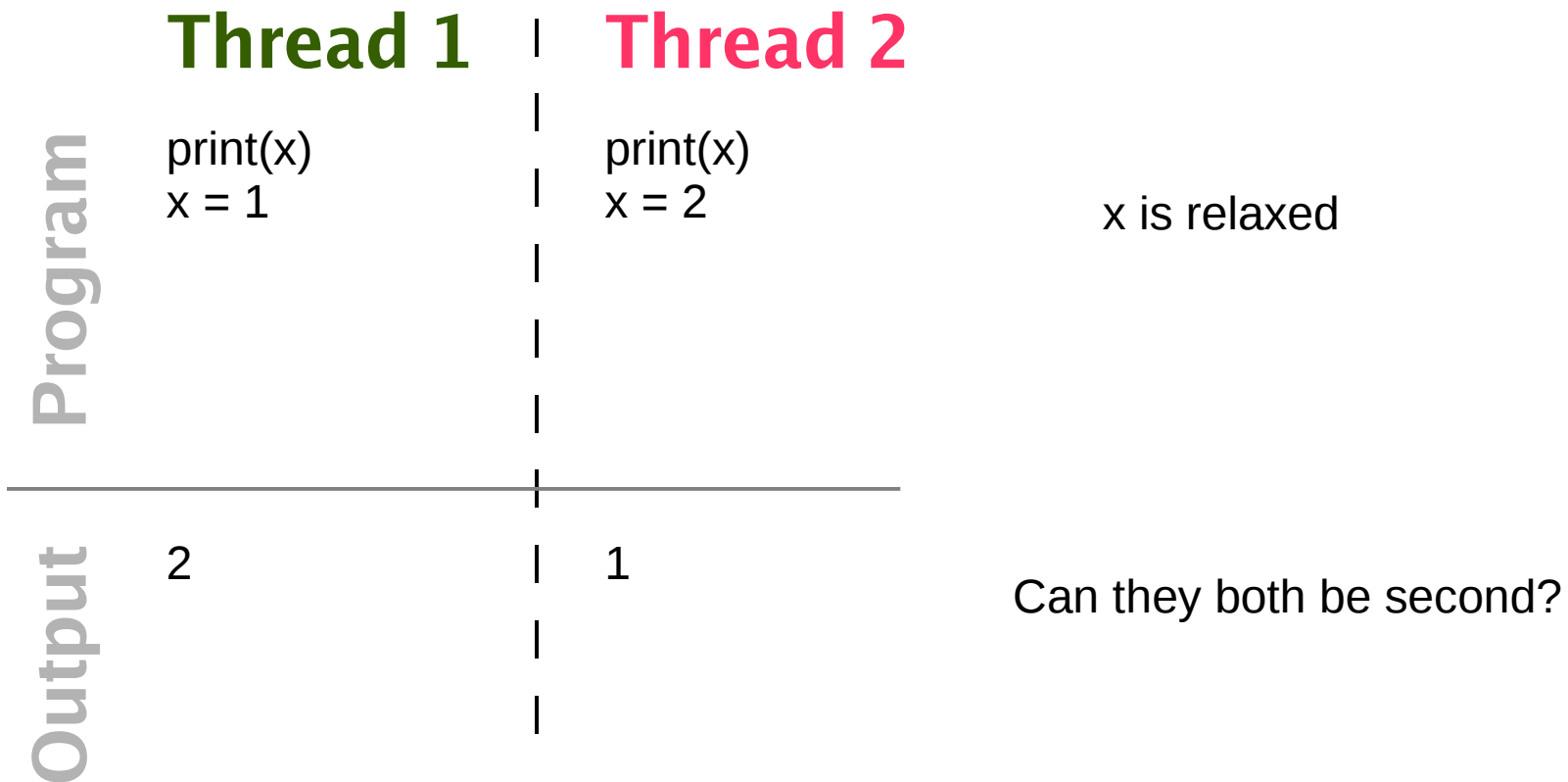
- Re-ordering restrictions on a thread:
 - ProgramOrder: Two operations cannot be reordered if they operate on the same variable and one of them is a write.
 - All threads must observe all writes (to shared memory) before the end of the program.
 - ReadOrder: A read operation gets its value from the latest observed write.

Memory model rules

- UPC specific rules:
 - StrictOrder: Two operations on the same thread cannot be reordered if at least one of them is strict.
 - All threads must observe all strict operations.
 - All threads must agree on the order of all strict operations.
 - All operations issued before a strict operation must be observed before that strict operation by all threads.

Example 1

- Behavior that is illegal in SC, but **legal** in UPC.
- “Who was here before me?”



Memory ops

- “print(x)” is converted to a relaxed read
- “x = ?” is converted to a relaxed write

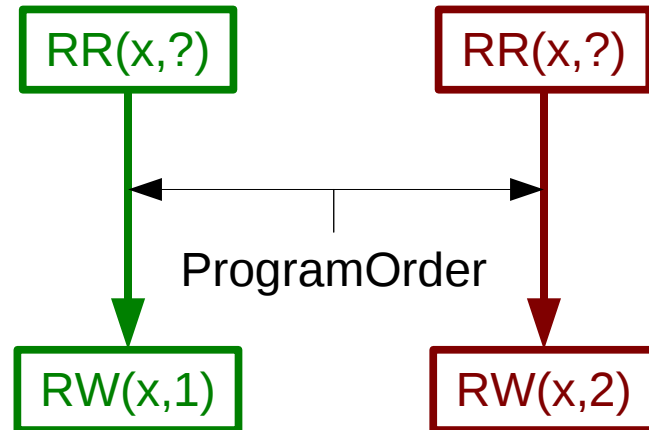
Program

Thread 1

print(x)
x = 1

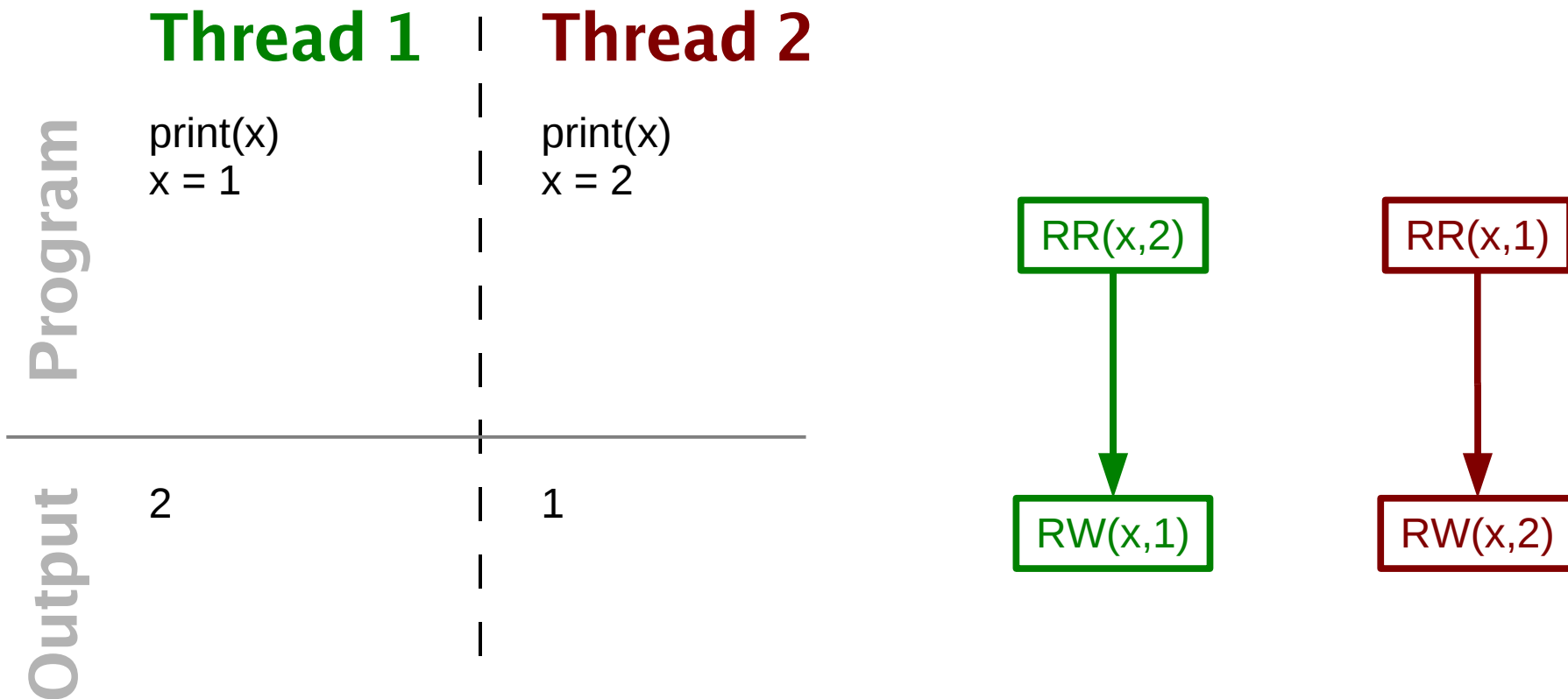
Thread 2

print(x)
x = 2



Adjusting for execution trace

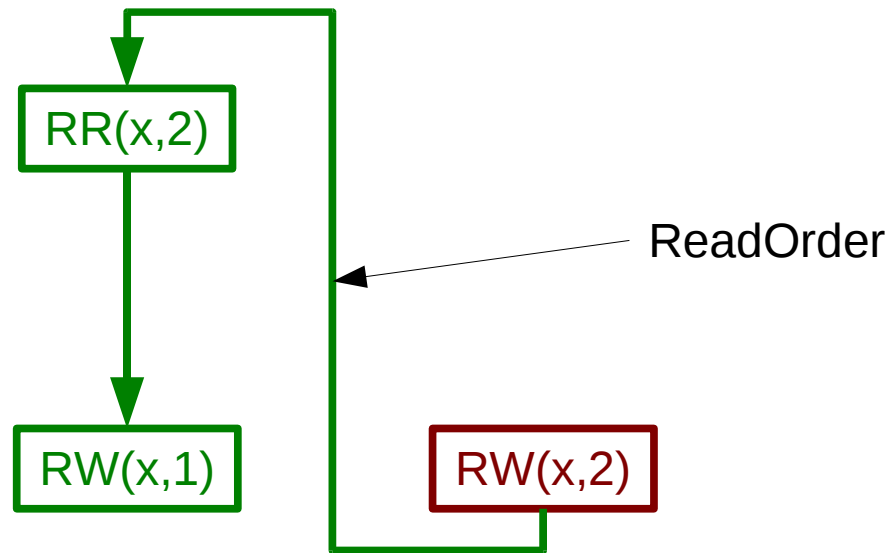
- Filling in the observed values for the reads



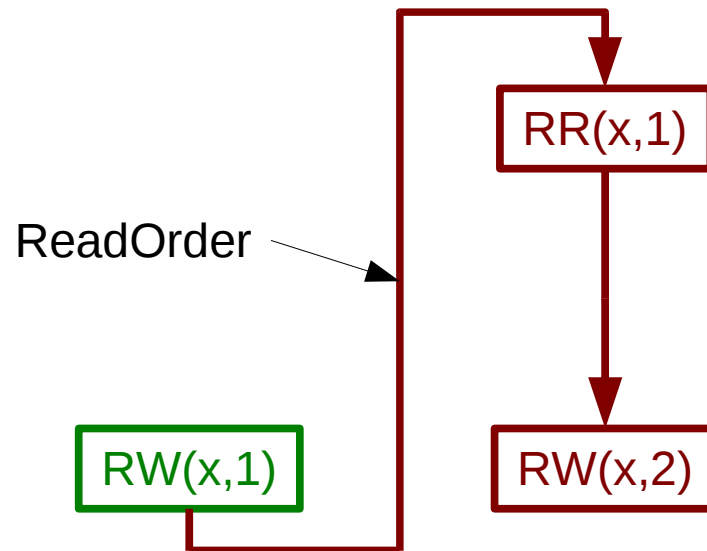
Strict global order

- There are no strict operations so no global strict order!

Thread 1 ordering

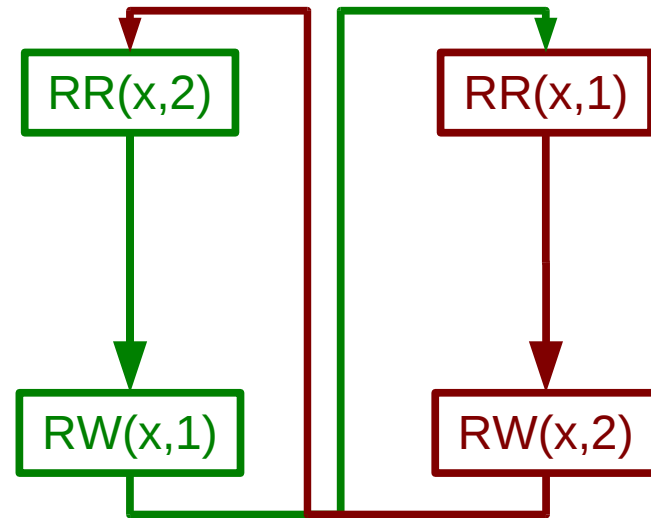


Thread 2 ordering



Example 1 - summary

- Both thread-orderings are legal, global ordering is legal, so execution trace is UPC MM compliant!
- Legal because threads can disagree on order of relaxed writes.



Example 1 - Fix

- Adding a fence

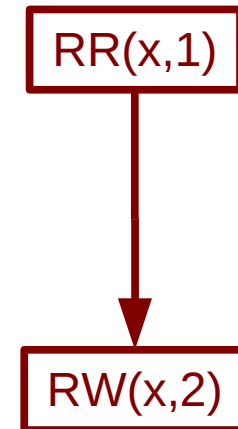
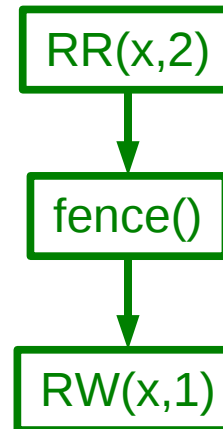
Program

Thread 1

```
print(x)  
upc_fence()  
x = 1
```

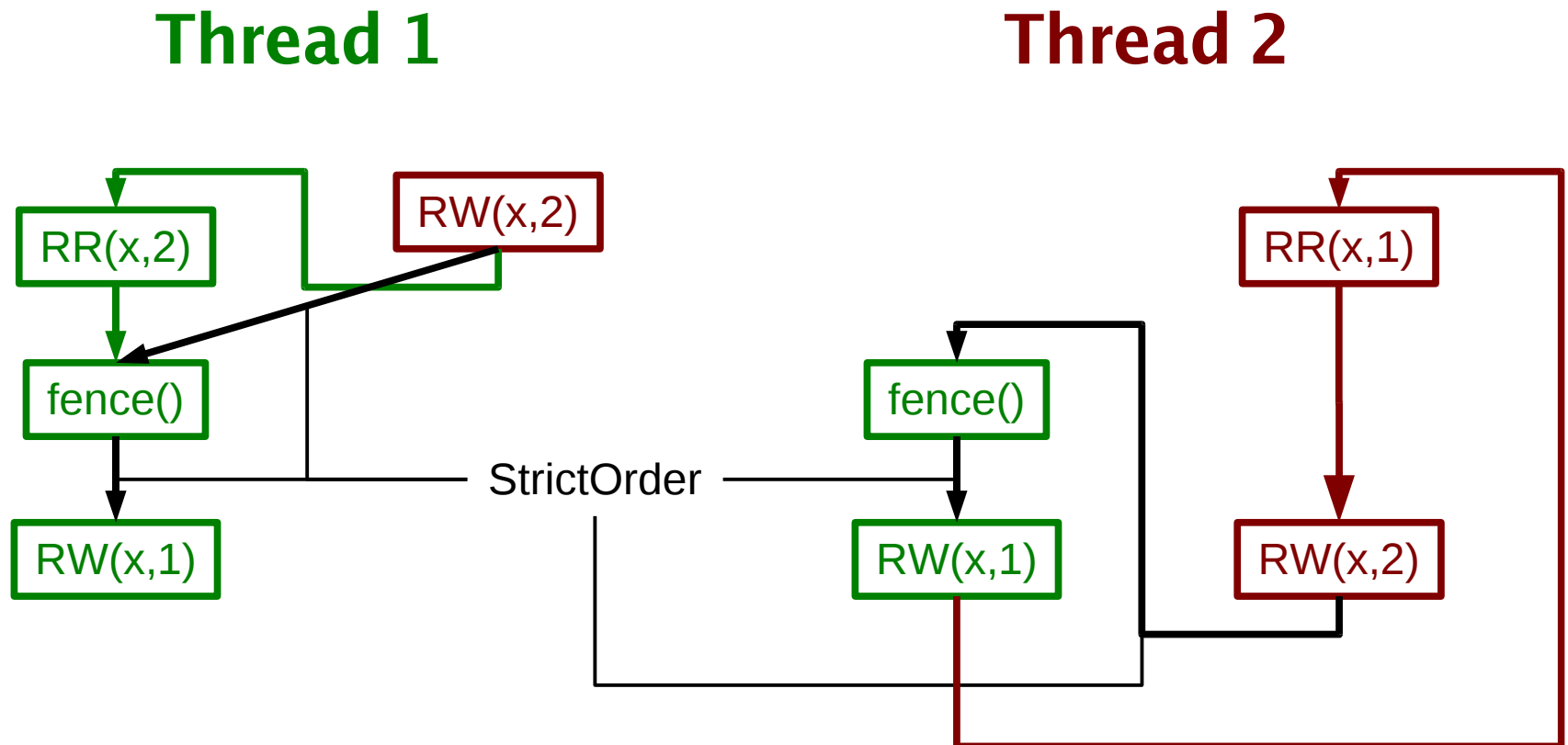
Thread 2

```
print(x)  
x = 2
```



Example 1 - Fix

- Adding a fence



Example 2

- Strict reads cannot be reordered

Program

Thread 1

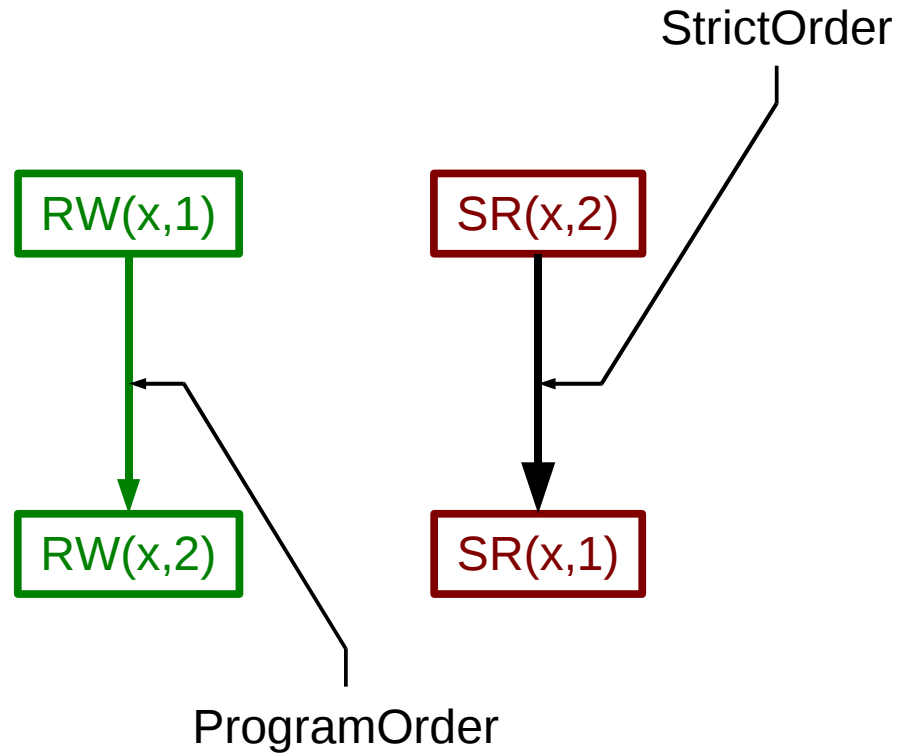
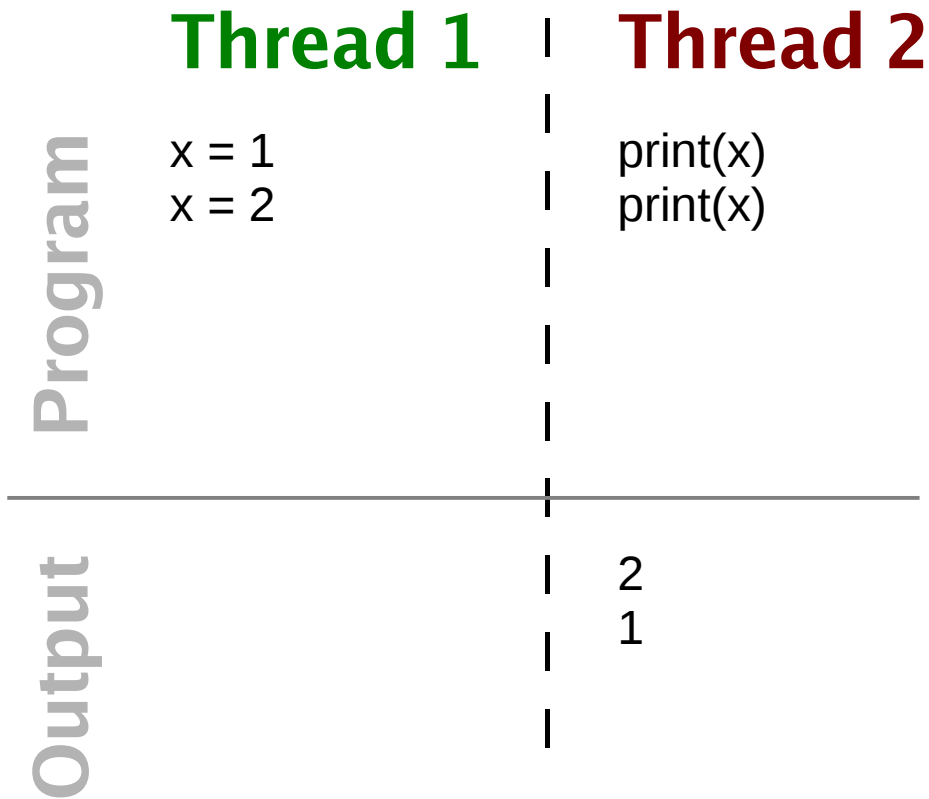
x = 1
x = 2

Thread 2

print(x)
print(x)

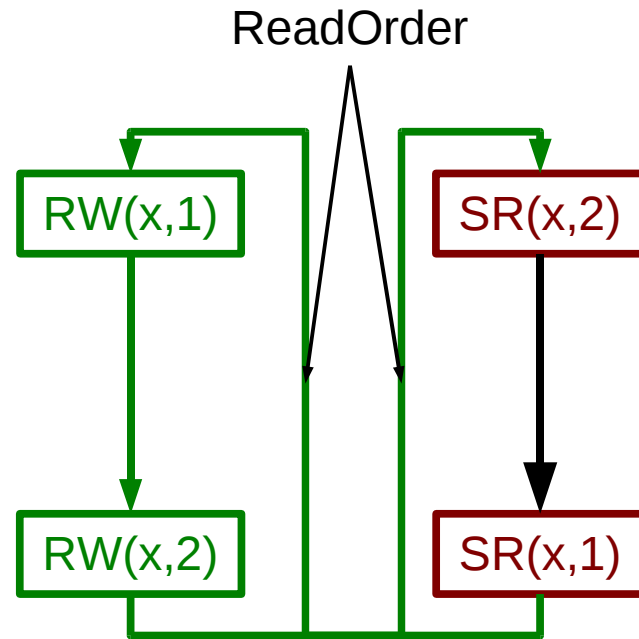
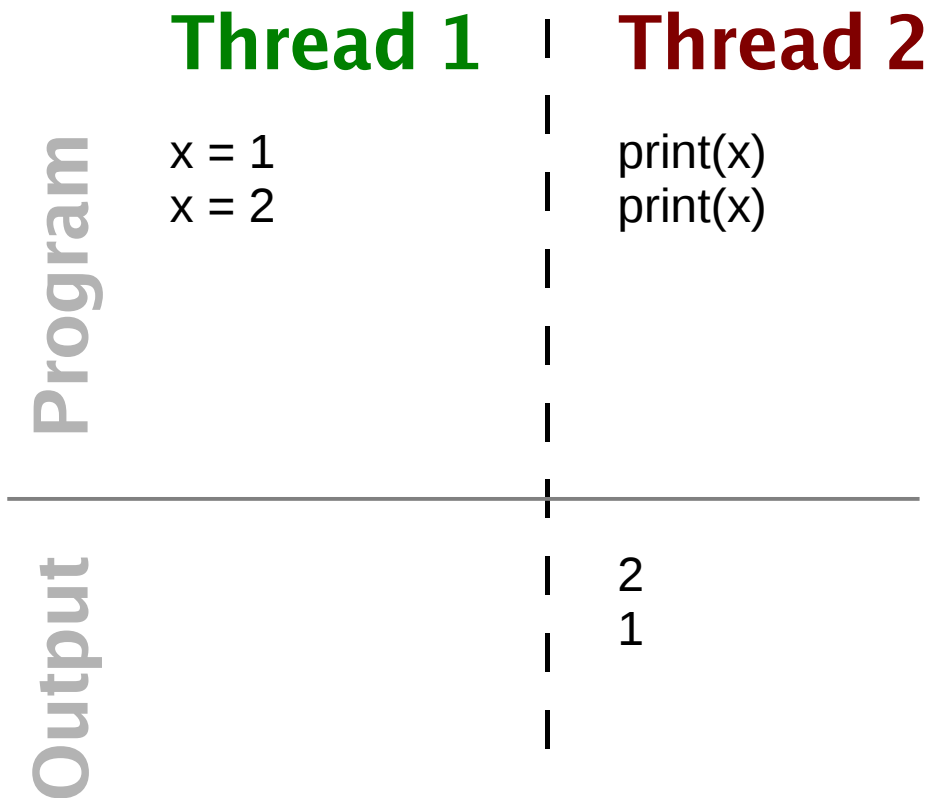
Thread 2 is using strict operations only.

Example 2



Example 2

- Both threads will have problems explaining the behavior.



Example 3

- But writes can be reordered

Program

Thread 1

x = 1
y = 1

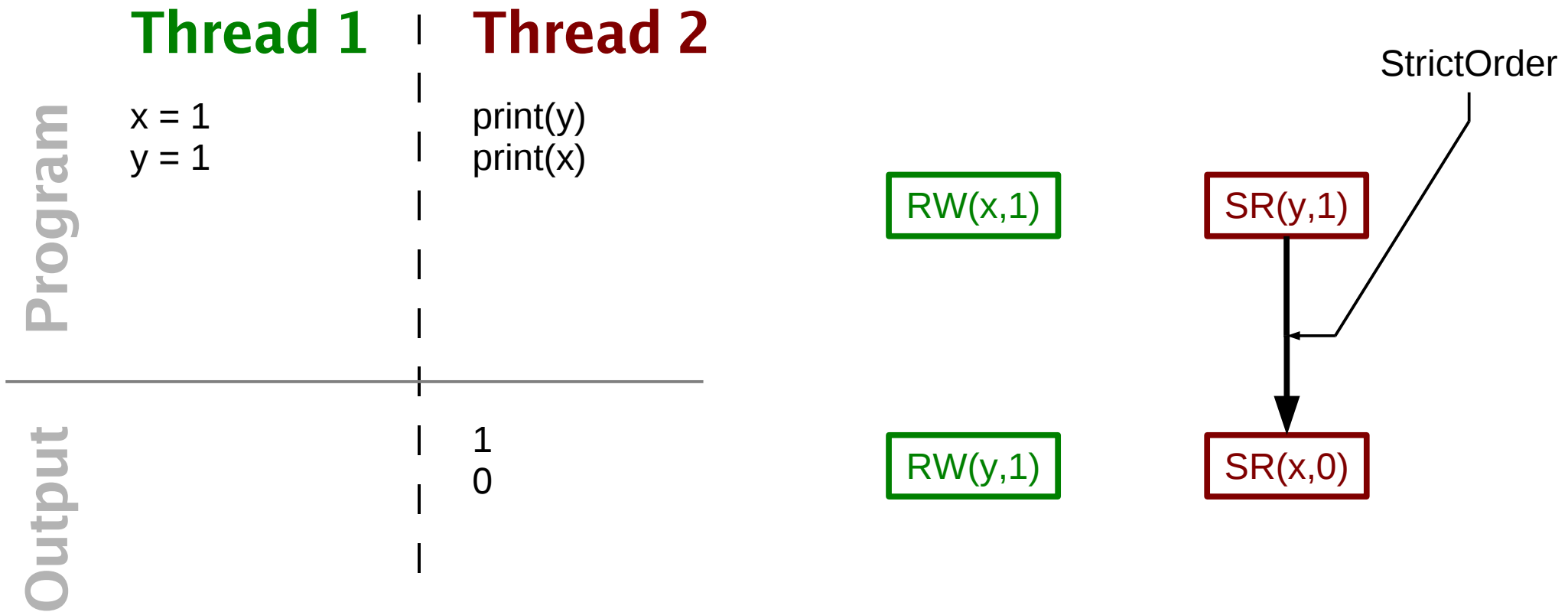
Thread 2

print(y)
print(x)

Thread 2 is using strict operations only.

Example 3

- But writes can be reordered



Example 3

- But writes can be reordered

