

SPINLOCK

В папке SpinLock/ лежат файл с описанием логики(spinlock.tla) работы и скрин с толбокс, что лок работает.

```
class SpinLock {
    std::atomic_flag locked = ATOMIC_FLAG_INIT ;
public:
    void lock() {
        while (locked.test_and_set(std::memory_order_acquire)) { ; }
    }
    void unlock() {
        locked.clear(std::memory_order_release);
    }
};
```

Инвариант:

TypeOK == $\bigvee (\text{who_is_in_critical} \setminus \text{subseq} \{ \text{first} \} \wedge \text{flag_locked} \setminus \text{subseq} \{ "T" \})$
 $\bigvee (\text{who_is_in_critical} \setminus \text{subseq} \{ \text{second} \} \wedge \text{flag_locked} \setminus \text{subseq} \{ "T" \})$
 $\bigvee (\text{who_is_in_critical} \setminus \text{subseq} \{ \text{no_one} \} \wedge \text{flag_locked} \setminus \text{subseq} \{ "F" \})$

Результат:

Инвариант не нарушен

Model Checking Results

General

Start: 16:55:08 (nov. 9) End: 16:55:09 (nov. 9) Not run

Fingerprint collision probability: calculated: 3.3E-19

Statistics

State space progress (click column header for graph)

Time	Diameter	States Found	Distinct States	Queue Size
00:00:01	2	5	3	0
00:00:01	0	1	1	1

Sub-actions of next-state (at 00:00:01)

Module	Action	Location	States Found	Distinct States
spinlock	GoOut	line 29, col 1 to line 29, col 13	2	0
spinlock	Init	line 14, col 1 to line 14, col 4	1	1
spinlock	GoIn	line 34, col 1 to line 34, col 12	2	2

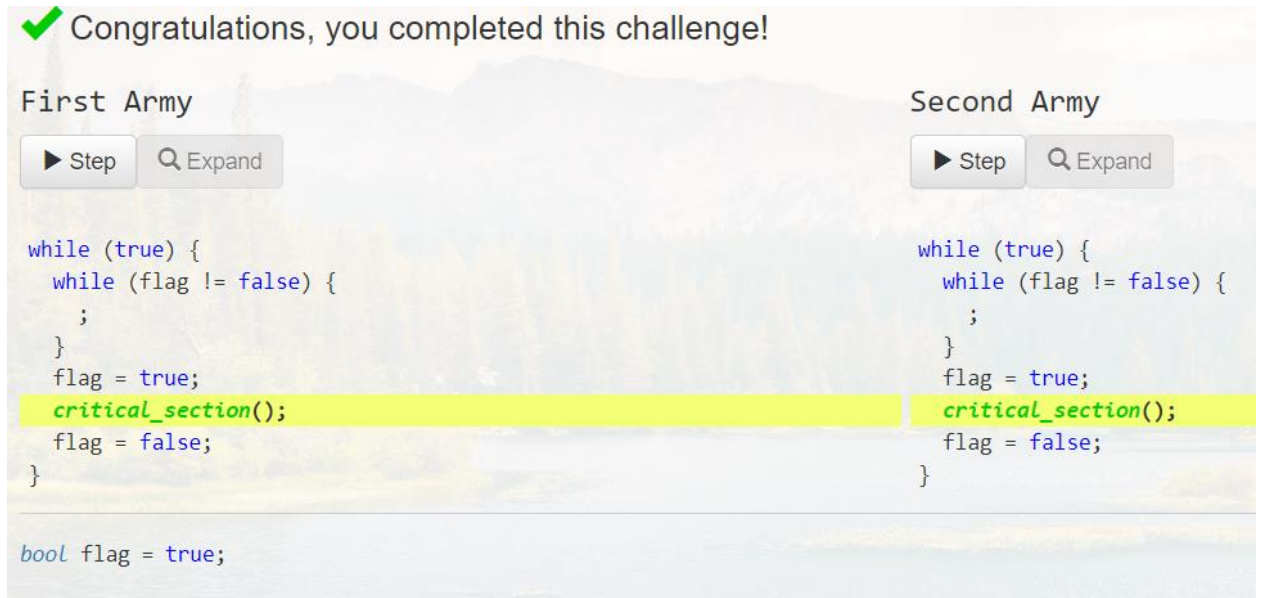
Evaluate Constant Expression

Expression:

Value:

Задача с ресурса

В папке 1/ есть решение задачи в 1.txt, описание логики в 1.tla, и описание задачи 1.PNG



Инвариант:

TypeOK == \bigvee (who_is_in_critical \subseteq { first })

\bigvee (who_is_in_critical \subseteq { second })

\bigvee (who_is_in_critical = {})

Результат:

Инвариант нарушен (лог в 1/1.txt я же приведу более читаемый вид)

- 1.
 - flag_locked \rightarrow "F",
 - who_is_after_while \rightarrow {},
 - who_is_in_critical \rightarrow {},
 - who_is_in_while \rightarrow {1, 2},
- 2.
 - flag_locked \rightarrow "F",
 - who_is_after_while \rightarrow {2},
 - who_is_in_critical \rightarrow {},
 - who_is_in_while \rightarrow {1}},
- 3.
 - flag_locked \rightarrow "F",

- who_is_after_while $\rightarrow \{1, 2\}$,
 - who_is_in_critical $\rightarrow \{\}$,
- 4.
 - flag_locked $\rightarrow \text{"T"}$,
 - who_is_after_while $\rightarrow \{1, 2\}$,
 - who_is_in_critical $\rightarrow \{\}$,
- 5.
 - flag_locked $\rightarrow \text{"T"}$,
 - who_is_after_while $\rightarrow \{2\}$,
 - who_is_in_critical $\rightarrow \{1\}$
- 6.
 - flag_locked $\rightarrow \text{"T"}$,
 - who_is_in_critical $\rightarrow \{1, 2\}$,